

## Appendiks C: CMS3 Plugin udviklings guide

*Dette dokument beskriver hvor plugins til CMS3 systemet skal udvikles, og gennemgår nogle af de interfaces man kan implementere.*

### Forord

Dette dokument beskriver udviklingen af plugins til CMS3 systemet. Den er rettet mod tredjeparts udviklere, der har lyst til at udvikle plugins til CMS3. Dokumentet er blevet til efter implementeringen af CMS3, men plugin arkitekturen har været defineret før implementeringen. Plugins udviklet i overensstemmelse med denne vejledning, burde kunne afvikles på en CMS3 installation. Denne vejledning forklarer nogle af de mest basale ting for plugin udvikling, mere avancerede features som f.eks. ISystemCallback er kun beskrevet i dokumentationen.

### Indledning

I CMS3 Systemet består stortset alle features af plugins. Plugins kan pakkes som .c3p pakker, formatet er specificeret i Appendiks D: CMS3 Pakke specifikation. Da alle plugins skal integreres med CMS3 systemet, og hinanden er det vigtigt at der ikke opstår konflikter, da sådanne vil kunne ødelægge et helt system. Derfor skal navngivnings konvention (Appendiks A) og filsystems konvention (Appendiks B) overholdes. Det anbefales af man læser disse før man begynder på plugin udvikling. Plugin udvikling kræver desuden en god forståelse af php5, erfaring med Ajax er også en meget god idé.

### Definition af plugin

Et plugin er per definition i CMS3, en klasse der hedder CMS3\_<PluginID>, implementere IPlugin og ligger i filen ./lib/<PluginID>/CMS3\_<PluginID>.php. Hvilket også er beskrevet i Appendiks A og B. Dette er minimum kravene til et plugin, derudover må et plugin implementere end lang række andre interfaces, som afgør hvordan den integreres med resten af systemet. Et plugin skal ifølge pakke format specifikationen (Appendiks D) også have en list med tjeksummer og OpenPGP kompatibel signatur af denne i mappen ./data/PackageManager/checksums/. Dette er kun nødvendigt hvis pluginet skal kunne installeres, verificeres og/eller fjernes med pakkemanageren.

### Indhold i frontenden

Hvis et plugin vil levere indhold til frontenden skal pluginet implementere IProvidesContent. Derudover skal pluginet have registreret et namespace. Namespacet skal være unikt, og må gerne ændres, det anbefales at man opsætter en configurations guide i kontrolpanelet til registrering af namespace. Sådan at slutbrugeren kan bestemme hvilket namespace et plugin skal benytte. Ved implementering af IProvidesContent anbefales det at man også implementere ILinkable. ILinkable definere metoden GetLinks, som returnerer et træ med links til det indhold et plugin udbyder, kig nærmere i dokumentationen for ILinkable.

### Editor i backenden

Hvis et plugin vil udbyde en editor i backenden skal den implementere IProvidesEditor. Dette interface specificere 3 metoder. En der returnerer overskriften for editoren, en der returnerer et ikon og en der skriver editoren ud til en implementering af IEmbedmentOutputHandler. Det anbefales at editoren er implementeret som en Ajax applikation, og bliver bygget oven på Dojo framework, for at give brugeren en ensartet oplevelse. En editor skal ikke implementeres med mindre det er strengt nødvendigt. Det anbefales at man benytter IConfigure til implementering af configurations guides. Dette skyldes at antallet af ikoner i toppen af backenden er begrænset, kun ting der bliver benyttet rimeligt ofte bør implementere IProvidesEditor og dermed placere et ikon der.

## ***Konfiguration af plugin***

Et plugin der implementere IConfigure vil få et ikon placeret i kontrolpanelet. Desuden vil pluginet udbyde en konfigurations side, som er tilgængelig fra kontrolpanelet. Det anbefales at denne konfigurations side, bliver implementeret som Dojo guide. På samme måde som alle andre plugins i systemet. Dette skaber en enkel og gennemgående oplevelse for brugeren. De dojo guides der skal benyttes bør starte med en oversigt, hvor brugeren kan se hvilke opgaver guiden vil hjælpe med. Dernæst kan guiden gå i gang med de forskellige opgaver. Hvis pluginet skal præsentere nogle komplekse data, kan det være en fordel at benytte en guide side til at forklare teorien.

## ***Installations scripts***

Hvis et plugin benytter database tabeller, skal disse oprettes ved installation og fjernes ved uninstallation. Til det formål findes der IInstall og IRemove som hver især implementere metoder der bliver afviklet ved (un)installation. Husk at database tabeller skal navngives som defineret i Appendiks A: CMS3 Navngivnings konvention.