

# FileMaker® Server 16

Handbok för anpassad webbpublicering



**FileMaker®**  
An Apple Subsidiary

© 2004-2017 FileMaker, Inc. Med ensamrätt.

FileMaker, Inc.

5201 Patrick Henry Drive

Santa Clara, Kalifornien 95054, USA

FileMaker, FileMaker Go och filappslogotypen är registrerade varumärken som tillhör FileMaker, Inc. i USA och andra länder.

FileMaker WebDirect och FileMaker Cloud är varumärken som tillhör FileMaker, Inc. Alla övriga varumärken tillhör respektive ägare.

FileMakers dokumentation skyddas av lagen om upphovsrätt. Det är därför inte tillåtet att mångfaldiga eller distribuera detta dokument utan FileMakers skriftliga medgivande. Dokumentationen får endast användas tillsammans med ett licensierat exemplar av FileMaker-programmet.

Samtliga personer, företag, e-postadresser och URL-adresser som förekommer i exempel är fiktiva och eventuella likheter med verkliga personer, företag, e-postadresser eller URL-adresser är fullständigt oavsiktliga. En lista över medverkande visas i dokumentet som medföljer den här programvaran. Omnämning av tredjepartsprodukter och URL-adresser förekommer endast i informationssyfte och ska inte tolkas som förslag eller rekommendationer. FileMaker, Inc. tar inget ansvar när det gäller prestandan hos dessa produkter.

Mer information finns på webbplatsen <http://www.filemaker.com/se/>.

Utgåva: 01

# Innehåll

<b>Förord</b>	8
Om denna handbok	8
Hitta FileMaker-dokumentation	8
 <b>Kapitel 1</b>	
<b><i>Introduktion till anpassad webbpublicering</i></b>	9
Om Web Publishing Engine	10
Hur en Web Publishing Engine-förfrågan behandlas	10
Anpassad webbpublicering med XML	11
Anpassad webbpublicering med PHP	11
Jämförelse mellan XML och PHP	11
Anledningar till att välja XML	11
Anledningar till att välja PHP	12
 <b>Kapitel 2</b>	
<b><i>Förbereda databaser för anpassad webbpublicering</i></b>	13
Aktivera anpassad webbpublicering i en databas	13
Få åtkomst till en skyddad databas	13
Skydda publicerade databaser	14
Webbserverstöd för Internet-mediatyper (MIME)	15
Publicera innehållet i containerfält på webben	15
Containerfältobjekt som bäddas in i en databas	15
Containerfält med lagrade filreferenser	16
Containerfält med externt lagrade data	16
Containerfält och progressiv nedladdning	17
Hur webbanvändare kan visa containerfältdata	17
FileMaker-scripts och anpassad webbpublicering	18
Tips och råd om scripts	18
Funktionssätt för scripts i egna webbpubliceringslösningar	19
Scripttrigger och anpassade webbpubliceringslösningar	19
 <b>Kapitel 3</b>	
<b><i>Anpassad webbpublicering med XML</i></b>	20
Skapa dynamiska webbplatser med Web Publishing Engine	20
Viktiga funktioner i anpassad webbpublicering med XML	21
Krav för webbpublicering	21
Vad som krävs för att publicera en databas med anpassad webbpublicering	21
Vad webbanvändare behöver för att kunna använda en anpassad webbpubliceringslösning	21
Ansluta till internet eller ett intranät	22
Så går du vidare	22

## Kapitel 4

<b>Åtkomst till XML-data med Web Publishing Engine</b>	23
Använda anpassad webbpublicering med XML	23
Skillnader mellan Web Publishing Engine och XML-import/export i FileMaker Pro	23
Hur Web Publishing Engine genererar XML-data från en förfrågan	24
Åtkomst till XML-data från Web Publishing Engine	25
URL-syntax för XML-data och containerobjekt	25
URL-syntax för XML-data	25
URL-syntax för FileMaker-containerobjekt i XML-lösningar	26
URL-textkodning	27
Åtkomst till XML-data med Web Publishing Engine	28
Namnuttryck för FileMaker XML	28
Felkoder i FileMaker-databaser	29
Hämta dokumenttypsdefinitionerna för syntaxerna i FileMaker	29
Använda syntaxen fmresultset	29
Beskrivning av element i syntaxen fmresultset	30
XML-data i syntaxen fmresultset	32
Använda andra FileMaker XML-syntaxer	33
Beskrivning av element i syntaxen FMPXMLRESULT	33
XML-data i syntaxen FMPXMLRESULT	34
Beskrivning av element i syntaxen FMPXMLLAYOUT	35
XML-data i syntaxen FMPXMLLAYOUT	37
Om UTF-8-kodade data	38
Efterfråga XML-data med hjälp av frågesträngar i FileMaker	38
Växla layout för ett XML-svar	40
Bearbetning av en XML-förfrågan	41
Felsöka åtkomst till XML-dokument	41

## Kapitel 5

<b>Giltiga namn i XML-frågesträngar</b>	42
Frågekommandon och frågeparametrar	42
Riktlinjer för frågekommandon och frågeparametrar	43
Tolkning av frågekommandon	44
Syntax för ett fullständigt fältnamn	45
Använda frågekommandon med portalfält	45
Syntax för globala fält	47
Referens för frågekommando	47
Frågekommandot –dbnames (databasnamn)	47
Frågekommandot –delete (radera post)	47
Frågekommandot –dup (duplicera post)	48
Frågekommandot –edit (redigera post)	48
Frågekommandona –find, –findall och –findany (sök efter poster)	49
Frågekommando –findquery (sammanfattad sökning)	49
Frågekommandot –layoutnames (layoutnamn)	50
Frågekommandot –new (ny post)	50
Frågekommandot –scriptnames (scriptnamn)	51
Frågekommandot –view (visa layoutinformation)	51

Referens för frågeparameter	51
Frågeparametern –db (databasnamn)	51
Frågeparametern –delete.related (radera portalposter)	52
Frågeparametern –field (namn på containerfält)	52
Frågeparametern fältnamn (namn på annat fält än containerfält)	52
Frågeparametern fältnamn.op (jämförelseoperator)	53
Frågeparametern –lay (layout)	54
Frågeparametern –lay.response (växla layout vid svar)	54
Frågeparametern –lop (logisk operator)	55
Frågeparametern –max (maximalt antal poster)	55
Frågeparametern –modid (ändrings-ID)	56
Frågeparametern –query (sammanfattad sökning)	56
Frågeparametern –recid (post-ID)	57
Frågeparametern –relatedsets.filter (filtrera portalposter)	58
Frågeparametern –relatedsets.max (begränsa portalposter)	59
Frågeparametern –script (script)	59
Frågeparametern –script.param (skicka parameter till script)	60
Frågeparametern –script.prefind (script före sökning)	60
Frågeparametern –script.prefind.param (skicka parameter till script före sökning)	60
Frågeparametern –script.presort (script före sortering)	61
Frågeparametern –script.presort.param (skicka parameter till script före sökning)	61
Frågeparametern –skip (hoppa över poster)	62
Frågeparametern –sortfield (sortera fält)	62
Frågeparametern –sortorder (sorteringsordning)	63

## Kapitel 6

<b><i>Om anpassad webbpublicering med PHP</i></b>	64
Viktiga funktioner i anpassad webbpublicering med PHP	64
Krav för anpassad webbpublicering	64
Vad som krävs för att publicera en databas med anpassad webbpublicering	64
Vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning	65
Ansluta till Internet eller ett intranät	65
Installera FileMaker API för PHP manuellt	66
Så går du vidare	67

## Kapitel 7

<b><i>Översikt över anpassad webbpublicering med PHP</i></b>	68
Hur Web Publishing Engine fungerar med PHP-lösningar	68
Allmänna steg för anpassad webbpublicering med PHP	68

## Kapitel 8

<b><i>Använda FileMaker API för PHP</i></b>	70
Mer information	70
Referens för FileMaker API för PHP	70
Support för FileMaker API för PHP	70

Använda FileMaker class	71
FileMaker class-objekt	71
FileMaker kommandoobjekt	71
Avkoda data för användning i FileMaker API	72
Ansluta till en FileMaker-databas	72
Arbeta med poster	73
Skapa en post	73
Duplicera en post	73
Redigera en post	74
Radera en post	74
Köra FileMaker-scripts	75
Hämta en lista på tillgängliga scripts	75
Köra ett FileMaker-script	75
Köra ett script innan ett kommando utförs	75
Köra ett script innan en resultatuppsättning sorteras	76
Köra ett script efter resultatuppsättningen har genererats	76
Ordning för att utföra scripts	76
Arbeta med FileMaker-layouter	77
Använda portaler	78
Lista de portaler som har definierats på en specifik layout	78
Hämta portalnamn för ett specifikt resultatobjekt	78
Hämta information om portaler för en specifik layout	78
Hämta information för en specifik portal	78
Hämta tabellnamnet för en portal	78
Hämta portalposterna för en specifik post	79
Skapa en ny post i en portal	79
Ta bort en post från en portal	79
Använda värdelistor	80
Hämta namnen på alla värdelistor för en specifik layout	80
Hämta en matris över alla värdelistor för en specifik layout	80
Hämta värdena för en namngiven värdelista	80
Utföra sökningar	81
Använda kommandot Visa alla	82
Använda kommandot Sök efter alla	82
Använda kommandot Utför sökning	82
Använda kommandot Sammanfattad sökning	83
Behandla posterna i ett sökresultat	85
Begränsa portalrader som returneras av en sökförfrågan	86
Förkontrollera kommandon, poster och fält	86
Förkontrollera poster i ett kommando	87
Förkontrollera poster	88
Förkontrollera fält	88
Behandla kontrollfel	88
Hantera fel	90

## Kapitel 9

<i>Iscensätta, testa och övervaka en webbplats</i>	91
Iscensätta en anpassad webbpubliceringsplats	91
Testa en anpassad webbpubliceringsplats	92
Formatmallar som du kan använda när du testar XML-utdata	93
Övervaka webbplatsen	94
Använda webbserveråtkomst och felloggar	94
Använda Web Publishing Engine-loggen	94
Använda felloggen för webbservern	96
Använda Tomcat-loggar	97

## Bilaga A

<i>Felkoder för anpassad webbpublicering</i>	98
Felkodsnummer i XML-format	98
Felkodsnummer för FileMaker-databaser	99

<i>Index</i>	100
--------------	-----

# Förord

## Om denna handbok

Den här handboken riktar sig till dig som har erfarenhet av att skapa databaser med hjälp av FileMaker® Pro. Du bör förstå grunderna i databasdesign i FileMaker Pro och känna till begreppen fält, relationer, layouter, portaler och containrar. Information om FileMaker Pro finns i [FileMaker Pro Hjälp](#).

Den här handboken förutsätter också att du har erfarenhet av att utveckla webbplatser, särskilt med tekniker som XML och PHP, för att integrera FileMaker-data med webbplatser och webbprogram.

Handboken innehåller följande information om anpassad webbpublicering med FileMaker Server:

- vad som krävs för att du ska kunna utveckla en anpassad webbpubliceringslösning
- hur man publicerar databaser med XML
- hur du hämtar XML-data från databaser på FileMaker Server
- hur man publicerar databaser med PHP
- hur man använder FileMaker API för PHP för att hämta data från databaser som FileMaker Server är värd för
- vad som krävs för att en webbanvändare ska kunna få åtkomst till en anpassad webbpubliceringslösning

## Hitta FileMaker-dokumentation

- Välj **Hjälp**-menyn > **FileMaker Server produktdokumentation** i FileMaker Server Admin Console.
- Klicka på länkarna på startsidan i FileMaker Server Admin Console.
- Besök <http://www.filemaker.com/documentation/sv> om du vill läsa om, visa och hämta mer FileMaker-dokumentation.

Du kan nå onlinehjälp från FileMaker Server Admin Console. Välj **Hjälp**-menyn > **FileMaker Server Hjälp**.



# Kapitel 1

## Introduktion till anpassad webbpublicering

Med FileMaker Server kan du publicera din FileMaker-databas på Internet eller ett intranät på följande sätt.

**FileMaker WebDirect:** Med FileMaker WebDirect publicerar du layouter från en databas snabbt och smidigt på webben. Du behöver inte installera ytterligare program – med en kompatibel webbläsare och tillgång till Internet eller ett intranät kan webbanvändare ansluta till din FileMaker WebDirect-lösning och visa, redigera, sortera och söka i poster om du ger dem behörighet.

Med FileMaker WebDirect måste värddatorn köra FileMaker Server. Användargränssnittet liknar FileMaker Pro-programmet. Webbsidorna och formulär som webbanvändaren samverkar med är beroende av de layouter och vyer som definierats i FileMaker Pro-databasen. Se [Guide för FileMaker WebDirect](#).

**Statisk publicering:** Om du ändrar dina data väldigt sällan eller om du inte vill att användarna ska ha en direktanslutning till din databas kan du använda dig av statisk publicering. Vid statisk publicering exporterar du data från en FileMaker Pro-databas och skapar en webbsida som du kan anpassa ytterligare med HTML. Webbsidan ändras inte när informationen i din databas ändras och användarna öppnar inte din databas. (Med FileMaker WebDirect uppdateras data i webbläsaren när dessa data uppdateras i databasen.) Se [FileMaker Pro Hjälp](#).

**FileMaker Data API:** Om du är van vid att använda REST-arkitektur (Representational State Transfer) erbjuder FileMaker en REST API-implementering som gör att webbtjänster kan komma åt data i värdbaserade lösningar. Din webbtjänst anropar FileMaker Data API för att hämta en autentiseringstoken för åtkomst till en värdbaserad lösning. Därefter används denna token i efterföljande anrop för att skapa poster, uppdatera poster, radera poster och utföra sökförfrågningar. FileMaker Data API returnerar data i JSON (JavaScript Object Notation). Se [FileMaker Guide för Data API](#).

**Anpassad webbpublicering:** Du kan integrera en FileMaker-databas med en anpassad webbplats genom att använda teknikerna för anpassad webbpublicering som finns tillgängliga i FileMaker Server. FileMaker Server, som är värd för de publicerade databaserna, kräver inte att FileMaker Pro är installerat eller igång för att anpassad webbpublicering ska vara tillgänglig.

Med Anpassad webbpublicering kan du:

- integrera databasen med en annan webbplats
- avgöra hur användarna ska få tillgång till data
- kontrollera hur data ska visas i webbläsare.

FileMaker Server innehåller två tekniker för anpassad webbpublicering:

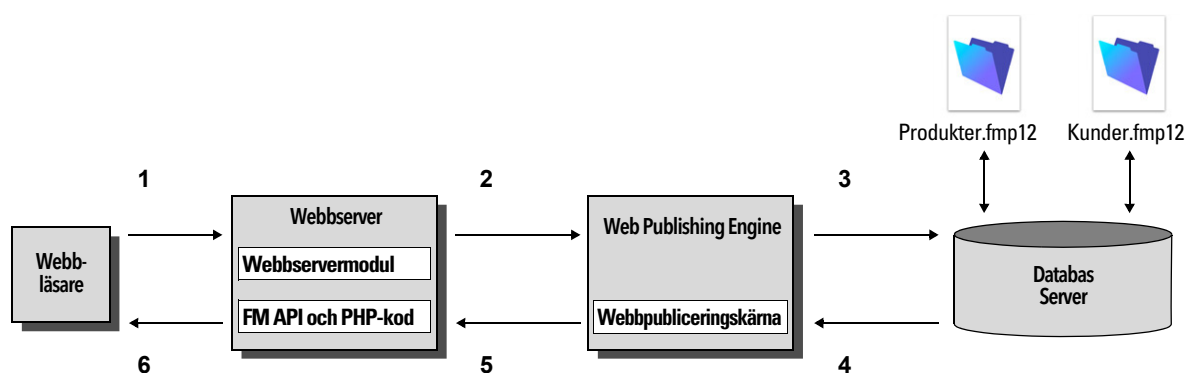
- Anpassad webbpublicering med XML: Använd XML-datapublicering för att utbyta FileMaker-data med andra webbplatser och program. Genom att använda HTTP URL-förfrågningar tillsammans med FileMaker-frågekommandon och tillhörande frågeparametrar kan du ställa frågor till en databas som drivs med FileMaker Server, hämta resultatet i XML-format och använda dessa data på valfritt sätt.
- Anpassad webbpublicering med PHP: Använd FileMaker API för PHP, med ett objektorienterat PHP-gränssnitt för FileMaker Pro-databaser, för att integrera dina FileMaker-data i ett PHP-webbprogram. Eftersom du kodar PHP-webbsidorna själv har du full kontroll över användargränssnittet och användarens möjligheter att interagera med alla data.

## Om Web Publishing Engine

För att stödja FileMaker WebDirect och anpassad webbpublicering använder FileMaker Server en uppsättning programvarukomponenter som kallas för *FileMaker Server Web Publishing Engine*. Web Publishing Engine hanterar samspillet mellan en webbanvändares webbläsare, din webserver och FileMaker Server.

**Anpassad webbpublicering med XML:** Webbanvändare får åtkomst till din anpassade webbpubliceringslösning genom att klicka på en HREF-länk eller skriva en URL-adress till webbservern och en FileMaker-frågesträng. Web Publishing Engine returnerar de XML-data som anges i frågesträngen.

**Anpassad webbpublicering med PHP:** När en webbanvändare ansluter till din anpassade webbpubliceringslösning ansluter PHP på FileMaker Server till Web Publishing Engine och svarar genom FileMaker API för PHP.



Använda FileMaker Servers Web Publishing Engine för anpassad webbpublicering

## Hur en Web Publishing Engine-förfrågan behandlas

1. En förfrågan skickas från en webbläsare eller ett program till webbservern.
2. Webbservern vidarebefordrar förfrågan genom FileMakers webbservermodul till Web Publishing Engine.
3. Web Publishing Engine efterfrågar data på databasen som databasservern är värd för.
4. FileMaker Server skickar efterfrågade FileMaker-data till Web Publishing Engine.
5. Web Publishing Engine konverterar FileMaker-data för att svara på förfrågan.
  - När det gäller PHP-förfrågningar konverterar FileMaker API för PHP dem till XML-förfrågningar. Web Publishing Engine bearbetar XML-förfrågan och skickar XML-data tillbaka till FileMaker API för PHP. FileMaker API för PHP konverterar sedan XML-data till PHP-objekt som kan användas av PHP-programmet.
  - För XML-förfrågningar skickar Web Publishing Engine XML-data direkt till webbservern.

6. Webbservern skickar utdata tillbaka till webbläsaren eller programmet.

**Viktigt!** Säkerheten är mycket viktig när du publicerar information på webben. Gå igenom säkerhetsriktlinjerna i handboken [FileMaker Säkerhet](#).

Mer information om att förbereda en databas för anpassad webbpublicering finns i kapitel 2, "Förbereda databaser för anpassad webbpublicering".

## Anpassad webbpublicering med XML

Med FileMaker anpassad webbpublicering med XML kan du skicka förfrågningar till en FileMaker Pro-databas som FileMaker Server är värd för, och visa, ändra eller hantera resulterande data. Med en HTTP-förfrågan som innehåller rätt frågekommandon och -parametrar kan du hämta FileMaker-data som ett XML-dokument. Därefter kan du exportera dina XML-data till andra program.

## Anpassad webbpublicering med PHP

FileMaker API för PHP ger ett objektorienterat PHP-gränssnitt för FileMaker-databaser. FileMaker API för PHP gör att man kan få åtkomst till både uppgifter och logik som har lagrats i en FileMaker Pro-databas och publicera dem på Internet, eller exportera dem till andra program. API stöder också komplexa och sammansatta sökkommandon för att extrahera och filtrera data som har lagrats i FileMaker Pro-databaser.

PHP utformades ursprungligen som ett procedurmässigt programmeringsspråk, och har utvecklats som ett objektorienterat webbutvecklingsspråk. PHP ger programspråksfunktionalitet för att skapa praktiskt taget alla typer av logik på en webbsida. Till exempel kan du använda villkorliga logikkonstruktioner för att kontrollera sidgenerering, datarouting eller arbetsflöde. PHP ger även möjlighet till webbplatsadministration och säkerhet.

## Jämförelse mellan XML och PHP

Följande avsnitt ger riktlinjer för hur du avgör vilken lösning som passar bäst för din webbplats.

### Anledningar till att välja XML

- FileMaker XML frågeparametersyntax är utformat för interaktion mellan databaser, vilket underlättar lösningsutveckling.
- XML är en W3C-standard.
- XML är ett format som kan läsas av både datorer och människor och som stöder Unicode, vilket gör att data kan kommuniceras på alla skriftliga språk.
- XML passar bra för att presentera poster, listor och trädstrukturerade data.
- Du kan använda FMPXMLRESULT för att komma åt XML-data med Anpassad webbpublicering och för att exportera XML från FileMaker Pro-databaser.

**Obs!** Information om anpassad webbpublicering med XML finns i kapitel 3, "Anpassad webbpublicering med XML".

## Anledningar till att välja PHP

- PHP är ett mer kraftfullt, objektorienterat procedurmässigt scriptspråk och är relativt enkelt att lära sig. Det finns många tillgängliga resurser för utbildning, utveckling och support.
- FileMaker API för PHP gör att man kan få åtkomst till uppgifter och logik som har lagrats i en FileMaker Pro-databas och publicera dem på Internet, eller exportera dem till andra program.
- Med PHP kan man använda villkorslogik för att kontrollera sidkonstruktionen eller flödet.
- PHP ger programspråksfunktionalitet för att skapa många typer av logik på en webbsida.
- PHP är ett av de populäraste webbscriptspråken.
- PHP är ett språk med öppen källkod, som finns på [php.net](http://php.net).
- PHP ger åtkomst till ett stort utbud av komponenter från tredje part som du kan integrera i dina lösningar.

**Obs!** Information om anpassad webbpublicering med PHP finns i kapitel 6, "Om anpassad webbpublicering med PHP".

# Kapitel 2

## Förbereda databaser för anpassad webbpublicering

Innan du kan använda anpassad webbpublicering med en databas måste du förbereda databasen och skydda den från obehörig åtkomst.

### Aktivera anpassad webbpublicering i en databas

Du måste aktivera en utökad behörighet för anpassad webbpublicering i varje databas som du vill publicera. Om du inte aktiverar en utökad behörighet för anpassad webbpublicering i databasen kommer webbanvändarna inte att kunna komma åt databasen med anpassad webbpublicering. Detta gäller även om den administreras med FileMaker Server och programmet har konfigurerats så att det stödjer Web Publishing Engine.

**Så här aktiverar du anpassad webbpublicering för en databas:**

1. Starta FileMaker Pro och öppna databasen som du vill publicera med hjälp av ett konto som har behörighetsuppsättningen Full åtkomst. Du kan också öppna databasen med hjälp av ett konto som har inställningen Hantera utökad behörighet i behörighetsuppsättningen.
2. Tilldela den utökade behörigheten för anpassad webbpublicering som du vill använda:
  - För anpassad webbpublicering med XML använder du fmxml
  - För anpassad webbpublicering med PHP använder du fmphp
3. Koppla behörighetsuppsättningen/behörighetsuppsättningarna som innehåller den utökade behörigheten för anpassad webbpublicering till ett eller fler konton eller till kontot Admin eller Gäst.

**Obs!** När du definierar kontonamn och lösenord för lösningar för anpassad webbpublicering använder du utskrivbara ASCII-tecken, till exempel, **a-z**, **A-Z** och **0-9**. För att göra kontonamn och lösenord säkrare inkluderar du skiljetecken, t.ex. **"!"** och **"%"** men inkludera inte kolon. Information om hur du konfigurerar konton finns i [FileMaker Pro Hjälp](#).

### Få åtkomst till en skyddad databas

Anpassad webbpublicering gör det möjligt att begränsa åtkomsten till publicerade databaser genom lösenordsskydd, databaskryptering och säkra anslutningar. När webbanvändare använder en anpassad webbpubliceringslösning för att få åtkomst till en databas kan de bli ombudda att ange kontoinformation. Om Gästkontot för databasen är inaktiverat eller inte har en behörighetsuppsättning med utökad behörighet för anpassad webbpublicering, använder Web Publishing Engine grundläggande HTTP-verifiering för verifiering av webbanvändarna. I webbanvändarens webbläsare visas dialogrutan för grundläggande HTTP-verifiering där användaren kan ange användarnamn och lösenord för ett konto som har utökad behörighet för anpassad webbpublicering.

I följande lista sammanfattas processen som sätts igång när en webbanvändare använder lösningen anpassad webbpublicering för att få åtkomst till en databas:

- Om du inte har angett något lösenord för ett konto behöver webbanvändaren bara ange kontonamnet.
- Om Gästkonto är inaktiverat måste användarna ange kontonamn och lösenord när de öppnar databasen. Kontot måste ha utökad behörighet för anpassad webbpublicering aktiverad.
- Om Gästkonto är aktiverat och har en behörighetsuppsättning som innehåller utökad behörighet för anpassad webbpublicering, kan alla webbanvändare automatiskt öppna databasen med den behörighet som Gästkonto har. Om Gästkonto har utökad behörighet för anpassad webbpublicering gäller följande:
  - Webbanvändare behöver inte ange kontonamn eller lösenord när de öppnar en fil.
  - Alla webbanvändare loggar automatiskt in med gästkontot och får den behörighet som hör till gästkontot. Du kan låta användarna ändra sina inloggningsuppgifter via webbläsaren med scriptsteget Logga in igen (till exempel för att växla från Gästkonto till ett konto med fler behörigheter).
  - Standardbehörigheten för gästkonton ger endast läsbehörighet. Du kan ändra standardbehörigheten, även den utökade behörigheten, för detta konto. Se [FileMaker Pro Hjälp](#).
- När en webbanvändare har angett giltiga kontouppgifter återanvänds de så länge som webbläsarsessionen är igång. När webbläsarsessionen avbryts uppmanas webbanvändaren att ange ett giltigt konto igen.

**Obs!** Som standardinställning kan webbanvändare inte ändra sina egna kontolösenord från en webbläsare. Du kan bygga in denna funktion för en databas med hjälp av scriptsteget Ändra lösenord, som gör att webbanvändarna kan ändra sina lösenord från webbläsaren. Se [FileMaker Pro Hjälp](#).

## Skydda publicerade databaser

När du använder anpassad webbpublicering kan du begränsa vilka som ska ha åtkomst till din publicerade databas.

- Tilldela lösenord till databaskonton som används för anpassad webbpublicering.
- Aktivera en utökad behörighet för anpassad webbpublicering endast i behörighetsuppsättningarna för de konton som du vill ge åtkomst till dina publicerade databaser.
- Inaktivera den utökade behörigheten för anpassad webbpublicering för en specifik databas genom att avmarkera den utökade fmxml- eller fmphp-behörigheten för alla behörighetsuppsättningar i den databasen. Se [FileMaker Pro Hjälp](#).
- Aktivera eller inaktivera Anpassad webbpublicering för alla Lösningar för anpassad webbpublicering med FileMaker Server Admin Console. Se [Installations- och konfigurationsguiden för FileMaker Server](#) och [FileMaker Server Help](#).
- Konfigurera webbservern för att begränsa vilka IP-adresser som ska kunna använda databasen via Web Publishing Engine. Du kan exempelvis ange att endast webbanvändare från IP-adress 192.168.100.101 ska ha tillgång till databaserna. Information om hur du begränsar åtkomst från IP-adresser finns i dokumentationen till webbservern.

