

Hur Mspecs pluginen fungerar

När pluginen aktiveras och när man laddar en sida med pluginen aktiverad går wordpress till filen *mspecs.php* och kör all kod där. Det är så allting startar. I den filen laddas alla classes och models in samt vissa controllers. Detta sker på **alla** sidor man besöker på hemsidan.

När man vill se en specifik sida till exempel *till salu* måste den sidan finnas i wordpress och som text måste den ha en **shortcode** som hämtar in bostäder från Mspecs. En **shortcode** är ett sätt att trigga PHP kod när en wordpress sida laddas. En shortcode kan se ut så här:

```
[mspecsapi api="getDeals" query="q=isActive=true and objectStatus IN ('ENUMS_OBJECTSTATUS_TYPE_IN_PROGRESS', 'ENUMS_OBJECTSTATUS_TYPE_FOR_SALE') and isInternational=false" sort="-updatedAt" view="deals"]
```

När man går in på en sida med en shortcode körs funktionen **render** i *shortcode.php* som tar fram det man vill se, t.ex. vilka hus som är till salu beroende på vilka parametrar man har ställt in.

Om man trycker på ett objekt som är till salu körs **init** i *frontend.php* och man kommer till en ny sida som visar objektet. Denna fil körs också om man vill se en bostad eller en visning men de två sistnämnda används sällan/aldrig. **Init** gör detta genom att ladda in den fil som har specificerats, från mappen *views* t.ex. *deals.phtml*. Värt att nämna med dessa views är att de alla använder en variabel som aldrig är deklarerad i den filen. Den variabeln kommer ifrån **extract** metoden i *controller.php*. **extract** skapar variabeln dynamisk och ger den alla värden som behövs. T.ex. i *deal.phtml* finns variabeln *\$deal* som har all info om objektet men den är aldrig deklarerad i *deal.phtml*. Värt att nämna om phtml filerna är att de kallar många metoder som inte är deklarerade. De kallen hamnar i **__call** i *model.php*.

För att ändringar ska synas på ett objekt måste man trycka på *uppdatera hemsidan* på Mspecs. Då går Mspecs till den länk som har specificerats i företagsinställningar. Den länken går till *{din hemsida}/wp-admin/admin-ajax.php* tillsammans med lite information (se länken i företagsinställningar). Det *admin-ajax.php* gör är att den kallar på den specificerade **action**. Den specificerade **action** kan ligga i vilken fil som helst men länkarna på Mspecs använder **action=mspecsapi** som är deklarerad i *ajax.php*. Det är från denna fil som cachan töms och deals uppdateras m.m.

Hur Mspecs och pluginen interagerar med varandra

Alla interaktioner som startar via något man gör på Mspecs går till *wp-admin/admin-ajax.php* på hemsidan med pluginen som i sin tur triggar den specificerade action. På Mspecs är alla länkar under företagsinställningar inställda på att trigga **action=mspecsapi**. När Mspecs kör alla dessa länkar hamnar de i **render** metoden i *ajax.php*. Om man ska avpublicera ett objekt körs metoden **deleteRequest**. Annars körs **getRequest**. Följande interaktioner kan triggas ifrån Mspecs:

Cache

Dessa interaktioner har **api=cache** i länken och kör metoden **cache** i *ajax.php*:

- **Uppdatera hela hemsidan** – När man trycker på denna knapp går Mspecs till den länken som är inskriven i *URL för att uppdatera hela hemsidan* under företagsinställningar. När länken körs och den till sist hamnar i **cache** metoden tömmer den cachen dvs. tar bort alla filer i cachemapparna.
- **Publicera/Uppdatera/Avpublicera ett objekt** – När man trycker på denna knapp går Mspecs till länken som är inskriven i **URL för att uppdatera uppdrag på hemsidan**. Se flödesschema *När ett objekt publiceras/uppdateras/tas bort*.
- **Skapa/uppdatera ett bud** – under budgivning på Mspecs kan man skapa nya bud och ändra tidigare. När man gör något av detta går Mspecs till länken som är inskriven i **URL för att uppdatera bud på hemsidan**. Man behöver alltså inte uppdatera manuellt. När länken körs och till sist hamnar i **cache** laddas hela budgivningen in på nytt och sparas i cachen.

