

always one step advance

wieschooCMS

Module
für Entwickler

In den folgenden Seiten erfahren Sie, wie Erweiterungen für das **wieschooCMS** erstellt werden können.

Erweiterungen für das wieschooCMS unterscheiden sich in:

Manager	Module
Verwalten Datensätze im Redaktionsbereich	Stehen im Frontend zur Interaktion mit Besuchern/Redakteuren zur Verfügung.

Module

Das erste Modul, soll einfach „Hello World“ ausgeben. Jedes Modul benötigt einen eigenen Ordner. Erstellen Sie nun einen Ordner mit dem Namen „helloworld“. Achten Sie auf Groß- und Kleinschreibung, da dies unter linuxbasierenden Servern unterschieden wird.

```
Dateisystem
module
.. helloworld
```

Der Ordner „helloworld“ muss mindestens 2 Dateien enthalten.

- helloworld.modul.php
- info.xml

```
Dateisystem
module
.. helloworld
.. .. helloworld.modul.php
.. .. info.xml
```

Achten Sie auch hierbei auf die Groß- und Kleinschreibung.

Das Grundgerüst für neue Module sieht so aus:

```
<?php

// helloworld.modul.php

include_once $globVar['Path'] . 'interface/modul.interface.php';

class helloworld implements modul {
    private $handle_id = null;

    public function handle( $id )
    {
        $this->handle_id = $id;
    }

    public function initialise( $parameter )
    {
    }

    public function run( $action )
    {
        return "";
    }

    public function destroy()
    {
    }
}

?>
```

der Aufbau orientiert sich am Interface „modul.interface.php“.

```
<?php

// module.interface.php

interface modul {
    public function handle( $id );
    public function initialise($parameter);
    public function run($action);
    public function destroy();
}

?>
```

Nun zurück zu unserem Modul:

```
public function handle( $id )
{
    $this->handle_id = $id;
}
```

Jedes Modul erhält eine eigene ID, um zur Laufzeit identifiziert zu werden. Dies ist wichtig, da ein Modul mehrfach auf einer Seite auftreten kann.

```
public function initialise( $parameter )
{
}
```

Das ist die Startfunktion eines Moduls. Mit dieser Funktion (genau genommen Methode) werden Startparameter übergeben. Bei dem StaticContent-Modul ist das zum Beispiel die ID des Datensatzes der den auszugebenen Text speichert.

Wir überspringen zunächst „run(\$aktion)“

```
public function destroy()
{
}
```

Wenn das Modul aufgelöst wird, ruft wieschooCMS diese Methode auf.

Nun geht es zur Hauptmethode, dem Kernstück eines Moduls:

```
public function run( $action )
{
    return "";
}
```

Jedes Modul hat nicht Initialisierungsparameter sondern bekommt auf die auszuführende Aktion zugeteilt.

Beim Login-Modul sind dies:

- Loginform aufrufen
- Logindaten überprüfen
- ausloggen
- Passwort ändern

Diese Aktionen können über einen switch unterschieden werden. pro Modul darf nur eine Aktion ausgeführt werden. Sollen mehrere Aktionen ausgeführt werden, müssen mehrere Modul in die Seite eingebunden werden.

Beispiel für Unterscheidung der Aktionen

```
switch ($action) {
    default:
    case "loginform":
        return $this->loginform();
        break;
    case "validate":
        return $this->validate();
        break;
    case "logout":
        return $this->logout();
        break;
    case "changepwd":
        return $this->changepwd();
        break;
}
```

Der Rückgabewert der „run“-Methode wird ausgegeben.

In unserem Fall sollte dies „Hello World!“ sein:

```
<?php
. . .
public function run( $action )
{
    return "Hello World!";
}
. . .
?>
```

Komplett sie der Quellcode so aus:

```
<?php
// helloworld.modul.php

include_once $globVar['Path'] . 'interface/modul.interface.php';

class helloworld implements modul {
    private $handle_id = null;

    public function handle( $id )
    {
        $this->handle_id = $id;
    }

    public function initialise( $parameter )
    {
    }

    . . .
}
```

```

. . .

public function run( $action )
{
    return "Hello World!";
}

public function destroy()
{
}
}

?>

```

Wir können hier getrost den Parameter `$action` ignorieren, da wir nur eine Aktionen anbieten.

Informationen für das wieschooCMS

Damit das System etwas mit dem neuen Modul anfangen kann, müssen wir ein paar Informationen in der „info.xml“ speichern.

```

<?xml version="1.0" encoding="utf-8"?>
<wieschoo>
  <version>1.0.0</version>
  <name>Hello World</name>
  <description>Beispielmodul fuer Funktionsweise des Systems.</description>
  <authors>
    <author>
      <name>Patrick Wieschollek</name>
      <email>info@wieschoo.de</email>
    </author>
  </authors>
  <actions>
    <action>default</action>
  </actions>
  <parameters>
  </parameters>
</wieschoo>

```

Vieles erklärt sich von selbst.

Bei `<actions>` werden die Aktionen bekannt gegeben, die das Modul ausführen kann. Da bei uns nur eine Aktion möglich ist, geben wird „default“ an.

Parameter wie Datensatz IDs benötigen wir auch keine.

Veröffentlichen

Jetzt brauch der Inhalt des Ordners (Wichtig: Nicht der Ordner mit dem Inhalt, sondern nur der Inhalt!) zu einem Zip-Archiv zusammen gefasst werden. Senden Sie mir ihn zu (dev@wieschoo.com). unter dem Addon-Server wird dieser dann veröffentlicht.