FileMaker Server stöder kryptering för data som sparats på disk och för data som överförs till klienter.

- Kryptera databasen med funktionen Databaskryptering i FileMaker Pro Advanced. Krypteringen skyddar FileMaker-databasfilen och alla temporära filer som sparats på disk. Se [Installations- och konfigurationsguiden för FileMaker Server](#) och [FileMaker Pro Help](#).
  - En krypterad databas som FileMaker Server är värd för öppnas med hjälp av Admin Console eller kommandoradsgränssnittet (CLI). Som FileMaker Server-administratör öppnar du filen med krypteringslösenordet, så att FileMaker-klienter kan använda den krypterade databasen.
  - När FileMaker Server-administratören har öppnat den krypterade FileMaker-databasen med krypteringslösenordet av, behöver FileMaker-klienterna inte ange krypteringslösenordet för att komma åt den krypterade databasen. Mer information om hur du öppnar en krypterad databas finns i [FileMaker Server Hjälp](#).
- Använd SSL-kryptering (Secure Sockets Layer) för kommunikationen mellan webbservern och webbläsaren. SSL-anslutningar nås genom en HTTPS-anslutning. FileMaker Server tillhandahåller ett SSL-standardcertifikat som signerats av FileMaker, Inc. men som inte verifierar servernamnet. FileMakers standardcertifikat är endast avsett att användas för intern testning. Ett anpassat SSL-certifikat krävs för publicering. Se [Installations- och konfigurationsguiden för FileMaker Server](#).

Om du aktiverar **Använd HSTS för webbklienter** i Admin Console använder du HTTPS-katalogen som värd för webbplatsens PHP-filer. Se kapitel 7, "Allmänna steg för anpassad webbpublicering med PHP".

Mer information om att skydda din databas finns i handboken [FileMaker Säkerhet](#).

**Obs!** Av säkerhetsskäl kan webbsidor som andra webbservrar är värd till inte använda <iframe>-taggen för att bädda in innehåll för anpassad webbpublicering. Om du vill bädda in innehåll för anpassad webbpublicering i <iframe>-taggarna för separata webbsidor måste webbservern FileMaker Server vara värd för de webbsidorna.

## Webbservernsupport för Internet-mediatyper (MIME)

Webbservern fastställer stöd för de aktuella MIME-typer (Multipurpose Internet Mail Extensions) som har registrerats för Internet. Web Publishing Engine ändrar inte en webbserverns MIME-stöd. Se dokumentationen till webbservern.

## Publicera innehållet i containerfält på webben

Innehållet i ett containerfält kan bäddas in i databasen med en länk som anges med en relativ sökväg, men det kan också lagras externt.

### Containerfältobjekt som bäddas in i en databas

Om ett containerfält innehåller de faktiska filerna i FileMaker-databasen behöver du inte göra något med innehållet i containerfältet, under förutsättning att FileMaker Server fungerar som värd för databasfilen och den är tillgänglig på FileMaker Server. Se "URL-syntax för FileMaker-containerobjekt i XML-lösningar" på sidan 26.



## Containerfält med lagrade filreferenser

Om ett containerfält lagrar en filreferens, måste du följa dessa steg för att publicera de externa filerna med hjälp av Web Publishing Engine.

1. Lagra containerobjektfilerna i webbmappen inuti FileMaker Pro-mappen.
2. Starta FileMaker Pro, sätt in objekten i containerfältet och välj **Spara endast länk till filen**.
3. Kopiera eller flytta objektfilerna som refereras till i webbmappen till samma relativa sökväg i rotmappen i webbrowserprogrammet.
  - För IIS (Windows):  
`[enhet] : \Program\FileMaker\FileMaker Server\HTTPServer\conf`  
där [enhet] är den enhet där Web Publishing Engine-komponenten för din FileMaker Server-driftsättning finns.
  - För Apache (macOS): `/Bibliotek/FileMaker Server/HTTPServer/htdocs`

**Obs!** För att containerobjekt ska kunna lagras som filreferenser måste webbservern kunna hantera MIME-typerna för de typer av filer du vill använda, till exempel filmer. Webbservern fastställer stöd för de aktuella MIME-typer som har registrerats för Internet. Web Publishing Engine ändrar inte en webbservers MIME-stöd. Se dokumentationen till webbservern.

## Containerfält med externt lagrade data

Om ett containerfält har lagrat objekt externt – det vill säga om du har valt **Lagra containerdata externt** i dialogrutan Tillval för fält i FileMaker Pro – överför du databasfilerna från klientens filsystem till FileMaker Server med hjälp av FileMaker Pro. När du överför en databas med hjälp av FileMaker Pro, överförs externt lagrade containerfältdata till FileMaker Server som en del av processen. Information om hur du överför databasfiler till FileMaker Server finns i [FileMaker Pro Hjälp](#).

När du manuellt överför en databas som använder ett containerfält med externt lagrade objekt, måste du följa dessa steg när du ska publicera de externt lagrade containerobjekten med Web Publishing Engine.

### Så här överför du en databas manuellt:

1. Placera databasfilen på rätt plats på servern. Placera de FileMaker Pro-databasfiler som ska öppnas av FileMaker Server – eller genvägar (Windows) eller alias (macOS) till filerna – i följande mappar:
  - Windows:  
`[enhet] : \Program\FileMaker\FileMaker Server\Data\Databases\`  
där [enhet] är den primära enhet där programmet startas.
  - macOS: `/Bibliotek/FileMaker Server/Data/Databases/`Du kan också placera dina filer i någon annan databasmapp som du har angett.
2. I den mapp där du placerade databasen skapar du en mapp med namnet RC\_Data\_FMS, om den inte redan finns.



3. I mappen RC\_Data\_FMS skapar du sedan en ny mapp med samma namn som din databas. Om din databas till exempel heter Kunder skapar du en mapp som också heter Kunder. Placera de externt lagrade objekten i den nya mappen som du just skapade.

**Obs!** När databaserna administreras med FileMaker Server finns det inget sätt för flera databaser att dela en gemensam mapp bestående av containerobjekt. Containerobjekten för varje databas måste finnas i en mapp som identifieras av databasens namn.

4. För filer som ska delas från macOS ändrar du dem så att de tillhör gruppen **fmsadmin**.

Mer information om hur du överför databaser manuellt finns i [FileMaker Server Hjälp](#).

## Containerfält och progressiv nedladdning

Web Publishing Engine stöder progressiv nedladdning av ljudfiler (.mp3), videofiler (.mov, .mp4, och .avi rekommenderas), och PDF filer för interaktiva containrar. En webbanvändare kan till exempel börja titta på en film även om inte hela filmfilen har laddats ner ännu. Om du vill möjliggöra progressiv nedladdning kan du behöva skapa filer genom att använda alternativ som stöder direktuppspelning eller som optimerar för visning på webben. Skapa till exempel PDF-filer genom att använda alternativet för webbvisningsoptimering.

När alternativet **Använd SSL för databasanslutningar** i FileMaker Server är valt, kommer FileMaker Server att använda säkra anslutningar för att överföra data över HTTPS.

- Interaktiva containerdata laddas ner över HTTPS.
- Data är lika säkra som om den värdbaserade lösningen var en lokal databas, eftersom det inte skapas några temporära cachefiler och data krypteras under överföring.

Om alternativet **Använd SSL för databasanslutningar** i FileMaker Server inte är valt, kommer anslutningarna som FileMaker Server använder för att överföra data inte vara krypterade under överföringen och data skickas då över HTTP.

- FileMaker-klienter ser interaktiva containerdata med minimal fördröjning.
- FileMaker Server cachelagrar containerfältdata till en cachemapp på servern när en FileMaker Pro-, FileMaker Go- eller webbklient skickar en förfrågan om data. Dessa data kan vara kvar i cachemappen på servern i två timmar innan FileMaker Server regelbundet tömmer cachemappen. Dessa data cachelagras inte lokalt på klienten.

Starta om FileMaker Server-tjänsten (Windows) eller FileMaker Server-bakgrundsprocesserna (macOS) när inställningen **Använd SSL för databasanslutningar** ändras för att de nya inställningarna ska börja gälla.

## Hur webbanvändare kan visa containerfältdata

När du publicerar en databas med Web Publishing Engine gäller följande begränsningar för containerfältobjekt:

- Webbanvändare kan inte ändra eller lägga till innehåll i ett containerfält. Webbanvändare kan inte använda containerfält för att ladda upp objekt till databasen.
- När det gäller databaser som använder ett containerfält som har miniatyrbilder aktiverat, laddar Web Publishing Engine ned hela filen, inte bara en miniatyrbild.

## FileMaker-scripts och anpassad webbpublicering

Funktionen Hantera scripts i FileMaker Pro kan automatisera uppgifter som utförs ofta eller kombinera flera uppgifter. Genom att använda FileMaker-scripts tillsammans med anpassad webbpublicering kan webbanvändare utföra flera uppgifter eller en serie uppgifter.

FileMaker har stöd för många scriptsteg för anpassad webbpublicering. Webbanvändare kan utföra en mängd olika automatiserade uppgifter när du använder scripts i en frågesträng för en URL-adress. Du kan se vilka scriptsteg som anpassad webbpublicering har stöd för genom att klicka på knappen **Kompatibilitet** och välja **Anpassad webbpublicering** i Scriptfönster i FileMaker Pro. Scriptstegen som inte är nedtonade stöds för anpassad webbpublicering. Information om hur du skapar scripts finns i [FileMaker Pro Hjälp](#).

### Tips och råd om scripts

Många scriptsteg fungerar på samma sätt på webben men det finns några som fungerar annorlunda. Se "Funktionssätt för scripts i egna webbpubliceringslösningar" på sidan 19. Testa alla scripts som ska användas från en webbläsare innan du delar din databas. Logga in med olika användarkonton så att du är säker på att de fungerar för alla klienter. Se om du hittar några scriptrelaterade fel i loggfilen för Web Publishing Engine (wpe.log). Se "Använda Web Publishing Engine-loggen" på sidan 94.

Här följer några tips och råd:

- Tänk på vilka värden ett script kan returnera. Var beredd att hantera alla data som returneras. I FileMaker Pro kan ett script returnera alla poster från en tabell eller från det aktuella sökresultatet. Men om ett script returnerar alla poster från en tabell kan ett webbprogram få slut på minne när det försöker att bearbeta posterna. Överväg att använda frågeparametern `-max`, med XML-frågor eller `setRange()`-metoden med PHP-frågor för att begränsa antalet poster som returneras.
- Använd konton och behörighet när du vill styra vilka scripts en webbanvändare kan köra. Kontrollera att scripten endast innehåller webbkompatibla scriptsteg och ge bara tillgång till scripts som ska användas från en webbläsare.
- Var också uppmärksam på eventuella biverkningar av scripts som utför kombinationer av steg som styrs via behörigheten. Om ett script exempelvis innehåller steget Radera poster och en webbanvändare inte loggar in med ett konto som ger behörighet att radera poster, utförs inte scriptsteget Radera poster. Scriptet kan dock fortsätta att utföras, vilket kan ge oväntade resultat.
- I Scriptfönster ger du ett script fullständiga behörigheter om du vill låta scriptet utföra uppgifter som du inte vill låta den enskilda användaren utföra. Du kan till exempel hindra användare från att radera poster via deras konton och behörighet, men ändå tillåta dem att utföra ett script som raderar vissa typer av poster under vissa förutsättningar som anges i scriptet.
- Du kan tillåta scripts att installera plugin-program för anpassad webbpublicering och FileMaker WebDirect-lösningar genom att aktivera inställningen **Tillåt att scriptsteget Installera plugin-fil får uppdatera plugin-program för webbpublicering** i FileMaker Server Admin Console. Du kan förhindra scripts från att installera plugin-program för webbpubliceringslösningar genom att avmarkera inställningen.

- Vissa scripts som fungerar med ett steg från en FileMaker Pro-klient kan dessutom kräva scriptsteget Verkställ post/sökpost för att spara data på värddatorn. Eftersom webbanvändarna inte har någon direktanslutning till värddatorn får de inte veta när data ändras. Funktioner som till exempel villkorliga värdelistor är inte lika smidiga för webbanvändare eftersom data måste sparas på värddatorn innan effekterna märks i fältet med värdelistan.
- Scripts som ändrar data bör innehålla scriptsteget Verkställ post/sökpost, eftersom vissa dataändringar inte syns i webbläsaren förrän data sparas eller "skickas" till servern. Detta gäller flera scriptsteg, exempelvis Klipp ut, Kopiera och Klistra in. Många enkelstegsåtgärder bör konverteras till scripts och innehålla steget Verkställ post/sökpost. När du skapar scripts som ska utföras från en webbläsare ska du inkludera steget Verkställ post/sökpost sist i scriptet, så att du är säker på att alla ändringar verkställs.
- Du kan skapa villkorliga scripts baserat på typen av klient med funktionen Get (Programversion). Om det returnerade värdet innehåller strängen "Web Publishing Engine" vet du att den aktuella användaren använder din databas via anpassad webbpublicering. Mer information om funktioner finns i [FileMaker Pro Hjälp](#).
- Öppna varje script som webbanvändare kan tänkas köra och kontrollera att scriptet kan köras ordentligt när databasen är konfigurerad som en anpassad webbpubliceringslösning. Kontrollera att scriptet bara använder scriptsteg som stöds för anpassad webbpublicering, enligt beskrivningen ovan.

### Funktionssätt för scripts i egna webbpubliceringslösningar

Vissa scriptsteg fungerar på ett annat sätt i anpassade webbpubliceringslösningar än i FileMaker Pro. I [FileMaker Pro Hjälp](#) finns information om kompatibilitet.

Scripts i anpassade webbpubliceringslösningar kan inte utföra scripts i andra FileMaker-filer om inte dessa filer är konfigurerade på samma installation av FileMaker Server och samma utökade behörigheter för anpassad webbpublicering är aktiverade i andra filer.

### Scripttrigger och anpassade webbpubliceringslösningar

I FileMaker Pro kan både scripts och användaråtgärder (som att användaren klickar i ett fält) aktivera scripttrigger. I anpassad webbpublicering kan endast scripts aktivera scripttrigger. Mer information om scripttrigger finns i [FileMaker Pro Hjälp](#).

**Obs!** I webbpubliceringslösningar aktiveras inte scripttriggern VidFörstaFönsterÖppna eller VidSistaFönsterStäng. Kör scripts manuellt med hjälp av frågeparametern XML –script eller PHP-metoden `newPerformScriptCommand()`.

# Kapitel 3

## Anpassad webbpublicering med XML

### Skapa dynamiska webbplatser med Web Publishing Engine

Med Web Publishing Engine kan du använda Anpassad webbpublicering tillsammans med FileMaker Server för XML-datapublicering. Anpassad webbpublicering har flera fördelar:

- **Anpassning:** Du kan bestämma vad webbanvändarna ska kunna göra med FileMaker-data och hur informationen visas i webbläsare.
- **Datautbyte:** Genom att använda FileMaker XML kan du utbyta FileMaker-data med andra webbplatser och program.
- **Dataintegrering:** Du kan integrera FileMaker-data med andra webbplatser, med mellanprogramvara och med anpassade program. Du kan göra så att informationen ser ut att höra till en annan webbplats i stället för att visa hela FileMaker-layouten i webbläsaren.
- **Säkerhet:** Administratören för FileMaker Server kan separat aktivera eller inaktivera XML-webbpublicering för alla databaser som körs på servern. Som ägare till en FileMaker-databas kan du styra användarnas åtkomst till XML-webbpublicering för varje databas.
- **Styra och filtrera publicerade data:** Du kan styra och filtrera informationen och vilken typ av databasinformation du vill publicera, och på så sätt förhindra obehörig användning av databasen. Du kan också dölja metadata, till exempel databas- och fältnamn.
- **Öppen standard:** Du har bättre åtkomst till verktyg, resurser och erfaren personal för anpassade webbpubliceringslösningar. Om du känner till standard-XML kan du börja utveckla lösningar så snart du har lärt dig ett par detaljer om anpassad webbpublicering med XML, till exempel URL-syntaxen och vilka frågeparametrar som ska användas.

Anpassad webbpublicering med XML medger att du hämtar data från FileMaker-databaser och enkelt använder dessa data i andra utdataformat. Med en HTTP-förfrågan som innehåller rätt frågekommandon och -parametrar kan du hämta FileMaker-data som ett XML-dokument. Därefter kan du använda dina XML-data i andra program. Se "Åtkomst till XML-data med Web Publishing Engine" på sidan 28.

## Viktiga funktioner i anpassad webbpublicering med XML

Anpassad webbpublicering med XML i FileMaker Server innehåller många viktiga funktioner:

- Databaserna finns på FileMaker Server, så FileMaker Pro behöver inte vara igång.
- Du kan använda bearbetning på serversidan av XML med hjälp av JavaScript.
- Precis som i FileMaker Pro är åtkomsten till data, layouter och fält baserad på användarkontots inställningar som anges i databasens behörighetsinställningar. Web Publishing Engine har också flera andra säkerhetsförbättringar. Se "Skydda publicerade databaser" på sidan 14.
- Webbanvändare kan utföra komplexa scripts med många steg. FileMaker har stöd för många scriptsteg för anpassad webbpublicering. Se "FileMaker-scripts och anpassad webbpublicering" på sidan 18.
- Du kan skicka ett parametervärde till ett FileMaker-script. Se "Frågeparametern –script.param (skicka parameter till script)" på sidan 60, "Frågeparametern –script.prefind.param (skicka parameter till script före sökning)" på sidan 60 och "Frågeparametern –script.presort.param (skicka parameter till script före sökning)" på sidan 61.
- Med den nya syntaxen `fmresultset` i XML får du åtkomst till fält efter namn och kan hantera `relatedset` (portaldata).
- Om du vill ha åtkomst till informationen i en databas måste du ange en layout. Se kapitel 5, "Giltiga namn i XML-frågesträngar",

## Krav för webbpublicering

### Vad som krävs för att publicera en databas med anpassad webbpublicering

Om du ska publicera databaser med Anpassad webbpublicering med XML, behöver du följande:

- en FileMaker Server-installation som innehåller
  - en webbserver, antingen Microsoft IIS (Windows) eller Apache (macOS)
  - FileMaker Server-databas, aktiverad för anpassad webbpublicering
  - Web Publishing Engine, installerad och konfigurerad
- en eller fler FileMaker Pro-databaser på FileMaker Server
- IP-adressen eller domännamnet på värden där webbservern körs
- en webbläsare och åtkomst till webbservern så att du kan utveckla och testa den egna webbpubliceringslösningen

Se [Installations- och konfigurationsguiden för FileMaker Server](#).

### Vad webbanvändare behöver för att kunna använda en anpassad webbpubliceringslösning

Webbanvändare som behöver ha åtkomst till en anpassad webbpubliceringslösning som använder XML måste ha tillgång till följande:

- en webbläsare
- åtkomst till Internet eller ett intranät och webbservern
- IP-adressen eller domännamnet på värden där webbservern körs

Om databasen är lösenordsskyddad måste webbanvändare även uppge ett användarnamn och ett lösenord för ett databaskonto.

## Ansluta till internet eller ett intranät

När du publicerar databaser på Internet eller ett intranät måste värddatorn köra FileMaker Server och de databaser du vill dela måste köras och vara tillgängliga. Dessutom gäller följande:

- Publicera din databas på en dator med fast Internet- eller intranätanslutning. Det går att publicera databaser utan att ha en fast anslutning, men dessa blir bara tillgängliga för webbanvändarna när din dator är ansluten till Internet eller ett intranät.
- Värddatorn för den webbserver som är en del av FileMaker Server-driftsättningen måste ha en dedicerad *statisk* (permanent) IP-adress eller ett domännamn. Om du ansluter till Internet via en Internetleverantör (ISP) kan din IP-adress vara *dynamiskt allokerad* (dvs. olika varje gång du ansluter). Med en dynamisk IP-adress blir det svårare för webbanvändarna att hitta dina databaser. Om du inte är säker på vilken typ av anslutning du ska välja kontaktar du din Internetleverantör eller nätverksadministratör.

## Så går du vidare

Här följer några förslag för att komma igång med att utveckla egna webbpubliceringslösningar:

- Aktivera anpassad webbpublicering med hjälp av FileMaker Server Admin Console om du inte redan har gjort det. Se [FileMaker Server Help](#) och [Installations- och konfigurationsguiden för FileMaker Server](#).
- I FileMaker Pro öppnar du varje FileMaker-databas som du vill publicera och ser till att databasen har rätt utökad behörighet aktiverad för anpassad webbpublicering. Se "Aktivera anpassad webbpublicering i en databas" på sidan 13.
- Information om hur du får åtkomst till data i FileMaker-databaser med XML finns i "Åtkomst till XML-data med Web Publishing Engine" på sidan 28.

# Kapitel 4

## Åtkomst till XML-data med Web Publishing Engine

Du kan visa och uppdatera FileMaker-data i formatet XML (Extensible Markup Language) genom att använda Web Publishing Engine. Många individer, organisationer och företag använder XML för överföring av produktinformation, transaktioner, lagerinformation och annan affärsinformation.

### Använda anpassad webbpublicering med XML

Om du känner till standard-XML kan du börja använda Web Publishing Engine så snart du har lärt dig några saker om Anpassad webbpublicering med XML, till exempel URL-syntaxen och vilka frågeparametrar som ska användas.

Genom att använda HTTP URL-förfrågningar tillsammans med FileMaker-frågekommandon och frågeparametrar kan du ställa frågor till en databas som ligger på FileMaker Server och hämta resultatet i XML-format. Du kan till exempel fråga efter alla poster som har ett visst postnummer och använda den resulterande XML-informationen på vilket sätt som helst.

Se [FileMaker Knowledge Base](#).

**Obs!** Web Publishing Engine skapar XML-data som är korrekt formaterade och kompatibla med XML 1.0-specifikationen. Mer information om kraven för korrekt XML-kod finns i XML-specifikationen på [www.w3.org](http://www.w3.org).

### Skillnader mellan Web Publishing Engine och XML-import/export i FileMaker Pro

Du kan använda XML-data med FileMaker-databaser med hjälp av Web Publishing Engine eller FileMaker Pro. Det finns dock viktiga skillnader mellan de två metoderna:

- När det gäller åtkomst av XML-data stöder Web Publishing Engine syntaxerna `fmresultset`, `FMPXMLRESULT` och `FMPXMLLAYOUT`. Vid XML-import använder FileMaker Pro syntaxen `FMPXMLRESULT` och vid export syntaxen `FMPXMLRESULT`. Se "Åtkomst till XML-data med Web Publishing Engine" på sidan 28.
- Om du vill få åtkomst till XML-data med Web Publishing Engine kan du använda en frågesträng i en URL-adress. Om du vill importera och exportera XML med FileMaker Pro använder du menykommandon och scripts i FileMaker Pro.
- Web Publishing Engine är serverbaserad och kan installeras på samma värd som FileMaker Server, eller en annan. Import och export av FileMaker Pro är skrivbordsbaserad.
- Du kan dynamiskt få åtkomst till XML-data från FileMaker-databaser genom att använda URL-förfrågningar med Web Publishing Engine. XML-exportfunktionen i FileMaker Pro genererar en särskild XML-datafil.
- Att arbeta med XML-data via Web Publishing Engine är en interaktiv åtgärd. FileMaker Pro XML-import/export är en gruppåtgärd.
- Web Publishing Engine kan hämta XML-data från en FileMaker-portal, vilket inte FileMaker Pro kan.
- Web Publishing Engine kan hämta data i ett containerfält, vilket inte FileMaker Pro kan.

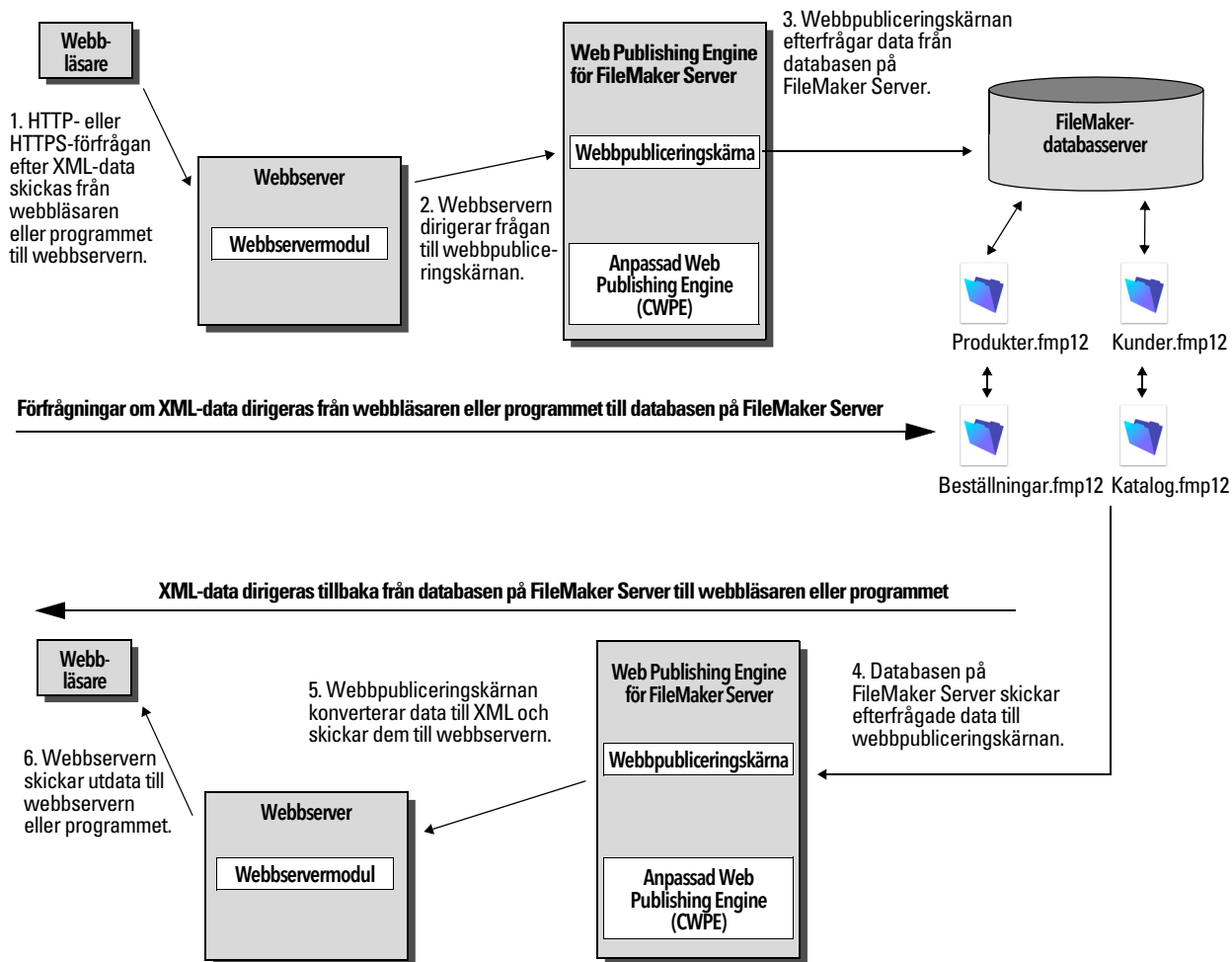


- Web Publishing Engine ger åtkomst i realtid till FileMaker-data via HTTP eller HTTPS, vilket inte FileMaker Pro gör.

