

MiniMVC

Index

- Inställningar - Vad inställningarna gör
- URLer - Hur funktioner körs
- Routing - Allt om snyggare länkar
- Filer & Struktur
- Exempel

Inställningar

Alla inställningar görs i MiniMVC/settings.php.

Databaser

Dessa inställningar är till för att ansluta till din databas och behövs. MiniMVC använder MySQL.

```
$this->db['host']      = 'localhost';
$this->db['user']       = '';
$this->db['pass']       = '';
$this->db['db']         = '';
```

Salt1 och salt2 används inte av MiniMVC, men gör det enkelt för dig att använda samma salt över allt, du kan lägga till salt3, saltFörLösenord, saltFörKryptering osv om du vill.

```
$this->db['salt1'] = "";
$this->db['salt2'] = "";
```

Prefix används inte av MiniMVC, till för dig så att du enkelt kan komma åt ev prefix för dina tabeller.

```
$this->db['tbl']['prefix'] = "";
```

Tbl[x] är till för att enkelt komma åt dina tabell namn. Används inte av MiniMVC. Gör det enkelt att kalla på tabeller text `SELECT * FROM $this->ini->db->['tbl']['news'];` Du kan även skriva `$tbl = $this->ini->db->['tbl'];` Då kan du skriva `SELECT * FROM $tbl['news'];`.

```
$this->db['tbl']['news'] = $prefix ."newsTableForMyCMS";
```

Default controller, 404

Default controller och default 404 sida sätts med dessa variabler.

```
$this->default['controller'] = "exampleController";
```

```
$this->default['404'] = "404"; //404 sidan
```

URLer

I MiniMVC funkar URLer så här

MiniMVC/funktion

Funktion körs från default controller utan argument.

MiniMVC/controller/funktion

Funktion körs från controller

MiniMVC/controller/funktion/argument

Funktion körs från controller med argument som en sträng.

MiniMVC/controller/funktion/argument/argumnt2/argmnt3/osv

Funktion körs från controller med argumenten som en array.

```
array(0 => "argument",  
      1 => "argumnt2",  
      2 => "argmnt3",  
      3 => "osv");
```

Routing

Med routing kan du göra snyggare URL:er / länkar.

Istället för att skriva `/newsController/category/blogg` så kan man med routing skriva `/blogg`.

```
$route["blogg"] = "/newsController/category/blogg";
```

Om du vill göra om `/newsController/newsFromId/*id*` till `/news/*id*` så går det också bra.

```
$route["news/([0-9]+)"] = "/newsController/newsFromId/$1" eller  
$route["news/(:num:+)"] = "/newsController/newsFromId/$1".
```

MiniMVC använder regexp för att matcha `$route[key]`. Alltså kan du använda vanlig regexp, tänk dock på att

- MiniMVC gör om `/` till `\` så du slipper det. `:num:` och `:alfa:` blir `[0-9]` respektive `[a-z]`. Du kan lägga till fler kort kommandon i `MiniMVC/core.php`, `route()`.
- MiniMVC lägger till `/^$route[key]$/` runt `key`, så du kan inte använda flaggor, tex `i`.
- Du kan använda `(?P<search>)` som sedan skickas till `$search`.
- Routing ställs in i `MiniMVC/settings.php`

Variabler & Konstanter

I MiniMVC finns 4 konstanter med info om sök vägar. Allt som returnerar sökvägar har inget avslutande "/"(slash).

- IS AJAX - `true` om scriptet körs från ett AJAX request.
`false` annars.
- MINIMVCFOLDER - "Webb" URLn till MiniMVC mappen. Bra för HTML länkar.
- VIEWFOLDER - "Webb" URLn till view mappen. Bra för att länka css och javascript kod samt bilder.
- MINIMVCFULLPATH - Script sökväg till MiniMVC mappen. Bra när man använder `include()`;

Filer & Struktur

`index.php` - kör `MiniMVC/core.php`

`.htaccess` - Snygga länkar

model/

-- | alla dina modeller, dessa innehåller funktioner som jobbar med databasen.

Du laddar dessa filer med `$this->load->model('exampleModel');`

Obs, utan `.php` ^

view/

-- | alla dina views, HTML, CSS, Javascript - kod, bilder, etc.

Views laddas med `$this->load->view('index');`

Obs, döp views till `.php`, inte `.html`. `.php` ^ läggs till

controller/

-- | dina controller filer, dessa innehåller alla funktioner och sköter det mesta

MiniMVC/

-- | `autoload.php` - Laddar in filer när man skriver `new x();`
Du slipper skriva `include()`;

-- | `core.php` - Sköter routing och vad som ska köras

-- | `functions.php` - Funktioner, kallas på med `$this->func->*funktionensNamn*`
Valfri fil. Används inte av MiniMVC.

-- | `load.php` - Laddar models, views och controllers

-- | `mainController.php` - Laddar `load.php`, `functions.php` (om den finns) och `settings.php` till variabler.
Detta skapar `$this->load->model`
syntaxen, istället för `$load->model`.

-- | `mainModel.php` - Ansluter till databasen och laddar `settings.php` till `$this->ini;`

-- | `settings.php` - Inställningar som DB uppgifter & routing

Exempel

Många lär sig bäst av exempel(kod), så här kommer några exempel.

Routing

```
// hello/hej, hello/u, hello/laaazooooooooor kommer blir
// exampleController/hello/hej, exampleController/hello/u,
// exampleController/hello/laaazooooooooor
$route["hello/([a-z]+)"] = "exampleController/hello/$1/";

// search/hundar/30/A blir searchEngine/advanceSearch/hunder/A/30/
// search/katt/1/Z blir searchEngine/advanceSearch/katt/Z/1/
// (?<*namn*>) skickas till $*namn*.
// () skickas till $1, nästa () skickas till $2, osv
$route["search/(?<search>[a-z]+)/(:num:+)/([A-Z]?)" ] =
"searchEngine/advanceSearch/$search/$3/$2/";

// pro/ blir exampleController/hello/pro
// pro blir exampleController/hello/pro
$route["pro"] = "exampleController/hello/pro";
```