Övriga

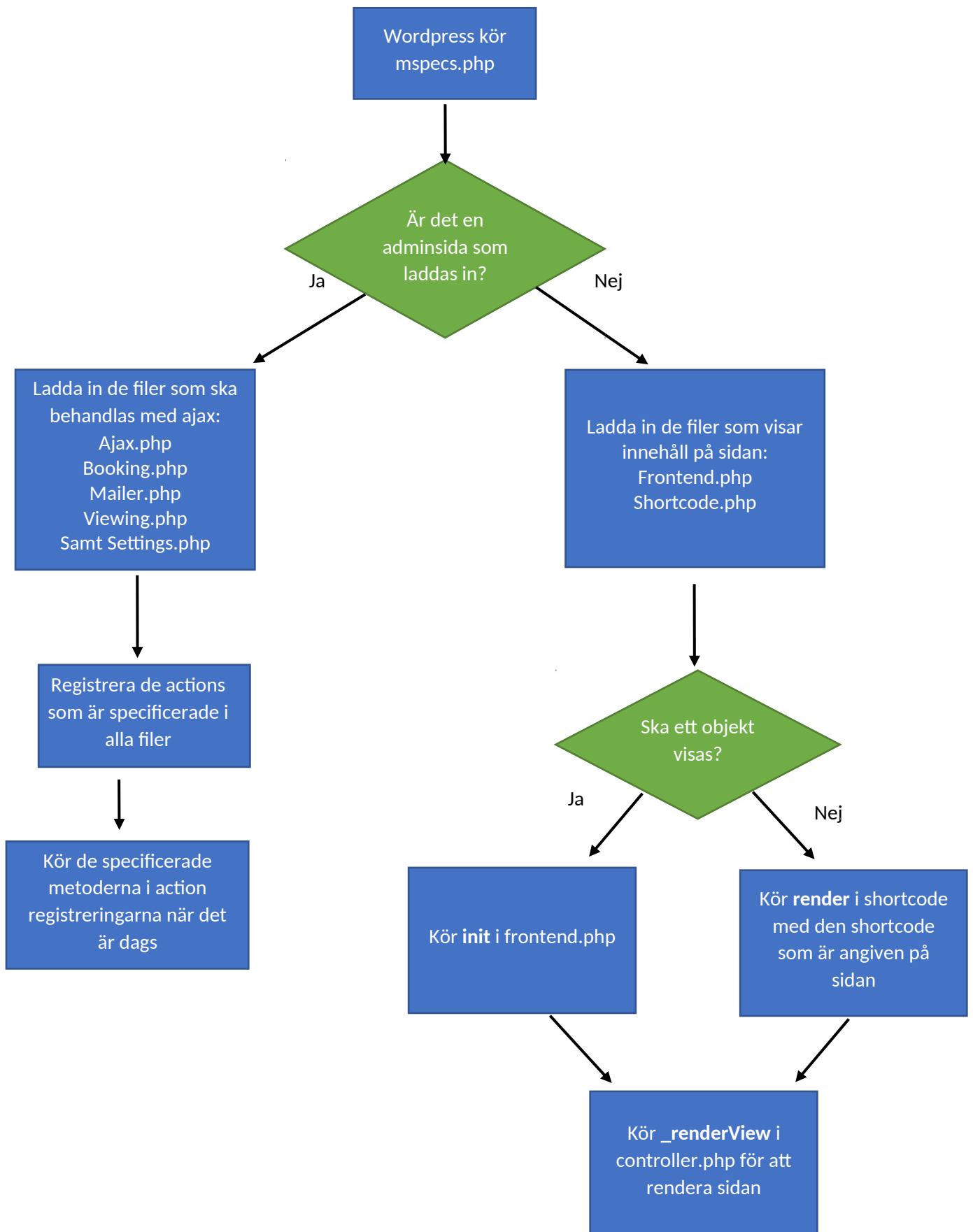
Dessa interaktioner har **api={något annat}**:

- **Visa objekt på hemsidan** – Under publicera på Mspecs finns en länk till objektet på hemsidan (om den är publicerad). Den länken hämtas ifrån företagsinställningarna och när man trycker på den kommer man till objektet på hemsidan med pluginen. Den länken har **api=dealPage**. Det pluginen gör är att den hämtar länken för det objektet och skickar den vidare till den länken.
- **Visa ett objekts bilder på hemsidan** – gör samma sak som ovanstående punkt men använder **api=dealImage** istället och har en annan länk. Det som skiljer dem åt är att denna länk går till bilderna.

Hur pluginen interagerar med Mspecs

De enda gångerna pluginen vill interagera med Mspecs är när den inte kan hitta data i cachen. Då hämtar den data som den behöver med hjälp av metoden **query/queryOne** i *api.php*.

Varje gång en sida laddas in



När ett objekt publiceras/uppdateras/tas bort