**Obs!** Information om hur du importerar och exporterar data i XML-format med hjälp av FileMaker Pro finns i [FileMaker Pro Hjälp](#).

### Hur Web Publishing Engine genererar XML-data från en förfrågan

När en förfrågan om XML-data skickas till webbservern skickar Web Publishing Engine frågan till FileMaker-databasen och returnerar data som ett XML-dokument.





## Åtkomst till XML-data från Web Publishing Engine

Här nedan visas en översikt över hur du använder Web Publishing Engine för att få åtkomst till XML-data i en FileMaker-databas:

1. Kontrollera att XSLT-publicering är aktiverat i FileMaker Server Admin Console. Se [FileMaker Server Hjälp](#).

2. I FileMaker Pro öppnar du varje FileMaker-databas som du vill publicera och ser till att databasen har den utökade behörigheten fmxml aktiverad för anpassad webbpublicering i XML. Se "Aktivera anpassad webbpublicering i en databas" på sidan 13.

Om du vill vi kunna använda XML-data i en portal ställer du in vyn för databaslayouten till **Visa en post** eller **Visa som lista**. Om en användare eller ett script ändrar vyn i databaslayouten till **Visa som tabell** är endast den första relaterade posten (den första raden i portalen) åtkomlig som XML-data.

XML-data matas ut i en ordning som motsvarar den ordning i vilken fältobjekten lades till i layouten. Om du vill att dataordningen för XML ska matcha den ordning som fälten visas i på skärmen (uppfifrån och ned, vänster till höger), väljer du alla fält, grupperar dem och delar sedan upp gruppen. Den här proceduren återställer layoutordningen så att den matchar skärmordningen.

3. Skicka en HTTP- eller HTTPS-förfrågan i form av en URL-adress som anger XML-syntaxen i FileMaker, ett frågekommando och en eller fler FileMaker-frågeparametrar till Web Publishing Engine via ett HTML-formulär, en HREF-länk eller ett script i programmet eller på webbsidan. Du kan också skriva URL-adressen i en webbläsare.

Information om hur du anger URL-adressen finns i nästa avsnitt, "URL-syntax för XML-data och containerobjekt". Information om frågekommandon och frågeparametrar finns i "Efterfråga XML-data med hjälp av frågesträngar i FileMaker" på sidan 38, och kapitel 5, "Giltiga namn i XML-frågesträngar".

4. Web Publishing Engine använder den syntax du angav i URL-adressen för att generera XML-data som innehåller resultatet av din förfrågan, till exempel en grupp poster från databasen, och returnerar det till programmet eller webbläsaren.
5. I webbläsaren visas data (om programmet har en XML-tolk) eller så använder programmet informationen på ett sätt som du har angett.

## URL-syntax för XML-data och containerobjekt

I det här avsnittet beskrivs den URL-syntax du använder för att Web Publishing Engine ska kunna använda XML-data och containerobjekt från FileMaker-databaser.

### URL-syntax för XML-data

Den URL-syntax du ska använda för att Web Publishing Engine ska få åtkomst till XML-data från FileMaker-databaser är:

```
<protokoll>://<värd>[:<port>]/fmi/xml/<xml-syntax>.xml[?<frågesträng>]
```

där:

- <protokoll> kan vara protokollet HTTP eller HTTPS.
- <värd> är IP-adressen eller domännamnet för värden som webbservern är installerad på.

- `<port>` är valfritt och anger porten som webbservern använder. Om ingen port anges används standardporten för respektive protokoll (port 80 för HTTP och port 443 för HTTPS).
- `<xml-syntax>` är namnet på XML-syntaxen i FileMaker. Möjliga värden är `fmresultset`, `FMPXMLRESULT` och `FMPXMLLAYOUT`. Se "Använda syntaxen `fmresultset`" på sidan 29 och "Använda andra FileMaker XML-syntaxer" på sidan 33.
- `<frågesträng>` är en kombination av ett frågekommando och en eller fler frågeparametrar för FileMaker XML-publicering. (Kommandot `-dbnames` kräver inga parametrar.) Se "Efterfråga XML-data med hjälp av frågesträngar i FileMaker" på sidan 38, och kapitel 5, "Giltiga namn i XML-frågesträngar".

**Obs!** URL-syntaxen, inklusive namnen på frågekommando och parametrar, är skiftlägeskänslig, utom för delar av frågesträngen. Huvuddelen av URL-adressen anges med små bokstäver (gemener), förutom de två syntaxnamnen i versaler: `FMPXMLRESULT` och `FMPXMLLAYOUT`. Information om vilka regler som gäller för stora och små bokstäver i frågesträngar finns i "Riktlinjer för frågekommandon och frågeparametrar" på sidan 43.

### Exempel

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=products&-lay=sales
&-findall
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=products&-lay=sales
&-findall
```

### URL-syntax för FileMaker-containerobjekt i XML-lösningar

I ett XML-dokument som har skapats för en XML-lösning är den syntax som du använder för att referera till ett containerobjekt annorlunda för containerfält som innehåller själva objektet i databasen, jämfört med containerfält som innehåller en referens till objektet.

#### Om ett containerfält lagrar det faktiska objektet i databasen

Containerfältets `<data>`-element refererar till objektet med följande relativa URL-syntax:

```
<data>/fmi/xml/cnt/data.<extension>?<query string></data>
```

där `<extension>` är filnamnstillägget som identifierar objekttypen, till exempel JPG. Med filnamnstillägget kan MIME-typen tillåta webbläsaren att identifiera rätt typ av containerdata. Information om `<query string>` finns i föregående avsnitt, "URL-syntax för XML-data".

### Exempel

```
<data>/fmi/xml/cnt/data.jpg?-db=products&-lay=sales
&-field=product_image(1)&-recid=2</data>
```

**Obs!** I XML-informationen som har skapats för ett containerfält är värdet för frågeparametern `-field` ett fullständigt fältnamn. Talet inom parentes indikerar repetitionsnumret för containerfältet och skapas för både repeterade fält och icke-repeterade fält. Se "Syntax för ett fullständigt fältnamn" på sidan 45.

Om du vill hämta containerdata från databasen använder du följande syntax:

`<protokoll>://<värd>[:<port>]/fmi/xml/cnt/data.<tillägg>?<frågesträng>`  
Information om `<protokoll>`, `<värd>` och `<port>` finns i föregående avsnitt, "URL-syntax för XML-data".

#### Exempel

```
http://www.company.com/fmi/xml/cnt/data.jpg?-db=products&-lay=sales
&-field=product_image(1)&-recid=2
```

#### Om ett containerfält lagrar en filreferens i stället för själva objektet

Containerfältets `<data>`-element innehåller en relativ sökväg som refererar till objektet.

#### Exempel

```
<data>/images/logo.jpg</data>
```

**Obs!** Det containerobjekt som referensen gäller måste lagras i mappen i FileMaker Pro Web när posten skapas eller redigeras och sedan kopieras eller flyttas till en mapp med samma relativa sökväg i rotmappen på webbserverprogrammet. Se "Publicera innehållet i containerfält på webben" på sidan 15.

#### Om ett containerfält är tomt

Containerfältets `<data>`-element är tomt.

#### URL-textkodning

URL-adresserna för åtkomst av XML-data och containerobjekt måste kodas i formatet UTF-8 (Unicode Transformation 8 Bit). Mer information finns i "Om UTF-8-kodade data" på sidan 38.

#### Exempel

Om du vill sätta värdet för fältet "info" till fiancée kan du använda följande URL:

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=members
&-lay=relationships&-recid=2&info=fianc%C3%A9e&-edit
%C3%A9 är den URL-kodade UTF-8-motsvarigheten till tecknet é.
```

Se URL-specifikationen på [www.w3.org](http://www.w3.org).

## Åtkomst till XML-data med Web Publishing Engine

Om du vill öppna XML-data via Web Publishing Engine använder du en URL-adress som innehåller namnet på den FileMaker-syntax som ska användas, ett FileMaker-frågekommando och en eller fler FileMaker-frågeparametrar. Web Publishing Engine genererar XML-data från din databas som har utformats av en av följande typer av XML-syntax:

- **fmresultset:** Denna syntax rekommenderas för Web Publishing Engine vid åtkomst av XML-data. Den är flexibel och optimerad för enklare åtkomst till fält via namn och enklare hantering av *relatedset*-data (portal). Denna syntax är också på ett mer direkt sätt kopplad till terminologi och funktioner i FileMaker, till exempel globala lagringsalternativ och identifiering av statistikfält och beräkningsfält. Denna syntax underlättar webbpublicering genom att den är mer ordrik än syntaxen *FMPXMLRESULT*. Se "Använda syntaxen *fmresultset*" på sidan 29.
- **FMPXMLRESULT och FMPXMLLAYOUT:** Du kan också använda syntaxerna *FMPXMLRESULT* och *FMPXMLLAYOUT* med Web Publishing Engine när du vill komma åt XML-data. Om du endast vill använda en formatmall för både XML-export och anpassad webbpublicering måste du använda syntaxen *FMPXMLRESULT*. Om du vill kunna visa värdelistor och fältinformation i layouter måste du använda syntaxen *FMPXMLLAYOUT*. Se "Använda andra FileMaker XML-syntaxer" på sidan 33.

Beroende på vilken syntax du anger i URL-förfrågningen kommer Web Publishing Engine att skapa ett XML-dokument med hjälp av en av syntaxerna. Alla XML-dokument innehåller ett förvalt XML-namntryck för syntaxen. Se nästa avsnitt, "Namnuttryck för FileMaker XML". Använd någon av syntaxerna i ditt dokument eller på webbsidan när du vill visa och arbeta med FileMaker-data i XML-format.

**Obs!** XML-data som skapas i Web Publishing Engine är kodade i UTF-8-format (Unicode Transformation Format 8). Se "Om UTF-8-kodade data" på sidan 38.

### Namntryck för FileMaker XML

Unika namnuttryck i XML särskiljer olika XML-markörer efter det program de har utformats för. Om XML-dokumentet exempelvis innehåller två *<DATABASE>*-element, ett för FileMaker-baserade XML-data och ett annat för Oracle-baserade XML-data identifieras *<DATABASE>*-elementet för vart och ett av dem med hjälp av namnuttrycken.

Web Publishing Engine skapar förvalda namnuttryck för varje syntax.

För denna syntax	Används följande namnuttryck
fmresultset	xmlns="http://www.filemaker.com/xml/fmresultset"
FMPXMLRESULT	xmlns="http://www.filemaker.com/fmpxmlresult"
FMPXMLLAYOUT	xmlns="http://www.filemaker.com/fmpxmllayout"

## Felkoder i FileMaker-databaser

Web Publishing Engine genererar en felkod som motsvarar felet, i felkodselementen i början av varje XML-dokument, om det uppstod något fel när det senaste frågekommandot kördes. Värdet noll (0) returneras om det inte finns något fel.

För denna syntax	Används följande syntax
fmresultset	<error code="0"></error>
FMPXMLRESULT	<ERRORCODE>0</ERRORCODE>
FMPXMLLAYOUT	<ERRORCODE>0</ERRORCODE>

Felkodselementet i XML-dokumentet visar fel som är relaterade till databasen och frågesträngarna. Se bilaga A, "Felkoder för anpassad webbpublicering".

## Hämta dokumenttypsdefinitionerna för syntaxerna i FileMaker

Du kan hämta dokumenttypsdefinitionerna (DTD-filer) för syntaxerna i FileMaker med hjälp av en HTTP-förfrågan.

För denna syntax	Använder du följande HTTP-förfrågan
fmresultset	http://<värd>[:<port>]/fmi/xml/fmresultset.dtd
FMPXMLRESULT	http://<värd>[:<port>]/fmi/xml/FMPXMLRESULT.dtd
FMPXMLLAYOUT	http://<värd>[:<port>]/fmi/xml/FMPXMLLAYOUT.dtd

## Använda syntaxen fmresultset

XML-elementnamnen i denna syntax använder FileMaker-terminologi och lagringen av fält separeras från fälttypen. Syntaxen innehåller även funktioner för identifiering av statistikfält, beräkningsfält och globala fält.

Om du vill använda syntaxen `fmresultset` anger du följande namn på syntaxen `fmresultset` i URL-adressen som används när du efterfrågar XML-dokumentet från Web Publishing Engine:

`fmresultset.xml`

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findall
```

**Obs!** När du anger syntaxen `fmresultset` måste du använda små bokstäver (gemener).

Web Publishing Engine skapar ett XML-dokument med hjälp av syntaxen `fmresultset`. I XML-dokumentet refererar Web Publishing Engine till dokumenttypsdefinitionen för syntaxen `fmresultset` i instruktionen `<!DOCTYPE>` på den andra raden i dokumentet, omedelbart efter instruktionen `<?xml . . . ?>`. `<!DOCTYPE>`-instruktionen anger vilken URL-adress du använder för att hämta DTD-filen för syntaxen `fmresultset`.

## Beskrivning av element i syntaxen fmresultset

Syntaxen fmresultset består huvudsakligen av <datasource>-elementet, <metadata>-elementet och <resultset>-elementet.

### <datasource>-element

I syntaxen fmresultset innehåller <datasource>-elementet tabell, layout, datumformat, tidsformat, tidsstämpelformat, totalt antal och databasattribut.

- Datumformatattributet för <datasource>-elementet anger datumformatet i XML-dokumentet.

yyyy-MM-dd

där:

- MM är det tvåsiffriga värdet för månaden (01 till 12, där 01 är januari och 12 är december)
- dd är det tvåsiffriga värdet för dagen i månaden (01 till 31)
- yyyy är det fyrsiffriga värdet för året
- Tidsformatattributet för <datasource>-elementet anger tidsformatet i XML-dokumentet:

HH:mm:ss

där:

- HH är det tvåsiffriga värdet för timmen (00 till 23) i formatet med 24 timmar)
- mm är det tvåsiffriga värdet för minuter (00 till 59)
- ss är det tvåsiffriga värdet för sekunder (00 till 59)
- Attributet för tidsstämpelattributet för <datasource>-elementet kombinerar datumformatet och tidsformatet till en tidsstämpel.

yyyy-MM-dd HH:mm:ss

### <metadata>-element

Elementet <metadata> i syntaxen fmresultset innehåller en eller flera <field-definition>- och <relatedset-definition>-element. Vart och ett innehåller attribut för ett av fälten i resultatet.

Attributet <field-definition> anger:

- vare sig fältet är ett fält för automatisk inmatning ("ja" eller "nej")
- vare sig fältet är ett fält för år angivet med fyra siffror ("ja" eller "nej")
- vare sig fältet är ett globalt fält ("ja" eller "nej")
- det maximala antalet repeterade värden (max-repeat, attribut)
- det maximala antalet tillåtna tecken (max characters, attribut)
- om fältet är ett ej tomt fält ("ja" eller "nej")
- om fältet endast är för numeriska data ("ja" eller "nej")
- result ("text", "number", "date", "time", "timestamp" eller "container")
- om fältet är ett fält för tidpunkt ("ja" eller "nej")
- typ ("normal", "calculation" eller "summary")
- och fältnamnet (fullständigt vid behov)

Elementet <relatedset-definition> motsvarar en portal. Varje relaterat fält i en portal motsvaras av elementet <field-definition> som finns i <relatedset-definition>-elementet. Om det finns flera relaterade fält i en portal grupperas fältdefinitionerna för de relaterade fälten i ett enda <relatedset-definition>-element.

### <resultset>-element

Elementet <resultset> innehåller alla <record>-element som returneras som resultat av en fråga och ett attribut för det totala antalet hittade poster. Varje <record>-element innehåller fältdata för en post i resultatet – inklusive attributen mod-id och record-id för posten och <data>-elementet som innehåller data för ett fält i posten.

Varje post i en portal motsvaras av ett <record>-element i <relatedset>-elementet. Nummerattributet för <relatedset>-elementet anger antalet poster i portalen och tabellattributet anger vilken tabell som är kopplad till portalen.

## XML-data i syntaxen fmresultset

### Exempel

```
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
  <error code="0"/>
  <product build="03/29/2017" name="FileMaker Web Publishing Engine"
    version="16.0.1.0"/>
  <datasource database="art" date-format="MM/dd/yyyy" layout="web3"
    table="art" time-format="HH:mm:ss" timestamp-format="MM/dd/yyyy HH:mm:ss"
    total-count="12"/>
  <metadata>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Title" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Artist" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Style" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="length" not-empty="no" numeric-only="no"
      result="number" time-of-day="no" type="calculation"/>
    <relatedset-definition table="artlocations">
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-
        repeat="1" name="artlocations::Location" not-empty="no" numeric-
        only="no" result="text" time-of-day="no" type="normal"/>
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-
        repeat="1" name="artlocations::Date" not-empty="no" numeric-only="no"
        result="date" time-of-day="no" type="normal"/>
    </relatedset-definition>
  </metadata>
  <resultset count="1" fetch-size="1">
    <record mod-id="6" record-id="14">
      <field name="Title">
        <data>Spring in Giverny 3</data>
      </field>
      <field name="Artist">
        <data>Claude Monet</data>
      </field>
      <field name="Style">
        <data/>
      </field>
      <field name="length">
        <data>19</data>
      </field>
      <relatedset count="0" table="artlocations"/>
    </record>
  </resultset>
</fmresultset>
```



## Använda andra FileMaker XML-syntaxer

De andra FileMaker XML-syntaxerna innehåller information om fälttyper, värdelistor och layouter. FMPXMLRESULT fungerar på samma sätt som fmresultset. Om du vill kunna visa värdelistor och fältinformation i layouter måste du använda syntaxen FMPXMLLAYOUT. Syntaxerna FMPXMLRESULT och FMPXMLLAYOUT är mer kompakta vid datautbyte.

Om du vill använda syntaxen FMPXMLRESULT anger du följande syntaxnamn i den URL-adress du använder för att efterfråga XML-dokumentet från Web Publishing Engine:

FMPXMLRESULT.xml

### Exempel

```
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=employees&-lay=family  
&-findall
```

Om du vill använda syntaxen FMPXMLLAYOUT anger du följande syntaxnamn med frågekommandot `-view` i den URL-adress du använder för att efterfråga XML-dokumentet från Web Publishing Engine:

FMPXMLLAYOUT.xml

### Exempel

```
http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?-db=employees&-lay=family  
&-view
```

**Obs!** När du anger syntaxerna FMPXMLRESULT och FMPXMLLAYOUT måste du skriva namnet med stora bokstäver.

I XML-dokumentet refererar Web Publishing Engine dokumenttypsdefinitionen för syntaxen fmresultset i `<!DOCTYPE>`-instruktionen på den andra raden i dokumentet, omedelbart efter `<?xml . . . ?>`-instruktionen. `<!DOCTYPE>`-instruktionen anger URL-adressen du använder för att hämta DTD-filen för syntaxen.

## Beskrivning av element i syntaxen FMPXMLRESULT

I syntaxen FMPXMLRESULT innehåller elementet `<DATABASE>` attributen NAME, RECORDS, DATEFORMAT, LAYOUT och TIMEFORMAT.

Attributet DATEFORMAT i `<DATABASE>`-elementet specificerar datumformatet i XML-dokumentet. Attributet TIMEFORMAT i `<DATABASE>`-elementet specificerar tidsformatet i XML-dokumentet. Datum- och tidsformaten för syntaxerna FMPXMLRESULT och fmresultset är desamma. Se "Beskrivning av element i syntaxen fmresultset" på sidan 30.

Elementet `<METADATA>` i FMPXMLRESULT-syntaxen innehåller ett eller flera `<FIELD>`-element där vart och ett innehåller information om något av fälten/kolumnerna i resultatet, inklusive fältets namn som det definieras i databasen, fälttypen, möjliga Ja/Nej-värden i tomma fält (EMPTYOK-attribut) och det maximala antalet upprepade värden (MAXREPEAT-attribut). Giltiga värden för fälttyper är TEXT, NUMBER, DATE, TIME, TIMESTAMP och CONTAINER.

Elementet `<RESULTSET>` innehåller alla `<ROW>`-element som returneras som resultat på en fråga och ett attribut för det totala antalet hittade poster. Varje `<ROW>`-element innehåller fält-/kolumndata för en rad i resultatet. Dessa data innehåller RECORDID och MODID för raden (se "Frågeparametern –modid (ändrings-ID)" på sidan 56) och `<COL>`-elementet. Elementet `<COL>` innehåller data för ett fält eller en kolumn på raden där flera `<DATA>`-element motsvarar ett av värdena i ett repeterat fält eller ett portalfält.

## XML-data i syntaxen FMPXMLRESULT

### Exempel

```
<FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="03/29/2017" NAME="FileMaker Web Publishing Engine"
    VERSION="16.0.1.0"/>
  <DATABASE DATEFORMAT="MM/dd/yyyy" LAYOUT="web" NAME="art" RECORDS="12"
    TIMEFORMAT="HH:mm:ss"/>
  <METADATA>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Title" TYPE="TEXT"/>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Artist" TYPE="TEXT"/>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Image" TYPE="CONTAINER"/>
  </METADATA>
  <RESULTSET FOUND="1">
    <ROW MODID="7" RECORDID="4">
      <COL>
        <DATA>Village Market</DATA>
      </COL>
      <COL>
        <DATA>Camille Pissarro</DATA>
      </COL>
      <COL>
        <DATA>/fmi/xml/cnt/Untitled.pct?-db=art&-lay=web&-recid=4
          &-field=Image(1)
        </DATA>
      </COL>
    </ROW>
  </RESULTSET>
</FMPXMLRESULT>
```

Ordningsföljden på `<COL>`-elementen motsvarar ordningsföljden på `<FIELD>`-elementen i `<METADATA>`-elementet. Exempel: Där fälten "Title" och "Artist" visas i `<METADATA>`-elementet, visas "Village Market" och sedan "Camille Pissarro" i samma ordning i `<RESULTSET>`- och `<ROW>`-elementen.

## Beskrivning av element i syntaxen FMPXMLLAYOUT

I syntaxen FMPXMLLAYOUT innehåller elementet <LAYOUT> namnet på layouten, namnet på databasen och <FIELD>-elementen för varje fält som hittas i motsvarande layout i databasen. I alla <FIELD>-element beskrivs stilen på fältet. Det innehåller dessutom VALUELIST-attributet för alla tillhörande värdelistor i fältet.

Elementet <VALUELISTS> innehåller ett eller flera <VALUELIST>-element för alla värdelistor som hittas i layouten – där var och en innehåller namnet på värdelistan och ett <VALUE>-element för vart och ett av värdena i listan.

Beroende på vilka alternativ du har valt i dialogrutan **Ange fält i värdelista** i FileMaker-databasen, innehåller elementet <VALUE> ett DISPLAY-attribut som innehåller värde endast i det första fältet, endast i det andra fältet eller i båda fälten i en värdelista. Antal t.ex. att det första fältet i en värdelista lagrar konststilens ID-nummer (t.ex. "100") och det andra fältet visar konststilens tillhörande namn (t.ex. "Impressionism"). Här är en sammanfattning av innehållet i DISPLAY-attributet när de olika kombinationerna av alternativ har valts i dialogrutan **Ange fält i värdelista**:

- Om **Visa även värden från det andra fältet** inte har markerats, innehåller DISPLAY-attributet endast värdet i det första fältet från en värdelista.

### Exempel

DISPLAY-attributet innehåller endast konststilens ID-nummer:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100">100</VALUE>
    <VALUE DISPLAY="101">101</VALUE>
    <VALUE DISPLAY="102">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Om både **Visa även värden från det andra fältet** och **Visa värden endast från det andra fältet** har markerats, innehåller DISPLAY-attributet värde endast i det andra fältet.

### Exempel

DISPLAY-attributet innehåller endast konststilens namn:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="Impressionism">100</VALUE>
    <VALUE DISPLAY="Kubism">101</VALUE>
    <VALUE DISPLAY="Abstrakt">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Om både **Visa även värden från det andra fältet** och **Visa värden endast från det andra fältet** har markerats, innehåller DISPLAY-attributet värden i båda fälten i en värdelista.

### Exempel

DISPLAY-attributet innehåller både konststilens ID-nummer och konststilens namn:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100 Impressionism">100</VALUE>
    <VALUE DISPLAY="101 Kubism">101</VALUE>
    <VALUE DISPLAY="102 Abstrakt">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

För fält för datum, tid och tidsstämpel formateras data i värdelistor med hjälp av "fm"-formatet för fälttypen. "fm"-formaten är MM/dd/yyyy för datum, HH:mm:ss för tid och MM/dd/yyyy HH:mm:ss för tidsstämpel. Om t.ex. en värdelista med födelsedagar används till en popup-meny i ett "födelsedagsfält" i en layout och om fältet "födelsedag" är av typen datum, så kommer alla utdatavärden för den värdelistan att vara i datumformatet "fm".

**Obs!** Om två fält med olika fälttyper i en layout delar samma värdelistan bestämmer det första fältets typ formatet för de data som finns i värdelistan.

## XML-data i syntaxen FMPXMLLAYOUT

### Exempel

```
<FMPXMLLAYOUT xmlns="http://www.filemaker.com/fmpxmllayout">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="03/29/2017" NAME="FileMaker Web Publishing Engine"
    VERSION="16.0.1.0"/>
  <LAYOUT DATABASE="art" NAME="web2">
    <FIELD NAME="Title">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Artist">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Image">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="artlocations::Location">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="artlocations::Date">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Style">
      <STYLE TYPE="POPUPMENU" VALUELIST="style"/>
    </FIELD>
  </LAYOUT>
  <VALUELISTS>
    <VALUELIST NAME="style">
      <VALUE DISPLAY="Impressionist">Impressionist</VALUE>
      <VALUE DISPLAY="Modern">Modern</VALUE>
      <VALUE DISPLAY="Abstract">Abstrakt</VALUE>
    </VALUELIST>
  </VALUELISTS>
</FMPXMLLAYOUT>
```

## Om UTF-8-kodade data

Alla XML-data som skapas i Web Publishing Engine är kodade i UTF-8-format. I detta format komprimeras data från 16-bitars Unicode-format (standard) till 8-bitarsformatet för ASCII-tecken. XML-tolkar krävs för Unicode- och UTF-8-kodning.

UTF-8-kodning innehåller en direkt motsvarighet av värdena mellan 0 och 127 för standarduppsättningen av ASCII-tecken samt tillhandahåller flerbytekodning för Unicode-tecken med högre värden.

**Obs!** Se till att använda en webbläsare eller ett textredigeringsprogram som stöder filer i UTF-8-format.

Kodningsformatet UTF-8 innehåller följande funktioner:

- Alla ASCII-tecken är UTF-8-tecken på 1 byte. En giltig ASCII-sträng är en giltig UTF-8-sträng.
- Alla icke-ASCII-tecken (alla tecken ur det övre ASCII-området) är en del av ett flerbytetecken.
- Ett UTF-8-teckens första byte indikerar antalet ytterligare byte i tecknet.
- Den första byten i ett flerbytetecken kan lätt urskiljas från den efterföljande byten, vilket gör det enkelt att hitta starttecknet i en godtycklig position i en dataström.
- Det är enkelt att konvertera mellan UTF-8 och Unicode.
- UTF-8-kodningen är relativt kompakt och för text med en stor andel ASCII-tecken är den kompaktare än Unicode. I värsta fall är en UTF-8-sträng 50 % större än motsvarande Unicode-sträng.

## Efterfråga XML-data med hjälp av frågesträngar i FileMaker

Om du vill efterfråga XML-data från en FileMaker-databas använder du frågekommandon och frågeparametrar i en frågesträng i FileMaker.

### Exempel

Du kan använda frågekommandot `-findall` i följande frågesträng i en URL när du vill göra en förfrågan om en lista över alla produkter i en FileMaker-databas med namnet "products":

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=products  
&-lay=sales&-findall
```

En frågesträng får endast innehålla ett frågekommando, till exempel `-new`. För de flesta frågekommandon krävs också olika frågeparametrar i frågesträngen. För alla frågekommandon förutom `-dbnames` krävs exempelvis parametern `-db`, som anger vilken databas frågan ska ställas till.

