

Team Omega – Infoskærm Dokumentation

Systemets opbygning

OmegaScreen er et stærkt modul baseret system. Vi gør meget ud af og isolere logik ind i modulerne hvilket giver et bedre overblik.

Fra roden er der følgende indgange:

Indgang

- Admin.php
Som giver adgang til at interagere med vores system.
- Infoscreen.php
Infoscreen.php som viser infoskærm
- Slidecontent.php
Som viser de individuelle slides.

Moduler

Som indeholder alle de forskellige moduler.

Systemet er opbygget således at alle moduler bliver hentet automatisk ind, og det burde ikke være nødvendigt at ændre i resten af systemet, for at oprette et nyt modul.

Et moduls funktion bliver defineret efter, hvilket metoder og properties som det indeholder.

Hvert modul har sin egen mappe, som indeholder alt hvad der har med modulet at gøre, vigtigst af alt en .php fil som skal hedde det samme som mappen. Inde i den fil skal der igen være en klasse af samme navn, denne klasse vil blive hentet ind som det første, når et modul bliver loaded.

Et modul kan indeholde følgende properties.

- \$name
Name definere navnet på modulet, og er den der vil blive vist i menuen og i dropdown menuen.
- \$menu
Menu hvis den er sat til true, vil modulet blive linket til ude fra admin menuen, hvis den er sat til et tal, vil det være dens placering i menuen.
- \$slideevents
Hvis sat til true, så vil javascriptet i slideviewer triggere slideReady() og slideunload() på det pågældende slide.

Et modul kan indeholde følgende metoder

Content metoder

Content metoder er de metoder som genere det html modulet skal vise. Alt hvad der bliver generet skal printes frem for retuneres.

- `getpage()`
getpage bliver kaldt, hvis modulet bliver loaded ind i admin panelet. Som f.eks fra admin menuen.
- `getElement()`
getElement bliver kaldt når modulet bliver brugt, som et element på forsiden, som f.eks det modul der viser søgninger.
- `getslidecontent($id)`
getslidecontent bliver kaldt, når modulet bliver hentet som et slide,
- `getslideurl($id)`
getslideurl skal sættes istedet for getslidecontent hvis der skal loades en ekstern side. F.eks url modulet eller ugenmenu modulet.

Argument metoder

Disse metoder bliver kaldt fra admin panelet, når brugeren prøver at oprette eller redigere et slide.

- `getcreateargs()`
Skal returnere de html input felter, som er nødvendige for modulet ud over navn og tid.
- `geteditargs($id)`
Udføre samme funktionalitet som getcreateargs() men bør indeholde værdierne for det aktuelle slide.

Redigerings metoder

Disse metoder bliver kaldt når et slide bliver oprettet, slettet eller redigeret.

- `createslide($id)`
bliver kaldt når et slide bliver oprettet. Skal bruges til hvis det er nødvendigt at oprette ekstra data for at et slide kan fungere.
- `deleteslide($id)`
bliver kaldt når et slide bliver slettet. Skal ryde op efter createslide().
- `updateslide($id)`
bliver kaldt når et slide skal redigeres.
- `ajax()`
ajax er standard metoden der bliver kaldt, hvis du bruger systemets indbygget ajax funktionalitet.

Editable

Hvert slide, har i databasen, en værdi som hedder editable, hvis den er sat til true, så vil modulets `getpage()` ind loaded istedet for den normale redigeringsboks. Eksempel `slidefromscratch` modulet.

Javascript

Alle scripts der afvikler i vores system, har adgang til metoden `Ajax()`

Ajax tager 3 parametre, den første af hvilke er.

1. Modul navn
Navnet på det modul du gerne vil kalde
2. Parametre
Er et objekt, indeholdende de parametre du gerne ville sende med i forespørgelsen.
3. Callback
Den metode der bliver kaldt, når forespørgelsen er færdig hvis første parametre er den retunerede tekst.

slideControl

Dette objekt som eksistere i window, indeholder funktionalitet, til at styre slidevieweren.

Slideevents

Hvis `$slideevent` er sat til true i modulet, vil følgende metoder i et slide, blive kaldt.

1.slideReady()

den vil blive kaldt når et slide befinder sig i midten af skærmen

2.slideUnload()

Den bliver kaldt lige inden et slide forlader central.

- `Slidecontrol.isPause()`
- `Slidecontrol.play()`
- `Slidecontrol.pause()`
- `Slidecontrol.next()`
- `Slidecontrol.setTimer(time)`
- `Slidecontrol.previous()`

Essentials

Som indeholder de klasser som skal være tilgængelig alle steder fra i systemet. Derfor har vi også valgt at gøre alle metoderne i klasser statiske.

Autoinclude.php

Som indeholder autoload funktionen, som sørger for at alle klasser i essential bliver loaded ind efter behov.

Acc.php

Indeholder nogle grundliggende funktioner som vi ikke kunne ligge ander steder.

- `acc::rmdirr($directory)`
Den rekursive sletter mappen og alt dens indhold
- `acc::datarequest($url)`
Den henter data fra admin.ordbogen.com
- `acc::endpoint($method)`
Den returner den retunerede værdi fra den pågældende metode fra endpointet.

DB.php

Den sørge for automatisk at forbinde til databasen og lukke for forbindelsen, når den ikke skal bruges mere.

- `db::query($query)`
Den afvikler query på databasen, og retunere et result objekt, medmindre det er en SELECT hvor i såfald vil returnere et array indeholdende de forspurgte rækker.
- `db::lastid()`
retunere det sidst indsatte id fra databasen.

Htmldoc.php

Klassen er designet til at give grundliggende html funktionalitet.

Alt hvad der skal ende med at være et html dokument ned arver fra denne klasse.

- `Htmldoc::addheader($header)`
Addatere \$header til dokumentets header.
- `Htmldoc::export()`
Den printer hele html dokumentet ud

Alle steder vi har brugt den her i vores program, har vi la den klasse som ned arver fra den, heddet main. Således at du alle steder fra, kan kalde `main::addheader` for at addatere en header.

Modloader.php

Modloaders funktion er og hente de forskellige moduler ind. Og sørge for at der på et værd givent tidspunkt, kun eksistere en instance af modulet. Derfor opfordres der også til at der ikke på noget tidspunkt bliver oprettet en instace af et modul, uden og bruge modloader.

- `Mod::get($name)`
Returnere en instance af det angivet modul
- `Mod::load($name)`
Opretter en instance af modulet uden og retunere den.
- `Mod::exists($name)`

- Retunere en boolean om det angived modul eksistere eller ej.
- `Mod::getallwithmethod($method)`
Retunere et array, indeholdende alle moduler, med den pågældende metode, sorteret efter modul navn.
- `Mod::getallwithproperty($property)`
Retunere et array, indeholdende alle moduler, med den pågældende property, sorteret efter modul navn.

Server Setup

Vores system er designet til at køre på en linux server med lamp.

I `php.ini` skal

`upload_max_filesize`

`post_max_size`

Sættes til at være store nok til at kunne oploade en video.

Rettighederne skal sættes til at serveren har adgang til at oprette filer, i de moduler der kræver upload.

Serveren skal have `cifs-utils` installeret og der skal sættes en cron-task op til at trigger `modules/newestvideo/newvideocatcher.sh` det er vigtigt at det bliver afviklet som root da den skal kunne mounte.

Login informationer

I de systemer vi har sat op nu er brugernavnet Admin og kodeordet `omeg4dr4gon` for OmegaScreen mens brugernavnet på serveren er omega med kodeordet dragon