Du kan också använda frågekommandon och parametrar i en URL-adress.

Det här avsnittet innehåller en sammanfattning av frågekommandona och frågeparametrarna i FileMaker. Information om hur du använder dem i en frågesträng finns i "Giltiga namn i XML-frågesträngar" på sidan 42.

Använd det här frågekommandot	Om du vill utföra följande kommando
<code>-dbnames</code>	Hämta namnen på alla databaser som finns på servern och som är webbdelade.
<code>-delete</code>	Radera post.
<code>-dup</code>	Duplicera post.
<code>-edit</code>	Redigera post.
<code>-find</code>	Sök post(er).
<code>-findall</code>	Sök efter alla poster.
<code>-findany</code>	Slumpmässig sökning efter post.
<code>-findquery</code>	Utför komplex eller sammansatt sökpost.
<code>-layoutnames</code>	Hämta namnen på alla tillgängliga layouter för databaser som finns på servern och som är webbdelade.
<code>-new</code>	Lägg till ny post.
<code>-scriptnames</code>	Hämta namnen på alla tillgängliga scripts för en databas som finns på servern och som är webbdelad.
<code>-view</code>	Hämta layoutinformation från en databas om syntaxen <code>FMPXMLLAYOUT</code> anges. Hämtar avsnittet <code>&lt;metadata&gt;</code> i XML-dokumentet och en tom resultatuppsättning om syntaxen <code>fmresultset</code> eller <code>FMPXMLRESULT</code> anges.

Använd följande frågeparametrar	Med följande frågekommandon
<code>-db</code> (databasnamn)	Obligatoriskt tillsammans med alla frågekommandon utom <code>-dbnames</code>
<code>-delete.related</code>	Valfritt tillsammans med <code>-edit</code>
<code>-field</code>	Krävs om du vill ange ett fält i en URL-adress för containerförfrågningar. Se "URL-syntax för FileMaker-containerobjekt i XML-lösningar" på sidan 26.
fältnamn	Minst ett fältnamn krävs för <code>-edit</code> . Valfritt tillsammans med <code>-find</code> . Se "Frågeparametern fältnamn (namn på annat fält än containerfält)" på sidan 52.
fältnamn.op (operator)	Valfritt tillsammans med <code>-find</code>
<code>-lay</code> (layoutnamn)	Obligatoriskt med alla frågekommandon utom <code>-dbnames</code> , <code>-layoutnames</code> och <code>-scriptnames</code>
<code>-lay.response</code> (växla layout för XML-svar)	Valfritt med alla frågekommandon utom <code>-dbnames</code> , <code>-layoutnames</code> och <code>-scriptnames</code>
<code>-lop</code> (logisk operator)	Valfritt tillsammans med <code>-find</code>
<code>-max</code> (maximalt antal poster)	Valfritt tillsammans med <code>-find</code> , <code>-findall</code> och <code>-findquery</code>
<code>-modid</code> (ändrings-id)	Valfritt tillsammans med <code>-edit</code>
<code>-query</code>	Krävs med den sammanfattande sökposten <code>-findquery</code>
<code>-recid</code> (post-id)	Krävs tillsammans med <code>-edit</code> , <code>-delete</code> , <code>-dup</code> . Valfritt tillsammans med <code>-find</code>
<code>-relatedsets.filter</code>	Valfritt tillsammans med <code>-find</code> , <code>-findall</code> , <code>-findany</code> , <code>-edit</code> , <code>-new</code> , <code>-dup</code> och <code>-findquery</code>
<code>-relatedsets.max</code>	Valfritt tillsammans med <code>-find</code> , <code>-edit</code> , <code>-new</code> , <code>-dup</code> och <code>-findquery</code>

Använd följande frågeparametrar	Med följande frågekommandon
<code>-script</code> (utför script)	Valfritt tillsammans med <code>-find</code> , <code>-findall</code> , <code>-findany</code> , <code>-new</code> , <code>-edit</code> , <code>-delete</code> , <code>-dup</code> , <code>-view</code> och <code>-findquery</code>
<code>-script.param</code> (skicka ett parametervärde till det script som anges av <code>-script</code> )	Valfritt tillsammans med <code>-script</code> och <code>-findquery</code>
<code>-script.prefind</code> (utför script före <code>-find</code> , <code>-findany</code> och <code>-findall</code> )	Valfritt tillsammans med <code>-find</code> , <code>-findany</code> , <code>-findall</code> och <code>-findquery</code>
<code>-script.prefind.param</code> (skicka ett parametervärde till det script som anges av <code>-script.prefind</code> )	Valfritt med <code>-script.prefind</code> och <code>-findquery</code>
<code>-script.presort</code> (utför script före sortering)	Valfritt tillsammans med <code>-find</code> , <code>-findall</code> och <code>-findquery</code>
<code>-script.presort.param</code> (skicka ett parametervärde till det script som anges av <code>-script.presort</code> )	Valfritt med <code>-script.presort</code> och <code>-findquery</code>
<code>-skip</code> (hoppa över poster)	Valfritt tillsammans med <code>-find</code> , <code>-findall</code> och <code>-findquery</code>
<code>-sortfield. [1-9]</code> (sortera fält)	Valfritt tillsammans med <code>-find</code> , <code>-findall</code> och <code>-findquery</code>
<code>-sortorder. [1-9]</code> (sorteringsordning)	Valfritt tillsammans med <code>-find</code> , <code>-findall</code>

## Växla layout för ett XML-svar

Med frågeparametern `-lay` anger du vilken layout du vill använda när du frågar efter XML-data. Ofta fungerar samma layout för bearbetning av data som förfrågningen resulterar i. Ibland kanske du vill söka efter data med en layout som innehåller fält som av säkerhetsskäl inte finns i någon annan layout som du vill använda för att visa resultatet. (Om du vill söka efter data i ett fält måste fältet placeras i den layout du angav i XML-förfrågningen.)

Om du vill visa ett XML-svar i en annan layout än den som används för bearbetning av XML-förfrågningen, kan du använda den valfria frågeparametern `-lay.response`.

### Exempel

Följande fråga söker efter värden som är större än 100 000 i fältet "Salary" i layouten "Budget". Resultatet visas med layouten "ExecList", som inte innehåller fältet "Salary".

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=Budget&Salary=100000&Salary.op=gt&-find&-lay.response=ExecList
```



## Bearbetning av en XML-förfrågan

Det finns flera frågeparametrar som påverkar bearbetningen av en XML-förfrågan och genereringen av ett XML-dokument.

FileMaker Server och Web Publishing Engine bearbetar en XML-fråga i följande ordning:

1. Bearbeta frågeparametern `-lay`.
2. Ange de globala fältvärden som angetts i frågan (`".global="`-delen av en URL-adress).
3. Bearbeta frågeparametern `-script.prefind`, om den har angetts.
4. Bearbeta frågekommandona, till exempel `-find` eller `-new`.
5. Bearbeta frågeparametern `-script.presort`, om den har angetts.
6. Resultatet sorteras, om sortering angavs.
7. Bearbeta frågeparametern `-script`, om den har angetts.
8. Bearbeta frågeparametern `-lay.response` så att den växlar till en annan layout, om detta har angetts.
9. XML-dokumentet genereras.

Om något av ovanstående steg genererar en felkod, stoppar bearbetningen och alla eventuella kvarvarande steg utförs inte. Eventuella tidigare steg i förfråningen utförs fortfarande.

Anta att en förfrågan tar bort den aktuella posten, sorterar posterna och sedan kör ett script. Om parametern `-sortfield` anger ett fält som inte finns, tar förfrågan bort den aktuella posten och returnerar felkod 102 ("Fältet saknas"), men scriptet utförs inte.

## Felsöka åtkomst till XML-dokument

Om du har problem med att öppna XML-dokument med Web Publishing Engine kontrollerar du att:

- Den utökade behörighetsuppsättningen i databasen är inställd för XML-anpassad webbpublicering och kopplad till ett användarkonto. Se "Aktivera anpassad webbpublicering i en databas" på sidan 13.
- Databasen öppnas i komponenten databasserver av FileMaker Server-installationen och öppnas av FileMaker Server. Se [FileMaker Server Hjälp](#).
- Kontonamnet och lösenordet till databasen är korrekta.
- Kontrollera att webbserverkomponenten i FileMaker Server-installationen körs.
- Komponenten Web Publishing Engine i FileMaker Server-installationen körs.
- XML-publicering är aktiverad i komponenten Web Publishing Engine. Se [FileMaker Server Hjälp](#).

# Kapitel 5

## Giltiga namn i XML-frågesträngar

Det här kapitlet innehåller giltiga namn på frågekommandon och frågeparametrar som du kan använda i en frågesträng för att hämta FileMaker-data med hjälp av Web Publishing Engine.

### Frågekommandon och frågeparametrar

Listan nedan innehåller namnen på alla frågekommandon och frågeparametrar:

Frågekommandon	Frågeparametrar
-dbnames (Se sidan 47.)	-db (Se sidan 51.)
-delete (Se sidan 47.)	-field (Se sidan 52.)
-dup (Se sidan 48.)	fältnamn (Se sidan 52.)
-edit (Se sidan 48.)	fältnamn.op (Se sidan 53.)
-find, -findall, -findany (Se sidan 49.)	-lay (Se sidan 54.)
-findquery (Se sidan 49.)	-lay.response (Se sidan 54.)
-layoutnames (Se sidan 50.)	-lop (Se sidan 55.)
-new (Se sidan 50.)	-max (Se sidan 55.)
-scriptnames (Se sidan 51.)	-modid (Se sidan 56.)
-view (Se sidan 51.)	-query (Se sidan 56.)
	-recid (Se sidan 57.)
	-relatedsets.filter (Se sidan 58.)
	-relatedsets.max (Se sidan 59.)
	-script (Se sidan 59.)
	-script.param (Se sidan 60.)
	-script.prefind (Se sidan 60.)
	-script.prefind.param (Se sidan 60.)
	-script.presort (Se sidan 61.)
	-script.presort.param (Se sidan 61.)
	-skip (Se sidan 62.)
	-sortfield. [1-9] (Se sidan 62.)
	-sortorder. [1-9] (Se sidan 63.)

**Viktigt!** Parametern `-lay`, som används för att ange en layout, krävs med alla frågekommandon utom `-dbnames`, `-layoutnames` och `-scriptnames`.

## Riktlinjer för frågekommandon och frågeparametrar

Tänk på följande när du använder frågekommandon och frågeparametrar i en frågesträng:

- En frågesträng måste innehålla ett frågekommando, varken fler eller färre. En frågesträng kan till exempel innehålla `-new` om du vill lägga till en ny post men den kan inte innehålla `-new` och `-edit` i samma frågesträng.
- För de flesta frågekommandon krävs diverse frågeparametrar i frågesträngen. För alla frågekommandon förutom `-dbnames` krävs exempelvis parametern `-db`, som anger vilken databas frågan ska ställas till. En tabell över vilka parametrar som krävs finns i "Efterfråga XML-data med hjälp av frågesträngar i FileMaker" på sidan 38.
- För frågeparametrar och fältnamn anger du det specifika värde du vill använda, till exempel `-db=employees`. För frågekommandon ska du inte ange tecknet "=" eller ett värde efter kommandonamnet, till exempel `-findall`.
- Web Publishing Engine konverterar alla reserverade ord till små bokstäver, inklusive frågekommandon, frågeparametrar och kommandovärden när specifika värden förväntas (till exempel: `-lop=and`, `-lop=or`, `-sortorder=ascend`, `-sortorder=descend`, `-max=all`).
- Databasnamn, layoutnamn och fältnamn som används i frågesträngar är inte skiftlägeskänsliga. Du kan till exempel skriva `-lay=mylayout` när du vill ange layoutnamnet `MyLayout`.
- Vi rekommenderar att du inte använder punkter och parenteser i fältnamn. I vissa fall kan fältnamn med punkter fungera, men fältnamn med följande undantag kan aldrig användas:
  - Punkten kan inte följas av en siffra. Exemplet `myfield.9` är ett ogiltigt fältnamn.
  - Punkten kan inte följas av textsträngen `op` (de två bokstäverna "op"). Exemplet `myfield.op` är ett ogiltigt fältnamn.
  - Punkten kan inte följas av textsträngen `global` (ordet "global"). Exemplet `myfield.global` är ett ogiltigt fältnamn.Fältnamn som innehåller något av dessa undantag kan inte nås via XML med hjälp av en http-fråga. Dessa ord är reserverade för post-ID enligt beskrivningen i avsnittet "Syntax för ett fullständigt fältnamn" på sidan 45.
- För kommandot `-find` är värdet i ett fält inte skiftlägeskänsligt. Du kan exempelvis använda `Field1=Blue` eller `Field1=blue`. För kommandona `-new` och `-edit` bevaras de stora och små bokstäver du använder i värdet för ett fält i databasen, på samma sätt som du skrev i frågesträngen. Exempel, `LastName=Doe`.

## Tolkning av frågekommandon

Web Publishing Engine tolkar frågekommandon i följande ordning och slutar tolka XML-frågor när det första felet upptäcks. Om en felkod returneras motsvarar den returnerade felkoden det första felet som påträffas.

### 1. Har frågan ett kommando och är frågekommandot giltigt?

Det är ett fel om frågan saknar kommandot eller om den använder ett okänt kommando.

#### Exempel

```
-database
```

### 2. Har frågan två kommandon?

#### Exempel

```
-find&-edit
```

### 3. Har frågan ett ogiltigt värde för ett kommando eller en parameter?

#### Exempel

```
-lop=amd
```

### 4. Saknar frågan den obligatoriska databasnamnsparametern (-db)?

### 5. Saknar frågan den obligatoriska layoutnamnsparametern (-lay)?

### 6. Har frågan en ogiltig sortering?

### 7. Har frågan ogiltiga fältparametrar?

**Obs!** Om en fråga innehåller giltig men irrelevant information, bearbetas frågan utan att något fel uppstår. Om du till exempel anger parametern `-lop` i ett `-delete`-kommando, ignoreras parametern `-lop` eftersom parametern inte gör frågan ogiltig eller flertydig.

Mer information om specifika returnerade felkoder finns i bilaga A, "Felkoder för anpassad webbpublicering".

## Syntax för ett fullständigt fältnamn

Ett fullständigt fältnamn identifierar en exakt förekomst av ett fält. Eftersom fält med vanliga namn kan baseras på olika tabeller, måste du använda fullständiga namn för att undvika fel.

Syntax för ett fullständigt fältnamn är:

```
tabellnamn::fältnamn(repetitionsnummer).post-id
```

där:

- `tabellnamn` är namnet på tabellen som innehåller fältet. Tabellnamnet krävs bara om fältet inte finns i den underliggande tabellen i den layout som anges i frågesträngen.
- `fältnamn(repetitionsnummer)` är värdet i ett repeterat fält och krävs endast för repeterade fält. Repetitionsnumret börjar räkna från siffran 1. `fältnamn(2)` hänvisar exempelvis till det andra värdet i det repeterade fältet. Om du inte anger ett repetitionsnummer för ett repeterat fält används det första värdet i det repeterade fältet. Repetitionsnumret krävs för frågekommandona `-new` och `-edit` när de hanterar repeterade fält, men inte för kommandot `-find`.
- `post-id` är postens ID och krävs endast om du vill lägga till eller redigera poster i portalfält med hjälp av en frågesträng. Se avsnitten "Lägga till poster i en portal", och "Redigera poster i en portal". Post-id krävs för frågekommandona `-new` och `-edit` när de hanterar portalfält, men krävs inte för kommandot `-find`.

**Obs!** För att fält ska kunna användas måste de vara placerade i den layout du har angett i frågesträngen.

## Använda frågekommandon med portalfält

I följande avsnitt finns beskrivningar om hur frågekommandon fungerar med portalfält.

### Lägga till poster i en portal

Om du vill lägga till en ny post i en portal samtidigt som du lägger till en överordnad post, använder du frågekommandot `-new` och gör följande i frågesträngen för förfrågan:

- Använd det fullständiga fältnamnet för det relaterade portalfältet.
- Ange 0 som post-ID efter namnet på det relaterade portalfältet.
- Ange minst ett av fälten för den överordnade posten innan du anger det relaterade portalfältet.
- Ange data för det matchande fältet (nyckelfältet) i den överordnade posten.

### Exempel

Följande URL:er lägger till en ny överordnad anställningspost för John Doe och en ny relaterad post för Jane i portalen på samma gång. Namnet på den relaterade tabellen är Dependents och namnet på det relaterade fältet i portalen är Names. I det matchande fältet, ID, sparas ett anställnings-ID.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family  
&FirstName=John&LastName=Doe&ID=9756&Dependents::Names.0=Jane&-new
```

**Obs!** Du kan enbart lägga till en relaterad post till en portal per anrop.

## Redigera poster i en portal

Om du vill redigera en eller flera poster i en portal använder du kommandot `-edit` och ett post-ID för att ange den överordnade post som innehåller de portalposter du vill redigera. Ange den portalpost du vill redigera genom att ange dess post-ID i ett fullständigt fältnamn. Du kan bestämma ett post-ID från postens ID-attribut i elementet `<record>` i elementet `<relatedset>` i XML-informationen. Se "Använda syntaxen `fmresultset`" på sidan 29.

### Exempel

Följande URL redigerar en post i en portal där den överordnade posten har post-ID 1001. "Dependents" är namnet på den relaterade tabellen, "Names" är namnet på det relaterade fältet i portalen och 2 i "Names.2" är en portalposts post-ID.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.2=Kevin&-edit
```

Så här kan du använda en förfrågan för att redigera flera portalposter via den överordnade posten:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.2=Kevin&Dependents::Names.5=Susan&-edit
```

Du kan också använda kommandot `-edit` och ange 0 som portalpostens ID när du vill lägga till en ny relaterad post i portalen för en befintlig överordnad post.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.0=Timothy&-edit
```

## Radera portalposter

Om du vill radera portalposter använder du parametern `-delete.related` tillsammans med kommandot `-edit` i stället för att använda kommandot `-delete`.

### Exempel

Följande URL tar t.ex. bort posten "1001" från tabellen "employees":

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&-delete
```

Följande URL-adress tar bort en portalpost med post-ID "3" från den relaterade tabellen "Dependents" med det överordnade ID-numret "1001".

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&-delete.related=Dependents.3&-edit
```

Se "Frågeparametern `-delete.related` (radera portalposter)" på sidan 52.

## Fråga portalfält

I en lösning som har många relaterade poster kan sökning och sortering av portalposter vara tidskrävande. För att begränsa antalet poster och rader som visas i ett sökresultat kan du använda parametrarna `-relatedsets.filter` och `-relatedsets.max` med sökförfrågningar. Se "Frågeparametern `-relatedsets.filter` (filtrera portalposter)" på sidan 58 och "Frågeparametern `-relatedsets.max` (begränsa portalposter)" på sidan 59.

## Syntax för globala fält

Syntaxen för globala fält är:

```
table-name::field-name(repetition-number).global
```

där `global` anger att ett fält använder global lagring. Information om tabellnamn och fältnamn(repetitionsnummer) finns i "Syntax för ett fullständigt fältnamn" på sidan 45. Information om fält som använder global lagring finns i [FileMaker Pro Hjälp](#).

Du måste använda syntaxen `.global` om du vill identifiera ett globalt fält i en frågesträng. Web Publishing Engine anger parametervärden för globala fält innan frågekommandot utförs eller några andra parametervärden ställs in i frågesträngen. För direkta XML-förfrågningar upphör de globala värden att gälla omedelbart efter att förfrågan har gjorts.

Om du inte identifierar ett globalt fält i en frågesträng med hjälp av syntaxen `.global`, utvärderar Web Publishing Engine det globala fältet tillsammans med återstoden av frågesträngen, utan att först ange det globala fältvärdet.

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&Country.global=USA&-recid=1&-edit
```

## Referens för frågekommando

Det här avsnittet innehåller information om de frågekommandon som är tillgängliga för XML-förfrågningar.

### Frågekommandot `-dbnames` (databasnamn)

Hämtar namnen på alla databaser som FileMaker Server är värd för och som är aktiverade för anpassad webbpublicering med XML.

**Obligatoriska frågeparametrar:** (inga)

### Exempel

Om du vill hämta databasnamnen:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-dbnames
```

### Frågekommandot `-delete` (radera post)

Tar bort den post som anges med parametern `-recid`.

**Obligatoriska frågeparametrar:** `-db`, `-lay`, `-recid`

**Valfri frågeparameter:** `-script`

### Exempel

Så här raderar du en post:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-recid=4&-delete
```

### Frågekommandot -dup (duplicera post)

Duplicerar den post som anges med parametern -recid.

**Obligatoriska frågeparametrar:** -db, -lay, -recid

**Valfri frågeparameter:** -script

#### Exempel

Om du vill duplicera den angivna posten:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-recid=14&-dup
```

### Frågekommandot -edit (redigera post)

Uppdaterar posten som anges med parametern -recid och fyller fälten med innehållet i ett fältnamn/värdepar. Parametern -recid anger vilken post som ska ändras.

**Obligatoriska frågeparametrar:** -db, -lay, -recid, ett eller flera fältnamn

**Valfri frågeparameter:** -modid, -script, fältnamn

**Obs!** Kommandot -edit kan användas för att redigera poster i en portal. Se "Redigera poster i en portal" på sidan 46.

#### Exempel

Så här ändrar du en post:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-recid=13&Country=USA&-edit
```



## Frågekommandona -find, -findall och -findany (sök efter poster)

Sänder ett sökanrop med definierade villkor.

**Obligatoriska frågeparametrar:** -db, -lay

**Valfria frågeparametrar:** -recid, -lop, -op, -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort, fältnamn

### Exempel

Om du vill söka efter en post med hjälp av ett fältnamn:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=family&Country=USA&-find
```

Om du vill söka efter en post med hjälp av post-ID:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family  
&-recid=427&-find
```

Om du vill söka efter alla poster i databasen använder du -findall:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family  
&-findall
```

Om du vill söka efter en post slumpmässigt använder du -findany:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family  
&-findany
```

### Kommentar

- Att ange att fältnamn flera gånger i en enda fråga stöds inte. FileMaker Server tolkar alla värden men använder endast det senast tolkade.
- När du använder kommandot -findall ska du undvika att låta datorns minne bli överbelastat av problem genom att ange maximalt antal returnerade poster per sida med hjälp av frågeparametern -max.

## Frågekommando -findquery (sammanfattad sökning)

Skickar en sökfråga med hjälp av flera sökposter och fråga för att utelämna poster.

**Obligatoriska frågeparametrar:** -db, -lay, -query

**Valfria frågeparametrar:** -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort

### Exempel

Sök poster med katter eller hundar som inte heter "Fluffy":

```
http://host/fmi/xml/fmresultset.xml?-db=vetclinic&-lay=animals  
&-query=(q1);(q2);!(q3)&-q1=typeofanimal&-q1.value=Cat&-q2=typeofanimal  
&-q2.value=Dog&-q3=name&-q3.value=Fluffy&-findquery
```

### Använd kommandot `-findquery` för sammansatta sökningar

Ett `-findquery`-uttryck består av fyra delar, i följande ordning:

- Parametern `-query`.
- Frågedeklarationerna som består av deklarationen med frågeidentifieraren och frågebearbetningar.
- Sökfältet och värdedefinitioner för varje frågeidentifierare.
  - Definiera frågeidentifierare. En frågeidentifierare är bokstaven "q" följt av ett nummer. Till exempel: `-q1`
  - Definiera värden för frågeidentifierare med parametern. Till exempel:  
`-q1.value=fieldvalue`
  - Definiera frågeidentifieraroperatorer genom att inkludera dem som en del i uttrycket `fieldvalue`. Om du till exempel vill använda en asterisk som operatör "begins with" (börjar med): `-q1.value=fieldvalue*`
- Kommandot `-findquery` i slutet av det fullständiga uttrycket.

För information om att använda parametern `-query`, se "Frågeparametern `-query` (sammanfattad sökning)" på sidan 56.

### Frågekommandot `-layoutnames (layoutnamn)`

Hämtar namnen på alla tillgängliga layouter för en angiven databas som FileMaker Server är värd för och som är aktiverad för anpassad webbpublicering med XML.

**Obligatoriska frågeparametrar:** `-db`

#### Exempel

Om du vill hämta namnen på tillgängliga layouter:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-layoutnames
```

### Frågekommandot `-new (ny post)`

Skapar en ny post och fyller den med innehållet i ett fältnamn/värdepar.

**Obligatoriska frågeparametrar:** `-db`, `-lay`

**Valfri frågeparameter:** ett eller flera fältnamn, `-script`

**Obs!** Information om hur du inkluderar nya data för en portal finns i "Lägga till poster i en portal" på sidan 45.

#### Exempel

Så här lägger du till en ny post:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&Country=Australia&-new
```

### Frågekommandot -scriptnames (scriptnamn)

Hämtar namnen på alla tillgängliga scripts för en angiven databas som FileMaker Server är värd för och som är aktiverad för anpassad webbpublicering med XML.

**Obligatoriska frågeparametrar:** -db

#### Exempel

Om du vill hämta namnen på alla scripts:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-scriptnames
```

### Frågekommandot -view (visa layoutinformation)

Om syntaxen FMPXMLLAYOUT har angivits hämtas layoutinformation från en databas. Den visas sedan i syntaxen FMPXMLLAYOUT. Om du anger datasyntaxen (fmresultset eller FMPXMLRESULT) hämtas metadataavsnittet i XML-dokumentet samt en tom resultatuppsättning.

**Obligatoriska frågeparametrar:** -db, -lay

**Valfri frågeparameter:** -script

#### Exempel

Om du vill hämta layoutinformation:

```
http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?-db=employees  
&-lay=departments&-view
```

Om du vill hämta metadatainformation:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-view
```

## Referens för frågeparameter

Det här avsnittet innehåller information om de frågeparametrar som är tillgängliga för XML-förfrågningar.

### Frågeparametern -db (databasnamn)

Anger i vilken databas frågekommandot ska användas.

**Värdet är:** Namnet på databasen, förutom eventuell filändelse.

**Obs!** När du anger namnet på databasen för parametern -db i en frågesträng, ska du inte ta med filnamnsändelsen. Det riktiga filnamnet på databasen kan innehålla en filändelse, men filändelser är inte tillåtna som värden för parametern -db.

**Krävs för:** Alla frågekommandon förutom -dbnames

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-findall
```

### Frågeparametern `-delete.related` (radera portalposter)

Raderar en post från ett portalfält.

**Kan användas med:** frågekommandot `-edit`

**Kräver:** Ett relaterat tabellnamn och ett post-ID

#### Exempel

Följande tar bort en portalpost med post-ID "20" från den relaterade tabellen "jobtable" med det överordnade ID-numret "7".

```
http://host/fmi/xml/fmresultset.xml?-db=career&-lay=applications&-recid=7
&-delete.related=jobtable.20&-edit
```

### Frågeparametern `-field` (namn på containerfält)

Anger namnet på ett containerfält.

**Krävs för:** Förfrågningar om data i ett containerfält

Se "URL-syntax för XML-data och containerobjekt" på sidan 25.

### Frågeparametern `fältnamn` (namn på annat fält än containerfält)

Fältnamn styr villkoren för frågekommandot `-find`. Du kan även använda fältnamn när du vill ändra innehållet i en post. Om du måste ange ett värde för ett annat fält än ett containerfält för ett frågekommando eller en frågeparameter, skriver du fältnamnet utan bindestreck (`-`) som namndelen i ett namn/värdepar.

**Namnet är:** namnet på fältet i FileMaker-databasen. Om fältet inte finns i den underliggande tabellen i den layout som anges i frågesträngen, måste du ange det fullständiga fältnamnet.

Vi rekommenderar att du inte använder punkter och parenteser i fältnamn. i vissa fall kan fältnamn med punkter fungera, men fältnamn med följande undantag kan aldrig användas:

- Punkten kan inte följas av en siffra. Exemplet `myfield.9` är ett ogiltigt fältnamn.
- Punkten kan inte följas av textsträngen `op` (de två bokstäverna "op"). Exemplet `myfield.op` är ett ogiltigt fältnamn.
- Punkten kan inte följas av textsträngen `global` (ordet "global"). Exemplet `myfield.global` är ett ogiltigt fältnamn.

Fältnamn som innehåller något av dessa undantag kan inte nås via XML med hjälp av en http-fråga. Användningen av punkter i fältnamn är reserverade för post-ID enligt beskrivningen i "Syntax för ett fullständigt fältnamn" på sidan 45.

**Värdet är:** För frågekommandona `-new` och `-edit` anger du det värde du vill spara i fältet i den aktuella posten. För `-find`-frågekommandon anger du det värde du vill söka efter i fältet. När du anger värdet för ett datumfält, ett tidsfält eller ett tidsstämpelfält måste du ange värdet enligt formatet "fm" för den fälttypen. "fm"-formaten är MM/DD/YYYY för datum, HH:mm:ss för tid och MM/DD/YYYY hh:mm:ss för tidsstämpel.

**Krävs för:** frågekommandot `-edit`

**Kan användas med:** Frågekommandona `-new` och `-find`.

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-op=eq&FirstName=Sam&-max=1&-find
```

**Obs!** Att ange att fältnamn flera gånger i en enda fråga stöds inte. FileMaker Server tolkar alla värden men använder endast det senast tolkade.

### Frågeparametern fältnamn.op (jämförelseoperator)

Anger vilken jämförelseoperator som ska användas på det fältnamn som föregår operatoren. Jämförelseoperatorerna används med frågekommandot `-find`.

**Värdet är:** den operator du vill använda. Följande giltiga operatörer finns:

Nyckelord	FileMaker Pro, motsvarande operator
eq	=ord
cn	*ord*
bw	ord*
ew	*ord
gt	> ord
gte	>= ord
lt	< ord
lte	<= ord
neq	uteslut, ord

**Kan användas med:** frågekommandot `-find`

**Kräver:** Ett fältnamn och ett värde

Syntax för en jämförelseoperator är:

`table-name::field-name=value&table-name::field-name.op=op-symbol`

där:

- `table-name` är tabellen som innehåller fältet. Det krävs bara om fältet inte finns i källtabellen i den layout som anges i frågesträngen.
- `op-symbol` är ett av nyckelorden i föregående tabell, till exempel `cn`.

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&name=Tim&name.op=cn&-find
```

**Obs!** Nyckelordet `bw` fungerar inte med datum-, tids- eller tidsstämpelsträngar eller med sökoperatoren för aktuellt datum (`/`).

Du kan använda valfri sökoperator i FileMaker Pro genom att inkludera den som en del av sökvillkoret i stället för att ange operatörnyckelordet fieldname.op. Om du till exempel vill söka efter ett intervall av värden med sökoperatör för intervall ( . . . ) behöver du inte ange något operatörnyckelord. Använd i stället tecknen " . . . " mellan intervallvärdena i sökvillkoret.

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&IDnum=915...925&-find
```

Mer information om vilka operatörer du kan använda när du söker efter text finns i [FileMaker Pro Hjälp](#).

### Frågeparametern -lay (layout)

Anger den databaslayout du vill använda.

**Värdet är:** namnet på layouten

**Krävs för:** Alla frågekommandon förutom -dbnames, -layoutnames och -scriptnames

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-view
```

### Frågeparametern -lay.response (växla layout vid svar)

Anger att FileMaker Server ska använda den layout som anges med parametern -lay vid bearbetning av en förfrågan och att layouten ska växla till den layout som anges med parametern -lay.response när XML-svaret bearbetas.

Om du inte inkluderar parametern -lay.response används den layout som anges av parametern -lay när både frågan och svaret bearbetas.

Du kan använda parametern -lay.response för XML-förfrågningar.

**Värdet är:** namnet på layouten

**Kan användas med:** Alla frågekommandon förutom -dbnames, -layoutnames och -scriptnames.

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=Budget&Salary=100000&Salary.op=gt&-find&-lay.response=ExecList
```

### Frågeparametern -lop (logisk operator)

Anger hur sökvillkoren i frågekommandot `-find` kombineras i en "and"- eller "or"-sökning.

**Värdet är:** `and` eller `or`

Om frågeparametern `-lop` inte tas med, använder frågekommandot `-find` värdet "and".

**Kan användas med:** frågekommandot `-find`

**Obs!** Stöds inte av frågekommandot `-findquery`.

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&Last+Name=Smith&Birthdate=2/5/1972&-lop=and&-find
```

### Frågeparametern -max (maximalt antal poster)

Anger det maximala antalet poster som ska returneras.

**Värdet är:** ett tal. Du kan också använda värdet `all` om du vill att alla poster ska returneras. Om du inte anger `-max` returneras alla poster.

**Kan användas med:** frågekommandona `-find`, `-findall` och `-findquery`

**Obs!** Frågeparametern `-max` påverkar inte de värden som returneras för portalposter. Mer information om hur du begränsar antalet rader som returneras för portalposter finns i "Frågeparametern `-relatedsets.max` (begränsa portalposter)" på sidan 59.

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-max=10&-findall
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-max=all&-findall
```

### Frågeparametern -modid (ändrings-ID)

Ändrings-ID är en inkrementell räknare som anger den aktuella versionen av en post. Genom att ange ett ändrings-ID tillsammans med frågekommandot `-edit` kan du vara säker på att du redigerar den aktuella versionen av en post. Om det ändrings-ID du anger inte matchar aktuellt ändrings-ID i databasen, kan frågekommandot `-edit` inte användas. I stället returneras en felkod.

**Värdet är:** ett ändrings-ID, som är en unik identifierare för den aktuella versionen av en post i en FileMaker-databas.

**Kan användas med:** frågekommandot `-edit`

**Kräver:** parametern `-recid`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=22&-modid=6&last_name=Jones&-edit
```

### Frågeparametern -query (sammanfattad sökning)

Anger frågenamn och sökkriterium för en sammansatt sökpost. Se "Frågekommando `-findquery` (sammanfattad sökning)" på sidan 49.

**Värdet är:** ett frågeuttryck.

**Krävs för:** frågekommandot `-findquery`

Syntaxen för en sammansatt sökpost är:

```
-query=<request-declarations><request-definitions>&-findquery
```

där:

`<request-declarations>` är två eller flera frågedeklarationer.

- Varje frågedeklaration är sammansatt av en eller flera identifierare, separerade av kommatecken och omgivna av parentestecken. En frågeidentifierare är bokstaven "q" följt av ett tal. Till exempel: `q1`
- Omgiven av parenteser, fungerar flera frågor som logiska sökningar med AND vilka begränsar sökresultatet. I exemplet `(q1, q2)` returneras poster som matchar `q1` och `q2`.

**Obs!** Vi rekommenderar att du inte använder samma fält för flera `q`-variabler i samma "and"-sökvillkor.

- I FileMaker Pro kan varje fråga vara antingen en sökfråga eller en uteslutningsfråga. En sökfråga lägger till de matchande posterna i sökresultatet. En uteslutningsfråga tar bort de matchande posterna från sökresultatet. Standardinställningen är sökfråga. För en uteslutningsfråga placerar du ett utropstecken (!) framför det första parentestecknet.

#### Exempel

```
(q1) ; ! (q2)
```

`q1` en sökfråga; `q2` är en uteslutningsfråga eftersom den föregås av ett utropstecken.



- Frågor separeras av semikolon. Flera sökfrågor fungerar som logiska OR-sökningar som utökar sökresultatet. I exemplet (q1) ; (q2) returneras poster som matchar q1 eller q2. Uteslutningsfrågor fungerar inte som logiska OR-sökningar eftersom uteslutningsfrågor tar bort poster från sökresultatet.
- Frågor utförs i den ordning de anges. Sökresultatet inkluderar resultatet av hela den sammansatta sökposten.

<request-definitions> är en frågedefinition för varje frågedeklaration. Varje frågedefinition består av ett sökfält och värdefinitioner. Ett minustecken (-) startar frågedefinitionen.

Syntax:

-<query-id>=<fieldname>&-<query-id>.value=<value>

### Exempel

```
-q1=typeofanimal&-q1.value=Cat
-q2=name&-q2.value=Fluffy
```

Sök poster med gråa katter som inte heter "Fluffy":

```
http://host/fmi/xml/fmresultset.xml?-db=petclinic&-lay=Patients
&-query=(q1, q2);!(q3)&-q1=typeofanimal&-q1.value=Cat&-q2=color
&-q2.value=Gray&-q3=name&-q3.value=Fluffy&-findquery
```

### Frågeparametern -recid (post-ID)

Anger posten du vill bearbeta. Används mest av frågekommandona -edit och -delete. Används av kommandot -view för att hämta data från en relaterad värdelista i syntaxen FMPXMLLAYOUT.

**Värdet är:** ett post-ID, som är unikt för en viss post i en FileMaker-databas.

**Krävs för:** frågekommandona -edit, -delete och -dup

**Kan användas med:** Kommandona -find query och -view

### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=22&-delete
http://localhost/fmi/xml/FMPXMLLAYOUT.xml?-db=test&-lay=empty&-view
&-recid=9
```

## Frågeparametern `-relatedsets.filter` (filtrera portalposter)

Anger om du vill filtrera portalposterna som ska returneras i resultatet för den här frågan.

**Värdet är:** `layout` eller `inget`.

- Om `-relatedsets.filter` ställs in på `layout`, tas det hänsyn till inställningen **Inledande rad**, som anges i dialogrutan Portalkonfiguration i FileMaker Pro.
  - Om inställningen **Tillåt vertikal rullning** har aktiverats i dialogrutan Portalkonfiguration, använder du alternativet `-relatedsets.max` när du vill ange maximalt antal poster som ska returneras. Se "Frågeparametern `-relatedsets.max` (begränsa portalposter)" nedan.
  - Om inställningen **Tillåt vertikal rullning** har inaktiverats eller om alternativet `-relatedsets.max` inte används, avgör inställningen **Antal rader** i dialogrutan Portalkonfiguration hur många portalposter som ska returneras.
- Standardvärdet är `none` om den här parametern inte har angetts. Om `-relatedsets.filter` har angetts till `none` kommer Web Publishing Engine att returnera alla poster i portalen. Värdena för **Inledande rad** och **Antal rader**, som anges i dialogrutan Portalkonfiguration, ignoreras.

### Kommentar

- Parametern `-relatedsets.filter` påverkar inte hur portalposterna sorteras i XML-frågor. Den sortering som angetts i FileMaker Pro följs oavsett om parametervärdet `-relatedsets.filter` är `layout` eller `none`.
- Inställningen **Filtrera portalposter** i dialogrutan Portalkonfiguration stöds inte för XML-frågor. Alla beräkningar som angetts för inställningen **Filtrera portalposter** ignoreras.

**Kan användas med:** `-find`, `-edit`, `-new`, `-dup` och `-findquery`.

### Exempel

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=none&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample
&-lay=English&-relatedsets.filter=layout&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=layout&-relatedsets.max=10&-findany
```

### Frågeparametern `-relatedsets.max` (begränsa portalposter)

Anger maximalt antal portalposter som returneras i resultatet för den här frågan.

**Värdet är:** ett heltal eller `all`.

- Parametern `-relatedsets.max` beaktas bara om inställningen **Tillåt vertikal rullning** har aktiverats i dialogrutan Portalkonfiguration i FileMaker Pro och parametern `-relatedsets.filter` har värdet `layout`.
- Om parametern `-relatedsets.max` anger ett heltal, returnerar Web Publishing Engine antalet portalposter som startar med en inledande rad.
- Om parametern `-relatedsets.max` har värdet `all`, returnerar Web Publishing Engine alla portalposter.

**Obs!** Mer information om hur du filtrerar portalposter finns i "Frågeparametern `-relatedsets.filter` (filtrera portalposter)" ovan.

**Kan användas med:** `-find`, `-edit`, `-new`, `-dup` och `-findquery`.

#### Exempel

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample
&-lay=English&relatedsets.filter=layout&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=layout&-relatedsets.max=10&-findany
```

### Frågeparametern `-script` (script)

Anger vilket FileMaker-script som ska utföras efter att frågekommandot och sorteringen har utförts. Se "Bearbetning av en XML-förfrågan" på sidan 41.

**Värdet är:** Scriptnamn

**Kan användas med:** Alla frågekommandon förutom `-dbnames`, `-layoutnames` och `-scriptnames`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script=myscript&-findall
```

### Frågeparametern `-script.param` (skicka parameter till script)

Skickar en parameter till det FileMaker-script som anges av `-script`.

**Värdet är:** en enda textparameter.

- Om du vill skicka flera parametrar kan du skapa en sträng som avgränsar parametrarna och låta ditt script tolka de enskilda parametrarna. Om du t.ex. skickar "param1 | param2 | param3" som en lista med tecknet "|" URL-kodat som denna: param1%7Cparam2%7Cparam3
- För att textparametern ska behandlas som ett värde som inte är text kan ditt script konvertera textvärdet. Om du t.ex. vill konvertera textvärdet till ett tal kan ditt script inkludera följande:  
GetAsNumber (Get (ScriptParam) )
- Om din fråga innehåller `-script.param` utan `-script` kommer `-script.param` att ignoreras.
- Om din fråga innehåller flera än en `-script.param` kommer Web Publishing Engine att använda det sista värde den tolkar.

**Kan användas med:** `-script`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script=myscript&-script.param=Smith%7CChatterjee%7CSu
&-findall
```

### Frågeparametern `-script.prefind` (script före sökning)

Anger vilket FileMaker-script som ska köras före sökning och sortering av poster (om så har angetts) under bearbetningen av frågekommandot `-find`.

**Värdet är:** Scriptnamn

**Kan användas med:** Alla frågekommandon förutom `-dbnames`, `-layoutnames` och `-scriptnames`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.prefind=myscript&-findall
```

### Frågeparametern `-script.prefind.param` (skicka parameter till script före sökning)

Skickar en parameter till det FileMaker-script som anges av `-script.prefind`.

**Värdet är:** en enda textparameter.

- Om du vill skicka flera parametrar kan du skapa en sträng som avgränsar parametrarna och låta ditt script tolka de enskilda parametrarna. Om du t.ex. skickar "param1 | param2 | param3" som en lista med tecknet "|" URL-kodat som denna: param1%7Cparam2%7Cparam3
- För att textparametern ska behandlas som ett värde som inte är text kan ditt script konvertera textvärdet. Om du t.ex. vill konvertera textvärdet till ett tal kan ditt script inkludera följande:  
GetAsNumber (Get (ScriptParam) )
- Om din fråga innehåller `-script.prefind.param` utan `-script.prefind` ignoreras `-script.prefind.param`.

- Om din fråga innehåller flera än en `-script.prefind.param` kommer Web Publishing Engine att använda det sista värde den tolkar.

**Kan användas med:** `-script.prefind`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.prefind=myscript&-script.prefind.param=payroll
&-findall
```

#### Frågeparametern `-script.presort` (script före sortering)

Anger vilket FileMaker-script som ska köras efter sökning av poster (om så har angetts) och före sortering av poster under bearbetningen av frågekommandot `-find`

**Kan användas med:** Alla frågekommandon förutom `-dbnames`, `-layoutnames` och `-scriptnames`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.presort=myscript&-sortfield.1=dept
&-sortfield.2=rating&-findall
```

#### Frågeparametern `-script.presort.param` (skicka parameter till script före sökning)

Skickar en parameter till det FileMaker-script som anges av `-script.presort`.

**Värdet är:** en enda textparameter.

- Om du vill skicka flera parametrar kan du skapa en sträng som avgränsar parametrarna och låta ditt script tolka de enskilda parametrarna. Om du t.ex. skickar "param1 | param2 | param3" som en lista med tecknet "|" URL-kodat som denna: param1%7Cparam2%7Cparam3
- För att textparametern ska behandlas som ett värde som inte är text kan ditt script konvertera textvärdet. Om du t.ex. vill konvertera textvärdet till ett tal kan ditt script inkludera följande: `GetAsNumber (Get (ScriptParam) )`
- Om din fråga innehåller `-script.presort.param` utan `-script.presort` ignoreras `-script.presort.param`.
- Om din fråga innehåller flera än en `-script.presort.param` kommer Web Publishing Engine att använda det sista värde den tolkar.

**Kan användas med:** `-script.presort`

#### Exempel

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.presort=myscript&-script.presort.param=18%7C65
&-sortfield.1=dept&-sortfield.2=rating&-findall
```

### Frågeparametern `-skip` (hoppa över poster)

Anger hur många poster som ska hoppas över i sökresultatet.

**Värdet är:** ett tal. Om värdet är större än antalet poster i sökresultatet visas inga poster. Standardvärdet är 0.

**Kan användas med:** frågekommandot `-find`

#### Exempel

De 10 första posterna i sökresultatet hoppas över och posterna 11 till och med 15 returneras:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-skip=10&-max=5&-findall
```

### Frågeparametern `-sortfield` (sortera fält)

Anger vilket fält som ska användas för sortering.

**Värdet är:** fältnamn

**Kan användas med:** frågekommandona `-find` eller `-findall`

Frågeparametern `-sortfield` kan användas flera gånger om du vill göra upprepade fältsorteringar. Använd följande syntax när du vill ange vilken prioritet sorteringsfälten ska ha:

`-sortfield.precedence-number=fully-qualified-field-name`

där `precedence-number` i frågeparametern `-sortfield.precedence-number` är ett tal som anger prioritet när flera fält ska sorteras. Värdet för `precedence-number`:

- måste börja från 1
- måste ökas stegvis
- får inte vara större än 9

#### Exempel

Fältet "dept" sorteras först och därefter fältet "rating". Båda fälten sorteras i stigande ordning eftersom frågeparametern `-sortorder` inte har angetts.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=performance&-sortfield.1=dept&-sortfield.2=rating&-findall
```

## Frågeparametern -sortorder (sorteringsordning)

Anger sorteringsordningens riktning.

**Värdet är:** Sorteringsordningen. Giltiga sorteringsordningar visas nedan, där <value-list-name> är namnet på en värdelista, till exempel Custom:

Nyckelord	FileMaker Pro, motsvarande operator
ascend (stigande)	Sortera a till ö, -10 till 10
descend (fallande)	Sortera ö till a, 10 till -10
<value-list-name>	Sortera enligt den angivna värdelistan som är associerad med fältet i layouten

**Kan användas med:** frågekommandona -find eller -findall

**Kräver:** frågeparametern -sortfield

Du kan använda frågeparametern -sortorder tillsammans med frågeparametern -sortfield om du vill ange sorteringsordning för flera sorteringsfält. Använd följande syntax när du vill ange sorteringsordningen för ett sorteringsfält:

-sortorder.precedence-number=sort-method

där:

- precedence-number i parametern -sortorder.precedence-number är ett tal mellan 1 och 9 som anger den -sortfield-frågeparameter som frågeparametern -sortorder ska användas på.
- sort-method är ett av nyckelorden i föregående tabell som anger sorteringsordning, till exempel ascend.

### Exempel

Sorteringsordningen för det högst prioriterade sorteringsfältet (dept) är stigande och sorteringsordningen för det näst högst prioriterade sorteringsfältet (rating) är fallande. Med precedence-number 2 i -sortorder.2 anges att frågeparametern -sortorder.2=descend gäller för frågeparametern -sortfield.2=rating.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=performance&-sortfield.1=dept&-sortorder.1=ascend
&-sortfield.2=rating&-sortorder.2=descend&-findall
```

**Obs!** Om ingen -sortorder-frågeparameter anges för ett sorteringsfält används standardsorteringen, som är stigande.

# Kapitel 6

## Om anpassad webbpublicering med PHP

Med anpassad webbpublicering med PHP kan du använda scriptspråket PHP för att integrera data från FileMaker-databaser med dina anpassade webbplatslayouter. Med anpassad webbpublicering med PHP får du FileMaker API för PHP, som är en PHP-klass som har skapats av FileMaker som ger åtkomst till databaser som FileMaker Server är värd för. Denna PHP-klass ansluter till FileMaker Servers Web Publishing Engine och gör data tillgängliga för din webbserverns PHP-motor.

### Viktiga funktioner i anpassad webbpublicering med PHP

- Skapa webbapplikationer som använder programmeringsspråket PHP med öppen källkod. Använd den version av PHP som stöds av FileMaker. Du kan även använda en egen version av PHP. (Om du tänker använda en egen version av PHP, se "Installera FileMaker API för PHP manuellt" på sidan 66.)
- Databaser på FileMaker Server FileMaker Pro krävs inte för anpassad webbpublicering eftersom FileMaker Server är värd för databaserna.
- Skriv PHP-kod som kan skapa, radera, redigera och duplicera poster i en delad FileMaker-databas. Din kod kan utföra kontroll av fält och poster innan ändringar verkställs i den delade databasen.
- Skriv PHP-kod som ger åtkomst till layouter, portaler, värdelistor och relaterade fält. Precis som i FileMaker Pro är åtkomsten till data, layouter och fält baserad på användarkontots inställningar som anges i databasens behörighetsinställningar. Web Publishing Engine har också flera andra säkerhetsförbättringar. Se "Skydda publicerade databaser" på sidan 14.
- Skriv PHP-kod som utför komplexa scripts med flera steg. FileMaker har stöd för många scriptsteg för anpassad webbpublicering. Se "FileMaker-scripts och anpassad webbpublicering" på sidan 18.
- Skriv PHP-kod som utför komplexa sökningar.

### Krav för anpassad webbpublicering

Detta avsnitt förklarar vad som krävs för att utveckla en anpassad webbpubliceringslösning med hjälp av PHP, vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning och vilken påverkan det kan ha på din server att vara värd för en webbpubliceringslösning.

#### Vad som krävs för att publicera en databas med anpassad webbpublicering

Om du ska publicera databaser med anpassad webbpublicering med PHP, behöver du följande:

- en FileMaker Server-lösning, som innehåller tre komponenter:
  - en webbserver, antingen Microsoft IIS (Windows) eller Apache (macOS) – FileMakers webbservermodul är installerad på webbservern
  - Web Publishing Engine för FileMaker
  - databasserver för FileMaker



- en eller fler FileMaker Pro-databaser på FileMaker Server
- IP-adressen eller domännamnet på värden där webbservern körs
- en webbläsare och åtkomst till webbservern så att du kan utveckla och testa den egna webbpubliceringslösningen
- PHP som installerats på webbservern – FileMaker Server kan installera den version av PHP som stöds, du kan även använda din anpassade version.
  - Information om det lägsta kravet för PHP finns i [tekniska specifikationer för FileMaker Server](#).
  - Mer information om PHP finns på [php.net](#).
  - Den version av PHP som är installerad på webbservern måste stödja cURL-funktioner (client URL library). Information om cURL finns på [php.net/curl](#).

**Viktigt!** När du installerar den version av PHP som stöds av FileMaker Server visas det inte i verktyget macOS Server Admin; den ska inte finnas med på listan. Om du använder verktyget macOS Server Admin för att aktivera PHP inaktiverar du den version av PHP som stöds av FileMaker, och aktiverar din anpassade version av PHP.

Se [Installations- och konfigurationsguiden för FileMaker Server](#).

### Vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning

Webbanvändare som vill ha åtkomst till en anpassad webbpubliceringslösning som använder PHP behöver följande:

- en webbläsare
- åtkomst till Internet eller ett intranät och webbservern
- IP-adressen eller domännamnet på värden där webbservern körs

Om databasen är lösenordsskyddad måste webbanvändare även uppge ett användarnamn och ett lösenord för ett databaskonto.

### Ansluta till Internet eller ett intranät

När du publicerar databaser på Internet eller ett intranät måste värddatorn köra FileMaker Server och de databaser du vill dela måste köras och vara tillgängliga. Dessutom gäller följande:

- Publicera din databas på en dator med fast Internet- eller intranätanslutning. Det går att publicera databaser utan att ha en fast anslutning, men dessa blir bara tillgängliga för webbanvändarna när din dator är ansluten till Internet eller ett intranät.
- Värddatorn för den webbserver som är en del av FileMaker Server-driftsättningen måste ha en dedicerad statisk (permanent) IP-adress eller ett domännamn. Om du ansluter till Internet via en Internetleverantör (ISP) kan din IP-adress vara dynamiskt allokerad (dvs olika varje gång du ansluter). Med en dynamisk IP-adress blir det svårare för webbanvändarna att hitta dina databaser. Om du inte är säker på vilken typ av anslutning du ska välja kontaktar du din Internetleverantör eller nätverksadministratör.

## Installera FileMaker API för PHP manuellt

När du installerar FileMaker Server har du möjlighet att installera den version av PHP som stöds av FileMaker. Om du redan har en PHP-motor installerad och konfigurerad och du endast vill lägga till FileMaker API för PHP kan du installera FileMaker API för PHP-klass manuellt för att göra det tillgängligt för dina PHP-scripts.

Om du inte har installerat den version av PHP som stöds av FileMaker, ska du se till att göra följande konfigureringsåtgärder i din version av PHP-motorn:

- Aktivera cURL-modulen i php.ini.
- Ange platsen för FileMaker API för PHP i variabeln `include_path` i php.ini.
- Om du använder en databas som innehåller datum och tidpunkter ska du installera [paketet "pear date"](#).

**Obs!** Information om det lägsta kravet för PHP finns i [tekniska specifikationer för FileMaker Server](#). Systemet fungerar bäst om du använder de rekommenderade versionerna av PHP.

### Att göra FileMaker API för PHP tillgängligt för dina PHP-scripts

När du installerade FileMaker Server följde FileMaker API för PHP-paketet med som en .zip-fil på följande plats:

- För IIS (Windows):  
`[enhet]:\Program Files\FileMaker\FileMaker Server\Web Publishing\FM_API_for_PHP_Standalone.zip`  
där [enhet] är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.
- För Apache (macOS):  
`/Bibliotek/FileMaker Server/Web Publishing/FM_API_for_PHP_Standalone.zip`

Filen `FM_API_for_PHP_Standalone.zip` innehåller filen `FileMaker.php` och mappen `FileMaker`. Packa upp filen och kopiera filen `FileMaker.php` och mappen `FileMaker` till någon av dessa platser:

- mappen där dina PHP-scripts finns.
  - För IIS (Windows) genom HTTP eller HTTPS:  
`[enhet]:\Program\FileMaker\FileMaker Server\HTTPServer\Conf`  
där [enhet] är den enhet där Web Publishing Engine-komponenten för din FileMaker Server-driftsättning finns.
  - För Apache (macOS) genom HTTP:  
`/Bibliotek/FileMaker Server/HTTPServer/htdocs`
- För Apache (macOS) genom HTTPS:  
`/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot`

**Obs!** Om du aktiverar **Använd HSTS för webbklienter** i Admin Console använder du HTTPS-katalogen som värd för webbplatsens PHP-filer.

- en av `include_path`-katalogerna i din PHP-installation. Standardplatsen för macOS är `/usr/lib/php`.

## Så går du vidare

- Använd FileMaker Server Admin Console för att aktivera anpassad webbpublicering. Se [FileMaker Server Help](#) och [Installations- och konfigurationsguiden för FileMaker Server](#).
- I FileMaker Pro öppnar du varje FileMaker-databas som du vill publicera och ser till att databasen har rätt utökad behörighet aktiverad för anpassad webbpublicering. Se "Aktivera anpassad webbpublicering i en databas" på sidan 13.
- Mer information om hur man får åtkomst till data i FileMaker-databaser med FileMaker API för PHP finns i kapitel 8, "Använda FileMaker API för PHP".

# Kapitel 7

## Översikt över anpassad webbpublicering med PHP

Med FileMaker API för PHP kan du integrera data från FileMaker Pro-databaser med PHP-lösningar. I detta kapitel beskrivs hur PHP fungerar tillsammans med FileMaker Servers anpassade Web Publishing Engine. Mer detaljerad information om FileMaker API för PHP finns i kapitel 8, "Använda FileMaker API för PHP".

### Hur Web Publishing Engine fungerar med PHP-lösningar

FileMaker Server består av tre komponenter: en webbserver, Web Publishing Engine och databasservern. Se [Installations- och konfigurationsguiden för FileMaker Server](#). För att stödja PHP-lösningar installeras en PHP-motor med webbservern på huvuddatorn. FileMaker Server är värd till en PHP-lösning när du placerar PHP-filerna på huvuddatorns webbserver.

- När en webbanvändare öppnar en PHP-lösning vidarebefordrar webbservern förfrågan till PHP-motorn, som behandlar PHP-koden.
- Om PHP-koden innehåller anrop till FileMaker API för PHP tolkas de anropen och skickas som förfrågningar till Web Publishing Engine.
- Web Publishing Engine efterfrågar data från databaser som databasservern är värd för.
- Databasservern skickar efterfrågade data till Web Publishing Engine.
- Web Publishing Engine skickar data till PHP-motorn på webbservern som svar på API-anropet.
- PHP-lösningen behandlar informationen och visar den för webbanvändaren.

### Allmänna steg för anpassad webbpublicering med PHP

1. Kontrollera att **Aktivera PHP-publicering** är aktiverat i Admin Console. Se [Installations- och konfigurationsguiden för FileMaker Server](#).
2. I Admin Console väljer du fönstret **Databaser** och ser till att varje FileMaker-databas som du vill publicera har den fmphp-utökade behörigheten aktiverad för anpassad webbpublicering med PHP.

Om det behövs använder du FileMaker Pro och aktiverar anpassad webbpublicering för en databas. Se kapitel 2, "Förbereda databaser för anpassad webbpublicering".

**Obs!** Se till att använda motsvarande FileMaker-databasbehörigheter när du utvecklar PHP-lösningar som ska användas av slutanvändaren. I annat fall kanske du har tillgång till layouter och funktioner i FileMaker-databasen som inte är tillgängliga för slutanvändaren, vilket ger inkonsekventa funktioner.

3. Skapa din PHP-lösning med PHP-redigeringsverktygen och integrera FileMaker API-funktioner i din PHP-kod för att få åtkomst till dina FileMaker-data. Se kapitel 8, "Använda FileMaker API för PHP".

4. Kopiera eller flytta webbplatsens katalogstruktur och filer till följande mapp på huvuddatorns webbserver.

- För IIS (Windows) genom HTTP eller HTTPS:  
[enhet]:\Program Files\FileMaker\FileMaker Server\HTTPServer\Conf  
där [enhet] är den enhet som innehåller din FileMaker Server-driftsättning.
- För Apache (macOS) genom HTTP:  
/Bibliotek/FileMaker Server/HTTPServer/htdocs
- För Apache (macOS) genom HTTPS:  
/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot

**Obs!** Använd HTTPS-katalogen som värd för webbplatsens PHP-filer när **Använd HSTS för webbklienter** är aktiverat i Admin Console. När **Använd HSTS för webbklienter** är aktiverat hindrar webbläsaren klienten från att använda en HTTP-anslutning när en webbklient ansluter till FileMaker Server med HTTPS. Det gäller för allt webbinnehåll som FileMaker Server är värd för.

5. Om ett containerfält i en databas lagrar en filreferens istället för en verklig fil måste det hänvisade containerobjektet lagras i FileMaker Pro-katalogen när posten skapas eller redigeras. Du måste kopiera eller flytta objektet till en katalog med samma relativa position i rotkatalogen på webbserverprogramvaran.

Se "Publicera innehållet i containerfält på webben" på sidan 15.

6. Se till att säkerhetsfunktionerna för platsen eller programmet fungerar.

7. Testa din webbplats genom att använda samma konton och behörigheter som har definierats för webbanvändare.

8. Informera om webbplatsen och gör den tillgänglig. Den URL som webbanvändarna anger har detta format:

`http://<server>/<webbplats_sökväg>`

- <server> är den maskin där FileMaker Server finns.
- <webbplats\_sökväg> är den relativa sökvägen till startsidan för din webbplats, som avgörs av den katalogstruktur som du använde i steg 4 ovan.

### Exempel

Om din webbserver är 192.168.123.101 och din webbplats startsida ligger på webbservern i `c:\inetpub\wwwroot\customers\index.php` anger webbanvändaren denna URL:

`http://192.168.123.101/customers/index.php`

**Obs!** PHP använder Latin-1-kodning (ISO-8859-1). FileMaker Server returnerar Unicode-data (UTF-8). Använd FileMaker Server Admin Console för att ange standardteckenkodningen för din webbplats. För PHP-webbplatser kan du ange antingen UTF-8 eller ISO-8859-1, men vi rekommenderar UTF-8. Ange samma inställning för `charset`-attributet i `<HEAD>`-avsnittet i webbplatsens PHP-filer.

Mer information om hur du installerar och använder en PHP-lösning finns i kapitel 9, "Iscensätta, testa och övervaka en webbplats".

# Kapitel 8

## Använda FileMaker API för PHP

I FileMaker API för PHP ingår en PHP-klass – FileMaker class – som ger ett objektorienterat gränssnitt för FileMaker-databaser. FileMaker API för PHP gör att man kan få åtkomst till både uppgifter och logik som har lagrats i FileMaker Pro-databaser och publicera dem på Internet, eller exportera dem till andra program.

Med FileMaker API för PHP kan PHP-kod utföra samma sorts funktioner som du redan har tillgång till i FileMaker Pro-databaser:

- skapa, radera, redigera och duplicera poster
- utföra sökningar
- kontrollera fält och poster
- använda layouter
- köra FileMaker-scripts
- visa portaler och relaterade poster
- använda värdelistor

I detta kapitel beskrivs hur man använder FileMaker class-objekt och metoder för att lägga till dessa vanliga funktioner i en PHP-lösning. Detta kapitel täcker inte hela FileMaker API för PHP, men introducerar nyckelobjekt och metoder.

### Mer information

Mer information om FileMaker API för PHP finns på följande ställen.

Om du redan har en PHP-motor installerad och konfigurerad och du endast vill lägga till FileMaker API för PHP går du till "Installera FileMaker API för PHP manuellt" på sidan 66.

### Referens för FileMaker API för PHP

Om du installerade FileMaker API för PHP hittar du referensinformation i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows):  
[enhet]:\Program Files\FileMaker\FileMaker Server\Documentation\PHP API Documentation\index.html  
där [enhet] är den enhet som innehåller webbserverkomponenten för din FileMaker Server-driftsättning.
- För Apache (macOS): /Library/FileMaker Server/Documentation/PHP API Documentation/index.html

### Support för FileMaker API för PHP

Du hittar mer information om FileMaker API för PHP på FileMakers [support](#) sida.

## Använda FileMaker class

Om du vill använda FileMaker class i din PHP-lösning lägger du till följande uttryck i din PHP-kod:

```
require_once ('FileMaker.php');
```

### FileMaker class-objekt

FileMaker class definierar klassobjekt som du kan använda för att hämta uppgifter från FileMaker Pro-databaser.

Klassobjekt	Använda objektet för att
FileMaker-databas	Definiera databasegenskaperna Ansluta till en FileMaker Pro-databas Skaffa information om FileMaker API för PHP
Kommando	Skapa kommandon som lägger till poster, tar bort poster, duplicerar poster, ändrar poster, utför sökningar och utför scripts
Layout	Arbeta med databaslayouter
Post	Arbeta med postdata
Fält	Arbeta med fältdata
Relaterad uppsättning	Arbeta med portalposter
Resultat	Bearbeta posterna från en sökning
Fel	Kontrollera om ett fel har inträffat Behandla fel

### FileMaker kommandoobjekt

FileMaker class definierar ett baskommandoobjekt som du kan använda för att exemplifiera ett specifikt kommando och för att ange kommandots parametrar. För att utföra kommandot måste du använda metoden `execute()`.

FileMaker class definierar följande specifika kommandon:

- Kommandot Lägg till
- Kommandot Sammanfattad sökning
- Kommandot Radera
- Kommandot Duplicera
- Kommandot Redigera
- Kommandot Utför sökning, kommandot Visa alla, kommandot Sök efter alla
- Kommandot Sökning, som läggs till i kommandot Sammanfattad sökning
- Kommandot Utför script

**Viktigt!** Kommandon har olika returvärden enligt definition av `FileMaker.php`-klassen. Vissa kommandon returnerar exempelvis det logiska värdet `TRUE` eller ett `FileMaker_Error`-objekt. Andra kommandon returnerar ett `FileMaker_Result`-objekt som kan innehålla ett helt "sökresultat" av poster i en layout. Undvik att låta datorns minne bli överbelastat av problem genom att kontrollera de förväntade returvärdena för de kommandon du använder. I "Referens för FileMaker API för PHP" finns detaljerad information om returvärden för varje kommando.

De grundläggande uppgifter som de flesta PHP-program behöver utföra beskrivs i följande:

- "Arbeta med poster" på sidan 73
- "Köra FileMaker-scripts" på sidan 75
- "Utföra sökningar" på sidan 81

## Avkoda data för användning i FileMaker API

Om ditt PHP-program hämtar data från en webbplats kan dessa data vara URL-kodade. FileMaker API för PHP förväntar sig data i form av avkodade strängar, inte strängar med URL-kod. Generellt kan det vara lämpligt att anropa funktionen `urldecode()` när du hämtar data i PHP-programmet.

### Exempel

```
$user = urldecode($_GET['user']);  
$event = urldecode($_GET['event']);
```

**Obs!** Undvik att använda strängar som innehåller et-tecken (&) med FileMaker API för PHP. Använd ett omvänt snedstreck som Escape-tecken före specialtecken i strängar som skickas till FileMaker API för PHP.

## Ansluta till en FileMaker-databas

FileMaker class definierar ett databasobjekt som du exemplifierar för att ansluta till en server eller en databas. Definiera objektens egenskaper med klasskonstruktören, eller genom att använda metoden `setProperty()`.

### Exempel

Ansluta till en server för att få en lista över databaser:

```
$u$fm = new FileMaker();  
$databases = $fm->listDatabases();
```

Ansluta till en specifik databas på en server:

```
$fm = new FileMaker();  
$fm->setProperty('database', 'questionnaire');  
$fm->setProperty('hostspec', 'http://192.168.100.110');  
$fm->setProperty('username', 'web');  
$fm->setProperty('password', 'web');
```

Egenskaperna för användarnamn och lösenord avgör behörighetsuppsättningen för denna anslutning.

**Obs!** Egenskapen `hostspec` har standardvärdet `http://localhost`. Eftersom PHP-motorn installeras med huvuddatorns webbserverkomponent behöver man inte ange `hostspec`-egenskapen.



## Arbeta med poster

FileMaker class definierar ett postobjekt som du kan exemplifiera för att arbeta med poster. Ett exempel på ett postobjekt representerar en post från en FileMaker Pro-databas. Använd ett postobjekt med kommandona Lägg till, Radera, Duplicera och Redigera för att ändra uppgifterna i posten. Sökkommandona – Utför sökning, Visa alla, Sök efter alla och Sammanfattad sökning – returnerar en samling postobjekt.

### Skapa en post

Det finns två sätt att skapa en post:

- Använd metoden `createRecord()` och ange ett layoutnamn, du kan även ange olika fältvärden. Du kan även ställa in värden individuellt i det nya postobjektet.

Metoden `createRecord()` sparar inte den nya posten i databasen. Om du vill spara posten i databasen använder du metoden `commit()`.

#### Exempel

```
$rec = $fm->createRecord('Form View', $values);  
$result = $rec->commit();
```

Med `FileMaker_Record commit()`-metoden tilldelas variabeln `$result` det logiska värdet `TRUE` och den nya posten skapas i FileMaker-databasen när det inte finns några fel.

Om något fel uppstår innehåller variabeln `$result` ett `FileMaker_Error`-objekt. Kontrollera om det finns några fel när du använder metoden `commit()`.

- Använda kommandot Lägg till. Använd metoden `newAddCommand()` för att skapa ett `FileMaker_Command_Add`-objekt, ange layoutnamnet och postdata. Om du vill spara posten i databasen använder du metoden `execute()`.

#### Exempel

```
$newAdd = $fm->newAddCommand('Respondent', $respondent_data);  
$result = $newAdd->execute();
```

Med `FileMaker_Command execute()`-metoden innehåller variabeln `$result` ett `FileMaker_Result`-objekt med all information om den skapade posten när det inte finns några fel.

Om något fel uppstår innehåller variabeln `$result` ett `FileMaker_Error`-objekt. Kontrollera om det finns några fel när du använder metoden `execute()`.

### Duplicera en post

Du kan duplicera en befintlig post med kommandot Duplicera. Använd metoden `newDuplicateCommand()` för att skapa ett `FileMaker_Command_Duplicate`-objekt, ange layoutnamn och post-ID för den post du vill duplicera. Duplicera sedan posten genom att använda metoden `execute()`.

### Exempel

```
$newDuplicate = $fm->newDuplicateCommand('Respondent', $rec_ID);  
$result = $newDuplicate->execute();
```

### Redigera en post

Det finns två sätt att redigera en post:

- Använda kommandot Redigera. Använd metoden `newEditCommand()` för att skapa ett `FileMaker_Command_Edit`-objekt, ange layoutnamn, post-ID för den post du vill redigera och värden som du vill uppdatera. Redigera sedan posten genom att använda metoden `execute()`.

#### Exempel

```
$newEdit = $fm->newEditCommand('Respondent', $rec_ID, $respondent_data);  
$result = $newEdit->execute();
```

- Använda ett postobjekt. Hämta en post från databasen, ändra fältvärdena och redigera sedan posten genom att använda metoden `commit()`.

#### Exempel

```
$rec = $fm->getRecordById('Form View', $rec_ID);  
$rec->setField('Name', $nameEntered);  
$result = $rec->commit();
```

### Radera en post

Det finns två sätt att radera en post:

- Hämta en post från databasen och använd sedan metoden `delete()`.

#### Exempel

```
$rec = $fm->getRecordById('Form View', $rec_ID);  
$rec->delete();
```

- Du kan radera en befintlig post med kommandot Radera. Använd metoden `newDeleteCommand()` för att skapa ett `FileMaker_Command_Delete`-objekt, ange layoutnamn och post-ID för den post du vill radera. Radera sedan posten genom att använda metoden `execute()`.

#### Exempel

```
$newDelete = $fm->newDeleteCommand('Respondent', $rec_ID);  
$result = $newDelete->execute();
```

## Köra FileMaker-scripts

Ett FileMaker-script är en namngiven uppsättning scriptsteg. FileMaker class definierar olika metoder som gör att du kan arbeta med FileMaker-scripts som har definierats i en FileMaker Pro-databas. Mer information om webbkompatibla scriptsteg (de scriptsteg som kan utföras i en webblösning) finns i "FileMaker-scripts och anpassad webbpublicering" på sidan 18.

### Hämta en lista på tillgängliga scripts

Använd metoden `listScripts()` för att hämta en lista över tillgängliga scripts från den aktuella anslutna databasen. Metoden `listScripts()` returnerar scripts som kan utföras av det användarnamn och lösenord som angavs när databasanslutningen definierades. (Se "Ansluta till en FileMaker-databas" på sidan 72.)

#### Exempel

```
$scripts = $fm->listScripts();
```

### Köra ett FileMaker-script

Använd metoden `newPerformScriptCommand()` för att skapa ett `FileMaker_Command_PerformScript`-objekt, ange layout, scriptnamn och alla scriptparametrar. Utför sedan scriptet genom att använda metoden `execute()`.

**Viktigt!** När du kör ett FileMaker-script är det dess beteende som avgör storleken på det `FileMaker_Result`-objekt som returneras. Om ett FileMaker-script till exempel växlar till en angiven layout kan alla poster från den layoutens tabell finnas i sökresultatet och alla poster i sökresultatet kan returneras i objektet `FileMaker_Result`. Undvik att låta datorns minne bli överbelastat av problem genom att kontrollera de data som returneras av ett FileMaker-script innan du kör det FileMaker-scriptet i ett PHP-program.

#### Exempel

```
$newPerformScript = $fm->newPerformScriptCommand('Order Summary',  
'ComputeTotal');  
$result = $newPerformScript->execute();
```

### Köra ett script innan ett kommando utförs

Använd metoden `setPreCommandScript()` för att ange ett script som körs innan ett kommando körs. Följande exempel använder ett sökkommando, men du kan använda metoden `setPreCommandScript()` med vilket kommando som helst.

#### Exempel

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->addFindCriterion('GPA', $searchValue);  
$findCommand->setPreCommandScript('UpdateGPA');  
$result = $findCommand->execute();
```

## Köra ett script innan en resultatuppsättning sorteras

Använd metoden `setPreSortScript()` för att ange ett script som körs efter ett sökresultat har genererats, men innan resultatuppsättningen sorteras. Se "Använda kommandot Utför sökning" på sidan 82.

### Exempel

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->setPreSortScript('RemoveExpelled');
```

## Köra ett script efter resultatuppsättningen har genererats

Använd metoden `setScript()` för att ange ett scriptsteg som körs efter ett sökresultat har genererats. Se "Använda kommandot Utför sökning" på sidan 82.

### Exempel

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->setScript('myScript','param1|param2|param3');
```

## Ordning för att utföra scripts

Du kan använda metoderna `setPreCommandScript()`, `setPreSortScript()` och `setScript()` tillsammans med metoderna `setResultLayout()` och `addSortRule()` för ett enda kommando.

FileMaker Server och Web Publishing Engine bearbetar dessa metoder i följande ordning:

1. Kör det script som har angetts för metoden `setPreCommandScript()` om detta har angetts.
2. Behandlar själva kommandot, exempelvis kommandona Utför sökning eller Radera post.
3. Kör det script som har angetts för metoden `setPreSortScript()` om detta har angetts.
4. Sorterar sökresultatet, om metoden `addSortRule()` har angetts.
5. Gör att metoden `setResultLayout()` växlar till en annan layout, om detta har angetts.
6. Kör det script som har angetts för metoden `setScript()` om detta har angetts.
7. Returnerar det slutgiltiga sökresultatet.

Om något av ovanstående steg genererar en felkod, stoppas kommandot och alla eventuella kvarvarande steg utförs inte. Eventuella tidigare steg i förfråganen utförs fortfarande.

Anta att ett kommando tar bort den aktuella posten, sorterar posterna och sedan kör ett script. Om metoden `addSortRule()` anger ett fält som inte finns, tar förfrågan bort den aktuella posten och genererar felkod 102 ("Fältet saknas"), men scriptet utförs inte.

Den layout som anges för metoden `newFindCommand()` används när sökförfrågan behandlas. När metoden `setResultLayout()` växlar till en annan layout är felobjektet för sökförfrågan som är baserad på originallayouten inte längre tillgänglig. Testa felobjektet från sökförfrågan som är baserad på originallayouten genom att markera felobjektet innan du ändrar layout.

### Exempel

```
request = $fm->newFindCommand('Students');
$request->addFindCriterion('Day', 'Wednesday');

// Utför sökningen
$result = $request->execute();

if (FileMaker::isError($result)) {
    if ($result->code = 401) {
        $findError = 'Det finns inga poster som matchar denna förfrågan: ' . ' ('
            . $result->code . ')';
    } else {
        $findError = 'Sökningsfel: ' . $result->getMessage() . ' (' . $result-
            >code . ')';
    }
}
$request->setResultLayout('Teachers');
// Växla till resultatlayouten.
$result = $request->execute();
```

## Arbeta med FileMaker-layouter

En layout är en samordning av fält, objekt, bilder och layoutdelar som visar på vilket sätt information organiseras och presenteras när användaren bearbetar, granskar eller skriver ut poster. FileMaker class definierar olika metoder som gör att du kan arbeta med de layouter som har definierats i en FileMaker Pro-databas. Du kan få information om layouter från flera av FileMaker class-objekten.

Med detta klassobjekt	Använd dessa metoder
Databas	<ul style="list-style-type: none"> <li>▪ <code>listLayouts()</code> hämtar en lista över tillgängliga layoutnamn.</li> <li>▪ <code>getLayout()</code> hämtar ett layoutobjekt genom att ange ett layoutnamn.</li> </ul>
Layout	<ul style="list-style-type: none"> <li>▪ <code>getName()</code> hämtar layoutnamnet för ett specifikt layoutobjekt.</li> <li>▪ <code>listFields()</code> hämtar alla fältnamn som används i en layout.</li> <li>▪ <code>getFields()</code> hämtar ett associativt urval av namn på alla fält som nycklar, och de associativa FileMaker_Field-objekten som matrisvärden.</li> <li>▪ <code>listValueLists()</code> hämtar en lista över namn på värdelistor.</li> <li>▪ <code>listRelatedSets()</code> hämtar en lista över namn på relaterade uppsättningar.</li> <li>▪ <code>getDatabase()</code> returnerar namnet på databasen.</li> </ul>
Post	<ul style="list-style-type: none"> <li>▪ <code>getLayout()</code> returnerar det layoutobjekt som är associerat med en specifik post.</li> </ul>
Fält	<ul style="list-style-type: none"> <li>▪ <code>getLayout()</code> returnerar det layoutobjekt som innehåller ett specifikt fält.</li> </ul>
Kommando	<ul style="list-style-type: none"> <li>▪ <code>setResultLayout()</code> returnerar kommandots resultat i en annan layout än den aktuella layouten.</li> </ul>

## Använda portaler

En portal är en tabell som visar rader med data från en eller flera relaterade poster. FileMaker class definierar en relaterad uppsättning objekt och flera metoder som gör att du kan arbeta med portaler som har definierats i en FileMaker Pro-databas.

Ett relaterat uppsättningsobjekt är en matris med postobjekt från den relaterade portalen – varje postobjekt representerar en rad med data i portalen.

### Lista de portaler som har definierats på en specifik layout

För ett specifikt layoutobjekt använder du metoden `listRelatedSets()` för att hämta en lista över tabellnamn för alla portaler som har definierats i denna layout.

#### Exempel

```
$tableNames = $currentLayout->listRelatedSets();
```

### Hämta portalnamn för ett specifikt resultatobjekt

För ett specifikt FileMaker\_Result-objekt använder du metoden `getRelatedSets()` för att hämta namnen på alla portaler i denna post.

#### Exempel

```
$relatedSetsNames = $result->getRelatedSets();
```

### Hämta information om portaler för en specifik layout

För ett specifikt layoutobjekt använder du metoden `getRelatedSets()` för att hämta FileMaker\_RelatedSet-objekt som beskriver portalerna i layouten. Den returnerade matrisen är en associativ matris med tabellnamnen som matrisnycklar och de associerade FileMaker\_RelatedSet-objekten som matrisvärden.

#### Exempel

```
$relatedSetsArray = $currentLayout->getRelatedSets();
```

### Hämta information för en specifik portal

För ett specifikt layoutobjekt använder du metoden `getRelatedSet()` för att hämta det FileMaker\_RelatedSet-objekt som beskriver den specifika portalen.

#### Exempel

```
$relatedSet = $currentLayout->getRelatedSet('customers');
```

### Hämta tabellnamnet för en portal

För ett relaterat uppsättningsobjekt använder du metoden `getName()` för att få tabellnamnet för portalen.

### Exempel

```
$tableName = $relatedSet->getName();
```

### Hämta portalposterna för en specifik post

För ett specifikt postobjekt använder du metoden `getRelatedSet()` för att hämta relaterade poster för en specifik portal på den posten.

### Exempel

```
$relatedRecordsArray = $currentRecord->getRelatedSet('customers');
```

### Skapa en ny post i en portal

Använd metoden `newRelatedRecord()` för att skapa en ny post i den angivna relaterade uppsättningen och utför ändringen i databasen genom att använda metoden `commit()`.

### Exempel

```
//skapa en ny portalrad i "kundportalen"
$new_row = $currentRecord->newRelatedRecord('customer');

//ställ in fältvärden på den nya portalraden
$new_row->setField('customer::name', $newName);
$new_row->setField('customer::company', $newCompany);

$result = $new_row->commit();
```

### Ta bort en post från en portal

Använd metoden `delete()` för att ta bort en post i en portal.

### Exempel

```
$relatedSet = $currentRecord->getRelatedSet('customers');
/* Kör igenom alla portalrader */
foreach ($relatedSet as $nextRow) {
    $nameField = $nextRow->getField('customer::name')
    if ($nameField == $badName) {
        $result = $newRow->delete();
    }
}
```

## Använda värdelistor

En värdelista är en uppsättning fördefinierade alternativ. FileMaker class definierar olika metoder som gör att du kan arbeta med värdelistor som har definierats i en FileMaker Pro-databas.

### Hämta namnen på alla värdelistor för en specifik layout

För en specifik layout använder du metoden `listValueLists()` för att hämta en matris som innehåller namnen på värdelistorna.

#### Exempel

```
$valueListNames = $currentLayout->listValueLists();
```

### Hämta en matris över alla värdelistor för en specifik layout

För ett specifikt layoutobjekt använder du metoden `getValueListsTwoFields()` för att hämta en matris som innehåller värden från alla värdelistor. Den returnerade matrisen är en associativ matris. Matrisnycklarna är värdelistnamnen och matrisvärdena är associerade matriser som listar visningsnamn och deras relaterade alternativ från varje värdelista.

#### Exempel

```
$valueListsArray = $currentLayout->getValueListsTwoFields();
```

**Obs!** Även om metoden `getValueLists()` fortfarande stöds i FileMaker API för PHP kommer den att avvecklas. Använd istället metoden `getValueListsTwoFields()`.

### Hämta värdena för en namngiven värdelista

För ett specifikt layoutobjekt använder du metoden `getValueListTwoFields()` för att hämta en matris som innehåller alternativen för den namngivna värdelistan. Den returnerade matrisen är en associerad matris som innehåller visade värden från det andra fältet i värdelistan som nycklar och de associerade lagrade värdena från det första fältet som matrisvärden.

Beroende på vilka alternativ du har markerat i dialogrutan Ange fält i värdelista i FileMaker-databasen returnerar metoden `getValueListTwoFields()` endast värdet i det första fältet, endast värdet i det andra fältet eller värdena i de båda fälten från en värdelista som lagrade och visade värden.

- Om du inte har markerat **Visa även värden från det andra fältet** kommer metoden `getValueListTwoFields()` att returnera värdet från det första fältet i värdelistan som både lagrat värde och visat värde.
- Om du har valt både **Visa även värden från det andra fältet** och **Visa endast värden från det andra fältet** returnerar metoden `getValueListTwoFields()` värdet från det första fältet som lagrat värde och värdet från det andra fältet som visat värde.
- Om du har markerat **Visa även värden från det andra fältet** och inte har markerat **Visa värden endast från det andra fältet** returnerar metoden `getValueListTwoFields()` värdet från det första fältet som lagrat värde och värdena från både det första och det andra fältet som visat värde.



Använd upprepning med metoden `getValueListTwoFields()` när du vill hitta visat värde och lagrat värde.

### Exempel

```
$layout = $fm->getLayout('customers');  
$valuearray = $layout->getValueListTwoFields("region", 4);  
foreach ($valuearray as $displayValue => $value) {  
    ....  
}
```

### Kommentar

- Även om metoden `getValueLists()` fortfarande stöds i FileMaker API för PHP kommer den att avvecklas. Använd istället metoden `getValueListsTwoFields()`.
- När du använder metoden `getValueListTwoFields()` ska du se till att använda en `foreach`-loop för varje loop i den associerade matrisen. Använd inte en `for`-loop eftersom den kan returnera oväntade resultat.

## Utföra sökningar

FileMaker class definierar fyra olika sökkommandoobjekt:

- Kommandot Visa alla. Se "Använda kommandot Visa alla" på sidan 82.
- Kommandot Sök efter alla. Se "Använda kommandot Sök efter alla" på sidan 82.
- Kommandot Utför sökning. Se "Använda kommandot Utför sökning" på sidan 82.
- Kommandot Sammanfattad sökning. Se "Använda kommandot Sammanfattad sökning" på sidan 83.

FileMaker class definierar även flera metoder som kan användas för alla fyra typer av sökkommandon:

- Använd metoden `addSortRule()` för att lägga till en regel som definierar hur sökresultatet ska sorteras. Använd metoden `clearSortRules()` för att rensa alla sorteringsregler som har definierats.
- Använd `setLogicalOperator()` för att växla mellan logiska AND-sökningar och logiska OR-sökningar.
- Använd metoden `setRange()` för att endast efterfråga en del av sökresultatet. Använd metoden `getRange()` för att hämta den aktuella intervalldefinitionen.

Genom att använda metoden `setRange()` kan man förbättra lösningens prestanda genom att minska antalet poster som returneras av sökförfrågan. Om en sökförfrågan exempelvis returnerar 100 poster kan du dela upp sökresultatet i fem grupper om 20 träffar istället för att behandla alla 100 poster på en gång.

- Du kan utföra FileMaker-scripts tillsammans med sökkommandon.
  - Om du vill köra ett script innan du utför sökkommandot använder du metoden `setPreCommandScript()`.
  - Om du vill köra ett script innan du sorterar sökresultatet använder du metoden `setPreSortScript()`.
  - Om du vill köra ett script efter ett sökresultat har genererats, men innan sökresultatet sorteras använder du metoden `setScript()`.

### Använda kommandot Visa alla

Använd kommandot Visa alla för att hämta alla poster från en specifik layout. Använd metoden `newFindAllCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_FindAll`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

#### Exempel

```
$findCommand = $fm->newFindAllCommand('Form View');  
$result = $findCommand->execute;
```

**Obs!** När du använder kommandot Visa alla ska du undvika att låta datorns minne bli överbelastat av problem genom att ange maximalt antal returnerade poster per sida.

### Använda kommandot Sök efter alla

Använd kommandot Sök efter alla för att slumpmässigt hämta en post från en specifik layout. Använd metoden `newFindAnyCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_FindAny`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

#### Exempel

```
$findCommand = $fm->newFindAnyCommand('Form View');  
$result = $findCommand->execute;
```

### Använda kommandot Utför sökning

Använd metoden `newFindCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_Find`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

**Obs!** Se till att layoutnamnet är unikt. Om databasen har två layouter med samma namn kan FileMaker API för PHP inte skilja dem åt. Dessutom är API inte skiftlägeskänsligt. Om din databas till exempel har en layout som heter Webbplatser och en annan layout som heter WebbPlatser kan API:et inte skilja dem åt.

Använd metoden `addFindCriterion()` för att lägga till villkor i sökförfrågan. Använd metoden `clearFindCriteria()` för att rensa alla sökvillkor som har definierats.

### Exempel

Söka efter en post med fältnamn:

```
$findCommand = $fm->newFindCommand('Form View');  
$findCommand->addFindCriterion('Frågeformulärs-ID',  
$active_questionnaire_id);  
$result = $findCommand->execute();
```

Lägga till en sorteringsordning:

```
$findCommand = $fm->newFindCommand('Customer List');  
$findCommand->addSortRule('Title', 1, FILEMAKER_SORT_ASCEND);  
$result = $findCommand->execute();
```

### Använda kommandot Sammanfattad sökning

Med kommandot Sammanfattad sökning kan du kombinera flera sökobjekt i ett enda kommando. Det finns flera sätt att skapa ett Compound Find-kommando:

- Skapa ett `FileMaker_Command_CompoundFind`-objekt genom att använda metoden `newCompoundFindCommand()`.
- Skapa ett eller flera `FileMaker_Command_FindRequest`-objekt genom att använda metoden `newFindRequest()`.
- Använd metoden `add()` för att lägga till sökobjekten i kommandoobjektet Sammanfattad sökning.
- Utför sedan den sammanfattade sökningen genom att använda metoden `execute()`.

## Exempel

### Kommandot Sammanfattad sökning:

```
// Skapa kommandoobjektet Sammanfattad sökning
$compoundFind = $fm->newCompoundFindCommand('Form View');

// Skapa den första sökförfrågan
$findreq1 = $fm->newFindRequest('Form View');

// Skapa den andra sökförfrågan
$findreq2 = $fm->newFindRequest('Form View');

// Skapa den tredje sökförfrågan
$findreq3 = $fm->newFindRequest('Form View');

// Ange sökvillkor för den första sökförfrågan
$findreq1->addFindCriterion('Quantity in Stock', '<100');

// Ange sökvillkor för den andra sökförfrågan
$findreq2->addFindCriterion('Quantity in Stock', '0');

// Ange sökvillkor för den tredje sökförfrågan
$findreq3->addFindCriterion('Cover Photo Credit', 'The London Morning News');

// Ange sökförfrågningar till kommandot Sammanställd sökning
$compoundFind->add(1,$findreq1);
$compoundFind->add(2,$findreq2);
$compoundFind->add(3,$findreq3);

// Ange sorteringsordning
$compoundFind->addSortRule('Title', 1, FILEMAKER_SORT_DESCEND);

// Utför kommandot Sammanfattad sökning
$result = $compoundFind->execute();

// Hämta poster från sökresultatet
$records = $result->getRecords();

// Skriv ut antalet funna poster
echo 'Found '. count($records) . " results.<br><br>";
```

## Behandla posterna i ett sökresultat

- Hämta en matris med alla poster i sökresultatet genom att använda metoden `getRecords()`. Varje del i matrisen är ett `FileMaker_Record`-objekt, eller ett exempel på det klassnamn som har angetts i API för att exemplifiera poster. Matrisen kan vara tom om sökresultatet inte innehåller några poster.
- Hämta en lista över fältnamn för alla fält i sökresultatet genom att använda metoden `getFields()`. Metoden returnerar endast fältnamnen. Om du behöver ytterligare information om fälten använder du associerade layoutobjekt.
- Hämta antalet poster i hela sökresultatet genom att använda metoden `getFoundSetCount()`.
- Hämta antalet poster i det filtrerade sökresultatet genom att använda metoden `getFetchCount()`. Om inga parametrar har ställts in för sökkommandot är detta värde lika med resultatet med metoden `getFoundSetCount()`. Det är alltid lika med värdet för `count($response->getRecords())`.
- För en specifik post använder du metoden `getField()` för att returnera innehållet i ett fält som en sträng.
- För en specifik post använder du metoden `getFieldAsTimestamp()` för att returnera innehållet i ett fält som en Unix-tidsstämpel (PHP-intern representation av ett datum).
  - Om fältet är ett datumfält anges tidsstämpeln för fältdatumet vid midnatt.
  - Om fältet är ett tidsfält anges tidsstämpeln för den tiden den 1 januari 1970.
  - Om fältet är ett tidsstämpelfält kopplas FileMaker-tidsstämpelvärdet direkt till Unix-tidsstämpeln.
  - Om det angivna fältet inte är ett datum- eller tidsfält, eller om den genererade tidsstämpeln skulle ligga utanför intervallet returnerar metoden `getFieldAsTimestamp()` ett `FileMaker_Error`-objekt.
- För en specifik post använder du metoden `getContainerData()` för att returnera ett containerfältobjekt som binära data.

```
<IMG src="img.php?-url=<?php echo urlencode($record->getField('Cover
Image')); ?>">
echo $fm->getContainerData($_GET['-url']);
```

- För en specifik post använder du metoden `getContainerDataURL()` för att returnera en fullständig URL för containerfältobjektet.

```
// För bilder använder du taggen HTML img
echo '';
// För filmer och PDF-filer använder du taggen HTML embed
//echo '<embed src="'. $fm->
getContainerDataURL($record->getField('container')) .'">';
```

## Begränsa portalrader som returneras av en sökförfrågan

I en lösning som har många relaterade poster kan sökning och sortering av portalposter vara tidskrävande. För att begränsa antalet poster som visas i ett sökresultat kan du använda metoden `setRelatedSetsFilters()` för sökresultat. Metoden `setRelatedSetsFilters()` använder två argument:

- en relaterad uppsättnings filtervärde: `layout` eller `inget`.
  - Om du anger värdet `inget` returnerar Web Publishing Engine alla rader i portalen, och portalposterna sorteras inte i förväg.
  - Om du anger värdet `layout` respekteras inställningarna som har angetts i dialogrutan FileMaker Pro Portalkonfiguration. Posterna sorteras baserat på den sortering som har angetts i dialogrutan Portalkonfiguration, med postuppsättningen filtrerad för att starta med den första angivna raden.
- det lägsta antalet portalposter som returneras: ett heltal eller `all`.
  - Detta värde används endast om inställningen **Tillåt vertikal rullning** är aktiverad i dialogrutan Portalkonfiguration. Om du anger ett heltal returneras det antalet rader efter den första raden. Om du anger `all` returnerar Web Publishing Engine alla relaterade poster.
  - Om inställningen **Tillåt vertikal rullning** är inaktiverad kommer inställningen **Antal rader** i dialogrutan Portalkonfiguration att avgöra det maximala antalet relaterade poster som returneras.

**Obs!** Inställningen **Filtrera portalposter** i dialogrutan Portalkonfiguration stöds inte för PHP-frågor. Alla beräkningar som angetts för inställningen **Filtrera portalposter** ignoreras.

## Förkontrollera kommandon, poster och fält

Med FileMaker class kan du *förkontrollera* fältuppgifter i en PHP-lösning på webbservern innan du skickar uppgifterna till databasen.

När du avgör om du ska använda förkontroll eller inte, ska du tänka över antalet datavärden som webbanvändaren anger. Om användaren uppdaterar ett litet antal fält kan du förbättra prestandan genom att inte använda förkontroll. Om användaren däremot anger data i många fält kan förkontroll förhindra att användaren blir frustrerad över att en post avvisas av databasen på grund av kontrollfel.

Med FileMaker class förkontrollerar PHP-motorn följande fältvillkor:

- `ej tomt`  
Giltiga data är en icke-tom teckensträng. Dessa data måste innehålla minst ett tecken.
- `endast numeriskt`  
Giltiga data innehåller endast numeriska tecken.
- `maximalt antal tecken`  
Giltiga data innehåller högst det antal tecken som angetts.

- **fysiffrigt årtal**

Giltiga data är en teckensträng som representerar ett datum med ett fysiffrigt årtal i formatet M/D/YYYY, där M är ett tal mellan 1 och 12, D är ett tal mellan 1 och 31 och YYYY är ett fysiffrigt tal mellan 0001 och 4000. Till exempel är 1/30/3030 ett giltigt fysiffrigt årtalsvärde. Strängen 4/31/2017 är dock ett ogiltigt fysiffrigt årtalsvärde eftersom april inte har 31 dagar. Datumkontroll stöder snedstreck (/), omvänt snedstreck (\) och bindestreck (-) som avgränsare. Strängen får dock inte innehålla en blandning av avgränsarna. Till exempel är 1\30-2017 ogiltigt.

- **tidpunkt**

Giltiga data är en teckensträng som representera ett 12-timmarsvärde i ett av dessa format:

- T
- T:M
- T:M:S
- T:M:S FM/EM
- T:M FM/EM

där T är ett tal mellan 1 och 12 och M och S är tal mellan 1 och 60.

Förkontrollen i PHP-motorn stöder implicit kontroll av fältdata baserat på fälttypen.

- **datum**

Ett fält om har definierats som ett datumfält kontrolleras enligt reglerna i "fysiffrigt årtal", förutom att årtalsvärdet kan innehålla 0-4 siffror (årtalsvärdet kan vara tomt). Till exempel är 1/30 ett giltigt datum även om det inte har något årtal angivet.

- **tid**

Ett fält som har definierats som ett tidsfält kontrolleras enligt reglerna för "tidpunkt", förutom att timkomponenten (T) kan vara ett tal mellan 1 och 24 för att stödja 24-timmarsvärden.

- **tidstämpel**

Ett fält som har definierats som ett tidsstämpelfält kontrolleras enligt reglerna för "tid" för tidskomponenten och enligt reglerna för "datum" för datumkomponenten.

FileMaker class kan inte förkontrollera alla fältkontrollalternativ som är tillgängliga i FileMaker Pro. Följande kontrollalternativ kan inte förkontrolleras eftersom de är beroende av status för alla data i databasen vid det tillfälle då data verkställs:

- unikt värde
- befintligt värde
- i intervall
- medlem i värdelista
- kontrollera genom beräkning

## Förkontrollera poster i ett kommando

För ett kommandoobjekt använder du metoden `validate()` för att kontrollera ett fält eller hela kommandot mot förkontrollreglerna som drivs igenom av PHP-motorn. Om du godkänns av argumentet valfritt fältnamn förkontrolleras endast det fältet.

Om förkontrollen godkänns returnerar metoden `validate()` `TRUE`. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

## Förkontrollera poster

För ett postobjekt använder du metoden `validate()` för att kontrollera ett fält eller alla fält i posten mot förkontrollreglerna som drivs igenom av PHP-motorn. Om du godkänns av argumentet valfritt fältnamn förkontrolleras endast det fältet.

Om förkontrollen godkänns returnerar metoden `validate()` `TRUE`. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

## Förkontrollera fält

För ett fältobjekt använder du metoden `validate()` för att avgöra om ett givet värde är giltigt för ett fält.

Om förkontrollen godkänns returnerar metoden `validate()` `TRUE`. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

## Behandla kontrollfel

När förkontrollen inte godkänns innehåller det `FileMaker_Error_Validation`-objekt som returneras en matris med tre element för varje kontrollfel:

1. Det fältobjekt som inte kunde förkontrolleras
2. Ett konstant kontrollvärde som anger vilken kontrollregel som inte godkändes:
  - 1 - `FILEMAKER_RULE_NOTEMPTY`
  - 2 - `FILEMAKER_RULE_NUMERICONLY`
  - 3 - `FILEMAKER_RULE_MAXCHARACTERS`
  - 4 - `FILEMAKER_RULE_FOURDIGITYEAR`
  - 5 - `FILEMAKER_RULE_TIMEOFDAY`
  - 6 - `FILEMAKER_RULE_TIMESTAMP_FIELD`
  - 7 - `FILEMAKER_RULE_DATE_FIELD`
  - 8 - `FILEMAKER_RULE_TIME_FIELD`
3. Det verkliga värdet som angetts för det fält som inte kunde förkontrolleras



Du kan även använda följande metoder med `FileMaker_Error_Validation`-objekt:

- Använd metoden `isValidationError()` för att testa om felet är ett kontrollfel.
- Använd metoden `numErrors()` för att visa det antal kontrollregler som inte kunde godkännas.

### Exempel

```
//Skapa en Lägg till-förfrågan
$request = $fm->newAddCommand('test', array('join' => 'added', 'maxchars'
=> 'abcx', 'field' => 'something' , 'numericonly' => 'abc'));

//Kontrollera alla fält
$result = $request->validate();

//Om metoden validate() returnerade några fel skriver du ut namnet på fältet,
felnumret och det värde som inte kunde godkännas.
if (FileMaker::isError($result)) {
    echo 'Validation failed:'. "\n";
    $validationErrors= $result->getErrors();
    foreach ($validationErrors as $error) {
        $field = $error[0];
        echo 'Field Name: ' . $field->getName(). "\n";
        echo 'Error Code: ' . $error[1] . "\n";
        echo 'Value: ' . $error[2] . "\n";
    }
}
```

### Resultat

```
Kontrollen misslyckades:
Fältnamn: numericonly
Felkod: 2
Värde: abc
Fältnamn: maxchars
Felkod: 3
Värde: abcx
```

## Hantera fel

FileMaker class definierar FileMaker\_Error-objekt för att hjälpa dig att hantera fel som uppstår i en PHP-lösning.

Ett fel kan uppstå när ett kommando körs. Om ett fel uppstår returnerar kommandot ett FileMaker\_Error-objekt. Det är bra att kontrollera det fel som returneras när ett kommando körs.

Använd följande metoder för att lära dig mer om felet som anges i FileMaker\_Error-objektet.

- Testa om en variabel är ett FileMaker\_Error-objekt genom att använda metoden `isError()`.
- Hämta antalet fel som uppstått genom att använda metoden `numErrors()`.
- Hämta en matris med matriser som beskriver de fel som uppstått genom att använda metoden `getErrors()`.
- Visa ett felmeddelande genom att använda metoden `getMessage()`.

### Exempel

```
$result = $findCommand->execute();  
if (FileMaker::isError($result)) {  
    echo "<p>Error: " . $result->getMessage() . "</p>";  
    exit;  
}
```

Mer information om de felkoder som returneras med FileMaker Error-objektet finns i bilaga A, "Felkoder för anpassad webbpublicering".

# Kapitel 9

## Iscensätta, testa och övervaka en webbplats

I detta kapitel finns instruktioner för att iscensätta och testa en webbplats med anpassad webbpublicering innan den körs i en produktionsmiljö. Instruktioner finns även för att använda loggfiler för att övervaka sidan vid testning och efter den har körts igång.

### Iscensätta en anpassad webbpubliceringsplats

Innan du kan testa din webbplats ordentligt måste du kopiera och flytta de filer som behövs till korrekta platser på servern eller servrarna där du ska iscensätta webbplatsen.

1. Genomför alla steg som anges i kapitel 2, "Förbereda databaser för anpassad webbpublicering".
2. Kontrollera att anpassad webbpublicering har aktiverats och konfigurerats på rätt sätt i FileMaker Server Admin Console. Se [FileMaker Server Hjälp](#).
3. Kontrollera att webbservern och Web Publishing Engine är i gång.
4. Kopiera eller flytta dina webbplatsfiler till webbserverkomponenten i din FileMaker Server-driftsättning.

Kopiera eller flytta dina webbplatsfiler till följande katalog:

- IIS (Windows) genom HTTP eller HTTPS:  
[enhet] : \Program\FileMaker\FileMaker Server\HTTPServer\Conf  
där [enhet] är den enhet som innehåller din FileMaker Server-driftsättning.
- Apache (macOS) genom HTTP:  
/Bibliotek/FileMaker Server/HTTPServer/htdocs
- Apache (macOS) genom HTTPS:  
/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot

**Obs!** Om du aktiverar **Använd HSTS för webbklienter** i Admin Console använder du HTTPS-katalogen som värd för webbplatsens filer.

5. Om du inte redan har gjort det ska du kopiera eller flytta alla hänvisade containerfältobjekt till rätt katalog på huvuddatorn.
  - Om containerfälten innehåller de faktiska filerna i FileMaker-databasen behöver du inte flytta innehållet i containerfältet, under förutsättning att FileMaker Server fungerar som värd för databasfilen och att filen är tillgänglig på FileMaker Server.
  - Om ett containerfält i en databas lagrar en filreferens istället för en verklig fil måste det containerobjektet som referensen leder till lagras i FileMaker Pro Web-katalogen när posten skapas eller redigeras. Om du vill iscensätta din webbplats måste du kopiera eller flytta de hänvisade containrarna till en mapp med samma relativa position i rotkatalogen på webbserverprogramvaran.
  - När du använder FileMaker Pro för att överföra en databas med containerfält som har lagrat objekt externt, överförs externt lagrade containerfältdata till FileMaker Server som en del av processen. Information om hur du överför databasfiler till FileMaker Server finns i [FileMaker Pro Hjälp](#).
  - När du manuellt överför en databas som använder ett containerfält med externt lagrade objekt, måste du kopiera eller flytta de externa objekten till en undermapp till mappen RC\_Data\_FMS, enligt beskrivningen i "Containerfält med externt lagrade data" på sidan 16.
6. Kopiera alla övriga komponenter som tillhör ditt webbprogram till huvuddatorn. För anpassad webbpublicering med XML bearbetar webbprogrammet dina XML-data innan de skickas till ett annat program eller till klienten.

## Testa en anpassad webbpubliceringsplats

Innan du meddelar användarna att din anpassade webbpubliceringsplats är tillgänglig ska du kontrollera att den ser ut och fungerar som du har tänkt dig.

- Testa funktioner som sökning och att lägga till, radera och sortera poster med olika konton och behörighet.
- Kontrollera att behörighetsinställningarna fungerar som väntat genom att logga in med olika konton. Kontrollera att inga obehöriga användare kan komma åt och ändra dina data.
- Kontrollera alla scripts så att du vet att resultatet blir det väntade. Information om hur du skapar scripts som fungerar väl på webben finns i "FileMaker-scripts och anpassad webbpublicering" på sidan 18.
- Testa webbplatsen med olika operativsystem och webbläsare.
- När du skapar lösningar som använder sig av FileMaker API för PHP rekommenderar vi att du bygger lösningar som har stöd för cookies aktiverat. FileMaker-API:et för PHP har bättre svarstider när cookies är aktiverade. Cookies är inte ett krav om du ska använda funktionerna i Anpassad webbpublicering, men om du använder cookies kan Web Publishing Engine cachelagra information om sessionen.

**Obs!** Du kan visa och testa webbplatsen på huvuddatorn utan att använda en nätverksanslutning och istället ange `http://127.0.0.1/` i URL:en.

- För PHP-lösningar använder du `http://127.0.0.1/<sökväg>` där `<sökväg>` är den relativa sökvägen till startsidan för din webbplats.

- Mer information om URL-syntaxen i XML-lösningar finns i "URL-syntax för XML-data och containerobjekt" på sidan 25.

## Formatmallar som du kan använda när du testar XML-utdata

### Exempel

Här är två exempel på XSLT-formatmallar som du kan använda när du vill testa XML-utdata.

Följande formatmall skickar efterfrågade XML-data utan att göra någon konvertering. Denna formatmall är bra när du vill visa den faktiska XML-information Web Publishing Engine använder.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <xsl:copy-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

När du debuggar en formatmall kan du använda följande exempel på en HTML <textarea>-markör när du vill visa det XML-källdokument som hämtades via formatmallen i ett textblåddringsfält.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="html"/>
  <html>
    <body>
      <xsl:template match="/fmrs:fmresultset">
        <textarea rows="20" cols="100">
          <xsl:copy-of select="."/>
        </textarea><br/>
      </xsl:template>
    </body>
  </html>
</xsl:stylesheet>
```

## Övervaka webbplatsen

Du kan använda följande typer av loggfiler när du vill övervaka den anpassade webbpubliceringsplatsen och hämta information om de webbanvändare som besöker webbplatsen:

- Webbserveråtkomst och felloggar
- Web Publishing Engine-loggen
- Fellogg för webbservermodulen
- Tomcat-loggar

### Använda webbserveråtkomst och felloggar

IIS (Windows): Webbservern Microsoft IIS genererar en åtkomstloggfil och visar felen i Loggboken (Windows) i stället för att skriva dem till en loggfil. Åtkomstloggfilen, som är i formatet W3C Extended Log File Format som standard, är en lista med alla inkommande HTTP-förfrågningar till webbservern. Du kan även använda formatet W3C Common Logfile Format för åtkomstloggen. Se dokumentationen till Microsoft IIS-webbservern.

Apache (macOS): Webbservern Apache genererar en åtkomstloggfil och en felloggfil. Åtkomstloggfilen på Apache, som är i formatet W3C Common Logfile Format som standard, är en post över alla inkommande HTTP-förfrågningar till webbservern. Felloggen på Apache är en lista med problem som rör bearbetningen av HTTP-förfrågningar. Se dokumentationen till Apache-webbservern.

**Obs!** Mer information om formaten W3C Common Logfile Format och W3C Extended Log File Format finns på webbplatsen [www.w3.org](http://www.w3.org).

### Använda Web Publishing Engine-loggen

Som standard genererar Web Publishing Engine en loggfil med namnet wpe.log som innehåller en förteckning över alla Web Publishing Engine-fel som har inträffat, inklusive applikationsfel, användningsfel och systemfel. Du kan även låta Web Publishing Engine ta med information som hör till Anpassad webbpublicering, såsom XML-förfrågningar från slutanvändare om att generera webbpubliceringsmaterial eller ändringar av inställningarna för Anpassad webbpublicering.

Filen wpe.log finns i följande katalog på Web Publishing Engine-komponenten i FileMaker Server-installationen:

- Windows: [enhet]:\Program\FileMaker\FileMaker Server\Logs\wpe.log  
där [enhet] är den primära enhet där programmet startas.
- macOS: /Bibliotek/FileMaker Server/Logs/wpe.log

## Inställningar för Web Publishing Engine-loggen

Filen wpe.log genereras om inställningen **Aktivera loggning för webbpublicering** har aktiverats i Admin Console.

Loggningsalternativ aktiverat	Information som registreras i wpe.log
Felnivåmeddelanden	Alla fel i Web Publishing Engine som har inträffat, inklusive applikationsfel, användningsfel och systemfel.
Informations- och felnivåmeddelanden	Alla fel som beskrivs ovan samt information om åtkomst till Web Publishing Engine. Den innehåller en förteckning över alla XML-förfrågningar från slutanvändare om att generera utdata från Anpassad webbpublicering.

Inställningen **Felnivåmeddelanden** är aktiverad som standard. Mer information om hur du ställer in dessa alternativ med Admin Console finns i [FileMaker Server Hjälp](#).

**Viktigt!** Med tiden kan filen wpe.log bli väldigt stor. I Admin Console kan du ange en maximal storlek för filen wpe.log. När filen wpe.log uppnår sin maxstorlek skapar Web Publishing Engine en säkerhetskopia av den med namnet wpe.log.1, samtidigt som den skapar en ny version av filen wpe.log. Om du vill behålla fler säkerhetskopior kopierar du filen wpe.log.1 med jämna mellanrum till ett eget arkiv.

## Formatet i Web Publishing Engine-loggen

Filen wpe.log använder följande format i varje post:

```
[TIMESTAMP_GMT] [WPC_HOSTNAME] [CLIENT_IP:PORT] [ACCOUNT_NAME] [MODULE_TYPE]
[SEVERITY] [FM_ERRORCODE] [RETURN_BYTES] [MESSAGE]
```

där:

- [TIMESTAMP\_GMT] är datum och tidpunkt för posten enligt Greenwich Mean Time (GMT).
- [WPC\_HOSTNAME] är namnet på huvuddatorn.
- [CLIENT\_IP:PORT] är IP-adressen och porten på den klient som XML-förfrågan kom ifrån.
- [ACCOUNT\_NAME] är kontonamnet som används för loggning i den värdbaseade FileMaker-databasen.
- [MODULE\_TYPE] är antingen: XML, för förfrågningar med Anpassad webbpublicering med XML, eller PHP, för förfrågningar med Anpassad webbpublicering med PHP.
- [SEVERITY] är antingen INFO, som visar att det är ett informerande meddelande, eller ERROR, som visar att det är ett felmeddelande.
- [FM\_ERROR\_CODE] är felnumret som returneras för ett felmeddelande. Felnumret kan vara en felkod för FileMaker-databaser (se "Felkodsnummer för FileMaker-databaser" på sidan 99). Dessutom kan felnumret vara ett HTTP-felnummer, som föregås av "HTTP:".
- [RETURN\_BYTES] är det antal byte som returneras av förfrågan.
- [MESSAGE] ger ytterligare information om loggposten.

## Meddelanden i Web Publishing Engine-loggen

### Exempel

Följande exempel visar vilka typer av meddelanden som kan ingå i filen wpe.log.

#### När Web Publishing Engine startas och stoppas:

```
2017-06-02 15:15:31 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine started.
2017-06-02 15:46:52 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine stopped.
```

#### Lyckade eller misslyckade XML-förfrågningar:

```
2017-06-02 15:21:08 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML INFO
0 3964 "/fmi/xml/fmresultset.xml?-db=Contacts&-lay=Contact_Details&-
findall"
2017-06-02 15:26:31 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR 5 596 "/fmi/xml/fmresultset.xml?-db=Contacts&-
layout=Contact_Details&-findall"
```

#### Scriptfel:

```
2017-06-02 17:33:12 -0700 WPC_SERVER 192.168.100.101:0 jdoe - ERROR
4 - Web Scripting Error: 4, File: "10b_MeetingsUpload", Script: "OnOpen",
Script Step: "Show Custom Dialog"
```

#### Ändringar i inställningarna till Anpassad webbpublicering:

```
2017-06-09 10:59:49 -0700 WPC_SERVER 192.168.100.101:0 jdoe - INFO
- - XML Web Publishing Engine is enabled.
```

#### Systemfel:

```
2017-06-02 15:30:42 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR - - Communication failed
```

## Använda felloggen för webbservern

Om webbservern inte kan ansluta till Web Publishing Engine skapar webbservermodulen en loggfil över eventuella fel som uppstår vid bearbetningen. Filen heter web\_server\_module\_log.txt och finns i mappen Logs i FileMaker Server-mappen på webbservervärdet.



## Använda Tomcat-loggar

När det har uppstått ett problem i FileMaker Server som orsakas av ett internt webbserverfel kan det hjälpa att ta en titt i Tomcat-loggarna. Tomcat-loggarna finns på webbserverkomponenten i FileMaker Server-driftsättningen:

- Windows:
  - [enhet]:\Program\FileMaker\FileMaker Server\Admin\admin-master-tomcat\logs\  
där [enhet] är den primära enhet där programmet startas.
  - [enhet]:\Program\FileMaker\FileMaker Server\Web Publishing\publishing-engine\jwpc-tomcat\logs  
där [enhet] är den primära enhet där programmet startas.
- macOS:
  - /Bibliotek/FileMaker Server/Admin/admin-master-tomcat/logs/
  - /Bibliotek/FileMaker Server/Web Publishing/publishing-engine/jwpc-tomcat/logs

# Bilaga A

## Felkoder för anpassad webbpublicering

Webbpubliceringsmotorn genererar felkoder för fel i databas- och frågesträngar som kan uppstå under en XML-dataförfrågan.

En lista med uppdaterade felkoder hittar du i [FileMaker Knowledge Base](#).

### Felkodsnummer i XML-format

Web Publishing Engine genererar en felkod för databaser som är publicerade i XML-format varje gång data efterfrågas. Denna typ av felkod infogas i början av XML-dokumentet i elementet `<error code>` för `fmresultset`-syntaxen eller i elementet `<ERRORCODE>` för `FMPXMLRESULT`- eller `FMPXMLLAYOUT`-syntaxen. Felkod 0 betyder att inget fel har inträffat.

#### Exempel

**Databasfelkod i syntaxen `fmresultset`:**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN"
"http://192.168.123.101/fmi/xml/fmresultset.dtd">
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset"
version="1.0">
  <error code="0"></error>
```

**Databasfelkod i syntaxen `FMPXMLRESULT`:**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN"
"http://192.168.123.101/fmi/xml/FMPXMLRESULT.dtd">
<fmpxmlresult xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
```

Som utvecklare av en anpassad webbpubliceringslösning måste du kontrollera värdet för elementet `<error code>` eller `<ERRORCODE>` och hantera det på rätt sätt. Web Publishing Engine hanterar inte databasfel.

## Felkodsnummer för FileMaker-databaser

Information om felkoder i FileMaker Pro finns i [FileMaker Pro Hjälp](#).

FileMaker Server ger felkod 959 för att indikera att en teknik har inaktiverats.

Om till exempel serveradministratören inaktiverar Anpassad webbpublicering med XML på fliken **Webbpublicering > XML** i FileMaker Server Admin Console returnerar XML-frågor felkoden 959.

### Exempel

Felkod 959 i syntaxen FMPXMLLAYOUT:

```
<FMPXMLLAYOUT>
  <ERRORCODE>959</ERRORCODE>
  <LAYOUT DATABASE=" " NAME=" " />
  <VALUELISTS/>
</FMPXMLLAYOUT>
```

# Index

## A

- add()-metoden 83
- addSortRule()-metoden 81
- Admin Console 14, 25, 67
- aktivera anpassad webbpublicering i en databas 13
- Ändra lösenord, scriptsteg 14
- ange layout vid fråga om XML-data 40
- Anpassad Web Publishing Engine (CWPE) 24
- Anpassad webbpublicering
  - aktivera i en databas 13
  - aktivera i Web Publishing Engine 14
  - använda en statisk IP-adress 22
  - använda scripts 18
  - begränsa åtkomst av IP-adresser på webbserver 14
  - definition 9
  - Gäst, konto 14
  - krav för 21
  - med PHP 11
  - med XML 11, 23
  - nya funktioner i 21
  - överblick 9
  - plugin-program för webbpubliceringslösningar 18
  - scripts 19
  - utökad behörighet för 13
  - webbanvändares åtkomst till lösningar 13
- ansluta till en FileMaker-databas med PHP 72
- användarnamn
  - ange för anpassad webbpublicering 13
  - grundläggande verifiering av webbanvändare 13
- ASCII-tecken, i XML-dokument 38
- åtkomstloggfiler för webbserver, beskrivning 94
- automatisk inmatning av attribut 31

## B

- begäran om XML-data 25
- begränsa portalfältposter 59
- behandla en webbpubliceringsförfrågan 10
- behandla ett sökresultat 85
- behörighet 14
- behörighetsuppsättning, tilldela för anpassad webbpublicering 13
- byta layout för ett XML-svar 40

## C

- clearSortRules()-metoden 81
- client URL library 65
- commit()-metoden 73

- containerfält
  - hur webbanvändare använder data 17
  - med externa filer 16
  - med externt lagrade data 16
  - progressiv nedladdning 17
  - publicera innehåll i 15
  - URL-syntax för åtkomst i XML-lösningar 26
- createRecord()-metoden 73
- cURL 65

## D

- databas, skydda publicerad 14
- databaser, felkoder 29, 98
- databasobjekt 72
- <datasource>-element 30
- datumfält 87
- datumrepresentation 85
- db, frågeparameter 51
- dbnames, frågekommando 47
- delete, frågekommando 47
- delete.related, frågeparameter 46
- delete()-metoden 74, 79
- dokumentation online 8
- dokumentation, FileMaker 8
- dokumenttypdefinitioner (DTER) 29, 33
- dup, frågekommando 48
- duplicera en post 73
- dynamisk IP-adress 65

## E

- edit, frågekommando 48
- element
  - databaser, felkoder 29
  - i FMPXMLLAYOUT-syntaxen 35
  - i FMPXMLRESULT-syntaxen 33
  - i syntaxen fmresultset 30
- endast numeriskt fält 86
- endast numeriskt, attribut 31
- <error code>- och <ERRORCODE>-element 98
- et-tecken, i PHP 72
- exempel på
  - genererad FMPXMLLAYOUT-syntax 37
  - genererad FMPXMLRESULT-syntax 34
  - genererad fmresultset-syntax 32
- exportera XML-data 23
- Extensible Markup Language (XML). Se XML

**F**

## fält

- attribut 31
- beräkning 28, 29
- container 15, 26, 33
- datum 33, 36, 87
- ej tomt 86
- endast numerisk 86
- fullständigt fältnamn 45
- fysiffrigt årtal 87
- global 29
- maximalt antal tecken 86
- nummer 33
- portal 31
- portaler 45
- relaterade i PHP 78
- relaterade i XML 31, 45
- repeterade 26, 45
- statistik 28, 29
- text 33
- tid 33, 36, 87
- tidpunkt 87
- tidstämpel 33, 36, 87
- fält med fysiffrigt årtal 87
- fält med maximalt antal tecken 86
- fältnamn, frågeparameter (ej container) 52
- fältnamn, fullständig syntax 45
- fältnamn.op, frågeparameter 53

## fel

- beskrivning 98
- databas, felkodsnummer 98
- databaser, felkodelement 29
- hantera 90
- loggfiler för webbserver 94

## felsöka

- Anpassad webbpublicering, webbplatser 92
- XML-dokumentåtkomst 41

## -field, frågeparameter (container) 52

## &lt;field-definition&gt; element 31

## FileMaker API för PHP 11

- manuell installation 66
- referens 70

## FileMaker class 71

## FileMaker class-objekt

- databas 72
- definition 71
- post 73
- relaterad uppsättning 78

## FileMaker kommandoobjekt

- Duplicera 73
- Lägga till 73
- Radera 74
- Redigera 74
- Sammanfattad sökning 83
- Söka 81, 82
- Söka efter alla 82
- Visa alla 82

## FileMaker Pro, jämfört med Web Publishing Engine 23

## FileMaker Server Admin Console 14, 25

## FileMaker Server-dokumentation 8

## FileMaker WebDirect 9

## filtrera portalfältposter 58

## -findall, frågekommando 49

## -findany, frågekommandot 49

## -findquery, frågekommando 49

## fmphp, nyckelord för aktivering av PHP-publicering 13

## fmresultset, syntax 23, 28, 29–32

## fmsadmin, grupp 17

## fmxml, nyckelord för aktivering av XML-publicering 13, 25

## förkontroll

- datum 87
- ej tomt 86
- endast numerisk 86
- fält 88
- fysiffrigt årtal 87
- kommandon 86
- maximalt antal tecken 86
- poster 88
- tid 87
- tidpunkt 87
- tidstämpel 87

## formatmallar, testa 92

## fråga portalfält 46

## frågekommandot -find.. 49

## frågesträngar

- efterfråga XML-data 38, 42
- fullständigt fältnamn, syntax för 45
- globala fält, syntax för 47
- kommandon och parametrar 38, 42
- lägga till poster till portaler 45
- redigera poster i portaler 46
- riktlinjer för 43

## fullständigt fältnamn, syntax för 45

## fysiffrigt årtal, attribut 31

**G**

## Gäst, konto

- aktivera 14
- inaktivera 14
- med Direkt webbpublicering 14

## getContainerData()-metoden 85

## getContainerDataURL()-metoden 85

## getDatabase()-metoden 77

## getErrors()-metoden 90

## getFetchCount()-metoden 85

## getField()-metoden 85

## getFieldAsTimestamp()-metoden 85

## getFields()-metoden 77, 85

## getFoundSetCount()-metoden 85

## getLayout()-metoden 77

## getMessage()-metoden 90

## getName()-metoden 77, 78

## getRange()-metoden 81

## getRecords()-metoden 85

## getRelatedSet()-metoden 78

## getRelatedSets()-metoden 78

getValueListsTwoFields()-metoden 80  
 getValueListTwoFields()-metoden 80  
 globala fält  
   i fältdefinitioner 31  
   syntax för 47  
 grundlägg. verifiering av webbanv. 13

## H

hämta  
   layoutinformation 51  
   layoutnamn 50  
   tillgängliga scriptnamn 51  
 hantera fel 90  
 hostspec-egenskap för PHP 72  
 HSTS med PHP 69  
 HTML-formulär för XML-förfrågningar 25  
 HTTP-katalog för PHP 69  
 HTTPS-katalog för PHP 69

## I

icke-tomt attribut 31  
 icke-tomt fält 86  
 importera XML-data 23  
 Initial radinställning 58  
 installation av FileMaker API för PHP 66  
 Inställningen Antal rader 58, 86  
 Inställningen Filtrera portalposter 58, 86  
 isError()-metoden 90  
 isValidatError()-metoden 89

## J

jämförelse av XML-syntaxer 28  
 jämförelseoperatorer för fält 53

## K

kodning  
   URL-adresser 27  
   XML-data 28, 38  
 kommandon för frågor. Se frågesträngar  
 Kommandot Duplicera 73  
 Kommandot Lägg till 73  
 Kommandot Radera 74  
 Kommandot Redigera 74  
 Kommandot Sök efter alla 82  
 Kommandot Utför sökning 82  
 Kommandot Visa alla 82  
 konton och behörighet  
   aktivera för anpassad webbpublicering 13  
   Gäst, konto 14  
   scripts 18

kontroll  
   datum 87  
   ej tomt 86  
   endast numerisk 86  
   fält 88  
   fysiffrigt årtal 87  
   kommandon 86  
   maximalt antal tecken 86  
   poster 88  
   tid 87  
   tidpunkt 87  
   tidstämpel 87  
 kontroll av befintligt värde 87  
 kontroll av medlem i värdelista 87  
 kontroll av unikt värde 87  
 kontroll av värden i intervall 87  
 kontrollera genom beräkning 87  
 krav för anpassad webbpublicering 21

## L

Latin-1-kodning 69  
 -lay, frågeparameter 40, 54  
 -lay.response, frågeparameter 40, 54  
 layouter  
   använda i PHP 77  
   växla för ett XML-svar 40  
 -layoutnames, frågekommando 50  
 listFields()-metoden 77  
 listLayouts()-metoden 77  
 listRelatedSets()-metoden 77, 78  
 listScripts()-metoden 75  
 listValueLists()-metoden 77, 80  
 Logga in igen, scriptsteg 14  
 loggfiler 92  
   beskrivning 94  
   Tomcat 97  
   web\_server\_module\_log.txt 96  
   webbserveråtkomst 94  
 -lop, frågeparameter 55  
 lösenord  
   Ändra lösenord, scriptsteg 14  
   ange för anpassad webbpublicering 13  
   grundläggande verifiering av webbanvändare 13  
   inget lösenord för inloggning 14

## M

macOS Server Admin 65  
 manuell installation av FileMaker API för PHP 66  
 -max, frågeparameter 55  
 max-characters, attribut 31  
 max-repeat, attribut 31  
 <metadata>-element 31

## metoder

- add() 83
- addSortRule() 81
- clearSortRules() 81
- commit() 73
- createRecord() 73
- delete() 74, 79
- getContainerData() 85
- getContainerDataURL() 85
- getDatabase() 77
- getErrors() 90
- getFetchCount() 85
- getField() 85
- getFieldAsTimestamp() 85
- getFields() 77, 85
- getFoundSetCount() 85
- getLayout() 77
- getMessage() 90
- getName() 77, 78
- getRange() 81
- getRecords() 85
- getRelatedSet() 78
- getRelatedSets() 78
- getValueListsTwoFields() 80
- getValueListTwoFields() 80
- isError() 90
- isValidationError() 89
- listFields() 77
- listLayouts() 77
- listRelatedSets() 77, 78
- listScripts() 75
- listValueLists() 77, 80
- newAddCommand() 73
- newCompoundFindCommand() 83
- newDeleteCommand() 74
- newDuplicateCommand() 73
- newEditCommand() 74
- newFindAllCommand() 82
- newFindAnyCommand() 82
- newFindCommand() 82
- newFindRequest() 83
- newPerformScriptCommand() 75
- newRelatedRecord() 79
- numErrors() 89, 90
- setLogicalOperator() 81
- setPreCommandScript() 75, 82
- setPreSortScript() 76, 82
- setProperty() 72
- setRange() 81
- setRelatedSetsFilters() 86
- setResultsLayout() 77
- setScript() 76, 82
- validate() 87

MIME-typer (Multipurpose Internet Mail Extensions) 15

-modid, frågeparameter 56

**N**

- namnattribut 31
- namnuttryck för XML 28
- new, frågekommando 50

- newAddCommand()-metoden 73
- newCompoundFindCommand()-metoden 83
- newDeleteCommand()-metoden 74
- newDuplicateCommand()-metoden 73
- newEditCommand()-metoden 74
- newFindAllCommand()-metoden 82
- newFindAnyCommand()-metoden 82
- newFindCommand()-metoden 82
- newFindRequest()-metoden 83
- newPerformScriptCommand()-metoden 75
- newRelatedRecord()-metoden 79
- numErrors()-metoden 89, 90
- nummer för felkoder i databaser 98
- nya funktioner i anpassad webbpublicering 21
- nyckelord för aktivering av anpassad webbpublicering 13, 25

**O**

- operatorer, jämförelse 53
- ordningsföljd för XML-frågebearbetning 41
- överblick
  - Anpassad webbpublicering 9
  - steg för åtkomst av XML-data 25
- övervaka webbplatser 94

**P**

- parametrar för frågor. Se frågesträngar
- PDF-filer 8
- PHP
  - aktivera i en databas 13
  - Anpassad webbpublicering, beskrivning 11
  - version som stöds 66
- PHP-fördelar 12
- PHP-version 65
- plugin-program 18
- portaler
  - antal poster 58
  - använda i PHP 78
  - inledande rad 58
  - lägga till poster 45
  - layout 58
  - radera poster 46
  - redigera poster 46
  - sortera poster 58
- portalfältfrågor 58, 59
- poster
  - duplicera i PHP 73
  - duplicera i XML 48
  - hoppa över i XML 62
  - radera i PHP 74
  - radera i XML 47
  - redigera i PHP 74
  - redigera i XML 48
  - skapa i PHP 73
  - skapa i XML 50
  - söka efter i PHP 81
  - söka efter i XML 49
- postobjekt 73

- programlogg 94
- progressiv nedladdning 17
- publicera på webben
  - ansluta till Internet eller intranät 22
  - använda XML 25
  - containerfältobjekt 15
  - databaser, felkoder 98
  - krav för 21
  - skydda databaser 14

## Q

- query, frågeparameter 56

## R

- radera en post 74
- radera portalposter 46
- recid, frågeparameter 57
- redigera en post 74
- referensinformation 70
- <relatedset-definition> element 31
- relatedsets.filter, frågeparameter 58
- relatedsets.max, frågeparameter 59
- relaterat uppsättningsobjekt 78
- resultatattribut 31
- <resultset>-element 31

## S

- säkerhet
  - begränsa åtkomst från IP-adresser 14
  - dokumentation 11
  - konton och lösenord 14
  - riktlinjer för att skydda publicerade databaser 14
- Sammanfattad sökning
  - exempel 84
  - kommando 83
- sammanfattad sökning
  - frågekommando 49
  - frågeparameter 56
- sammanfattning av steg för åtkomst av XML-data 25
- SAT. Se FileMaker Server Admin Console
- script, frågeparameter 59
- script.param, frågeparameter 60
- script.prefind, frågeparameter 60
- script.prefind.param, frågeparameter 60
- script.presort, frågeparameter 61
- script.presort.param, frågeparameter 61
- scriptnames, frågekommando 51
- scripts
  - använda i PHP 75
  - använda i XML-förfrågningar 25
  - i anpassad webbpublicering 18
  - konton och behörighet 18
  - tips och råd 18
- scriptsteg
  - Ändra lösenord 14
  - Logga in igen 14

- scripttriggers 19
- serverkrav 64
- setLogicalOperator()-metoden 81
- setPreCommandScript()-metoden 75, 82
- setPreSortScript()-metoden 76, 82
- setProperty()-metoden 72
- setRange()-metoden 81
- setRelatedSetsFilters()-metoden 86
- setResultsLayout()-metoden 77
- setScript()-metoden 76, 82
- skapa en post
  - använda PHP 73
  - använda XML 50
- skip, frågeparameter 62
- skydda publicerade databaser 14
- slut på minne, fel 18
- Sökkommandoobjekt 81
- sökresultat 85
- sortera portalfältposter 58
- sortfield, frågeparameter 62
- sortorder, frågeparameter 63
- SSL-kryptering (Secure Sockets Layer) 15
- statisk IP-adress 65
- statisk publicering, definition 9
- syntax för XML, beskrivning 28
- Syntaxen FMPXMLLAYOUT 23, 28, 35–37
- Syntaxen FMPXMLRESULT 23, 28, 33–34

## T

- testa
  - webbplatser 92
  - XML-utdata 93
- textkodning
  - genererade XML-data 28
  - URL-adresser 27
- tidpunkt, attribut 31
- tidpunktsfält 87
- tidsfält 87
- tidsstämpelfält 85, 87
- Tillåt Inställningen Vertikal rullning 58, 86
- tillgängliga databaslayouter 50
- tillgängliga scripts 51
- Tomcat, använda loggfiler 97
- typtribut 31

## U

- UAC. Se FileMaker Server Admin Console
- Unicode
  - dataformat som returneras av FileMaker Server 69
  - tecken som används i XML-tolkar 38
- Unix-tidsstämpel 85
- URL-syntax för
  - containerobjekt i XML-lösningar 26
  - XML-förfrågningar 25
- URL-textkodning 27
- UTF-8-format (Unicode Transformation 8 Bit) 27, 38



UTF-8-kodning 69  
utföra sökningar 81  
utökad behörighet för anpassad webbpublicering 13

## V

validate()-metoden 87  
värdelistor  
    använda i PHP 80  
    använda i XML 35  
verifiering av webbanvändare 13  
Verktuget Server Admin. Se macOS Server Admin  
-view, frågekommando 51

## W

Web Publishing Engine  
    Admin Console 25  
    behandling av förfrågan 10  
    beskrivning 10  
    fördelar med 20  
    generera XML-data 24  
    genererade felkoder 98  
    programlogg 94  
    skapa XML-dokument 25  
web\_server\_module\_log.txt, loggfil 96  
webbanvändare  
    använda containerfältdata 17  
    åtkomst till skyddade databaser 13  
    krav för åtkomst till anpassade  
        webbpubliceringslösningar 21  
webbläsarens roll i XML-förfrågningar 24  
webbmapp, kopiera containerfältobjekt 16  
webbplatser  
    FileMakers supportsidor 8  
    övervaka 94  
    skapa med Web Publishing Engine 20  
    testa 92  
Webbpubliceringskärna illustrerad 24  
webbserver  
    loggfiler 94  
    MIME-typer, stöd för 15  
    roll i XML-förfrågningar 24

## X

### XML

aktivera i en databas 13  
Anpassad webbpublicering, beskrivning 11  
dokumenttypdefinitioner (DTDer) 29, 33  
efterfråga data 25  
felsöka åtkomst till XML-dokument 41  
FMPXMLLAYOUT-syntax 35  
FMPXMLRESULT-syntax 33  
fmresultset, syntax 30  
    <datasource>-element 30  
    <field-definition>-element 31  
    <metadata>-element 31  
    <relatedset-definition>-element 31  
    <resultset>-element 31  
fördelar 11  
förfrågan, ange layout 40  
frågesträngar 38, 42  
generera XML-data från förfrågan 24  
kodad med UTF-8-format 28, 38  
namnuttryck för 28  
ordningsföljd för frågebearbetning 41  
sammanfattning av steg för åtkomst av XML-data 25  
svar, växla layouter 40  
syntaxer, jämförelse 28  
tolkar 25, 38  
URL-textkodning 27  
XML 1.0-specifikation 23  
<xsl:stylesheet>-element 93  
<xsl:template> element 93