Framework Foton

Версия 0.12.278

Быстрый старт

Требование к системе

- Сервер Apache 2.4+
- Тип сервера баз данных MySql/Postgresql/LiteSQL
- Версия интерпретатора php 5.6.* - 8.*.*
- ОС: Widows/Linux
- Установленные модули (zip,unzip)
- Библиотеки:
  1. Curl
  2. XMLWriter
  3. SoapClient (не обязательно)
  4. PDO
  5. Smtp
- Разрешенные функции
  1. exec
  2. eval

Установка системы

Для установки системы скачайте вашу версию с сайта foton.name и распакуйте архив в директорию вашего сайта.
Перейдите по адресу в строке браузера http{s}://вашсайт/install.php
Создайте базу данных и укажите ее параметры на данной странице, а также лицензионный ключ, после этого войдите в открытую форму указав везде demo, перейдите в модуль Обновление модулей, и обновите модули до последней версии, также рекомендуем обновить ядро системы в верхней панели, а также в разделе пользователи изменить ваши данные, более безопасно создать нового администратора, зайти под ним и удалить demo администратора, иначе после изменения логина вас выбросит из системы с ошибкой, и если вы не верно ввели пароль или забыли его вам придется вручную менять ваш пароль в базе данных. Также рекомендуем создать ваш личный ключ нажав на соответствующий значок в верхней панели РСИ, данный ключ требует синхронизации по времени вашего устройства с вашим сервером.
Установка для LiteSQL - Для установки системы скачайте вашу версию с сайта inc-lab.ru и распакуйте архив в директорию вашего сайта.
Удалите лишние файлы pbase.sql,base.sql,install.php откройте файл core/config.php и укажите: $GLOBALS["host"]='localhost';
$GLOBALS["sql"]='lite';
$GLOBALS["dbname"]='foton';
$GLOBALS["license"]='вашалицензия';
Далее перейдите по адресу http{s}://вашсайт/admin/, далее шаги такие же как и для других SQL версий.

Создание одностраничного приложения

В качестве базы данных мы будем использовать mysql.
Мы будем использовать только один метод из ядра:
getlist - для вывода списка элементов таблицы
Директории, которые нам понадобятся для работы:
/app/model/site/
/app/controller/site/

/app/view/site/

/app/view/json/site/

Добавим настройку 'route'=>true в файле /core/config.php

Создаем модель

Создаем файл model_news.php в директории /app/model/site/

Заполняем его следующим содержимым:

```
<?php
 class Model_news extends Model{
public function nameinclude(){
    return 'Контент'; //название раздела в админке
}
public function names(){
    $arr = array('news'=>'Новости','status'=>'Статусы'); //название таблицы и
подраздела таблицы в админ. панели
    return $arr;
}
}
?>
```

Добавляем права на разделы для ролей пользователей:

```
public function news_chmod(){
    return '1,2'; //id ролей, которым доступна таблица
}
public function status_chmod(){
    return '1,2'; //id ролей, которым доступна таблица
}
```

Добавляем метод для создания самих таблиц:

```
public function interfaces(){
```

```
   $arr = array(
"news" => array(
"field" => array('id','name','text','statusf','photo'), //поля таблицы в базе
"name" => array('id','Название','Содержание новости','Статус','Фото'), //описание полей, для описание полей в админке
"format" => array('int','text','text','text','text'), //формат полей в базе данных*
"format_select" => array("ids","input","textarea",'select:status','img'), //формат типов данных**(***)
"key" => "id" //ключ таблицы
));
return $arr;
}
```

* Для mysql:
'text'=>'text(64300) NOT NULL',
'int'=>'int(70) NOT NULL',
'date'=>'DATE NOT NULL',
'bit'=>'BIT NOT NULL',
'poly'=>'POLYGON NOT NULL',
'real'=>'REAL NOT NULL',
'time'=>'TIME NOT NULL',
'medium'=>'mediumtext NOT NULL'

** все типы данных вы можете посмотреть по ссылке: вашдомен//list/html/html в поле «Код»
*** - select2:status – select2 – тип данных, status аргумент, если аргументов несколько перечисляются через запятую, тип данных select использует для вывода поле text, так что наша таблица status должна иметь это поле, давайте создадим ее

Добавляем в массив $arr еще один массив:

```
"status" => array(
"field" => array('id','name','text'), //поля таблицы в базе
"name" => array('id','Название','Код'), //описание полей, для описание полей в админке
```

```
"format" => array('int','text','text'), //формат полей в базе данных
"format_select" => array("ids","input",'input'), //формат типов данных
"key" => "id" //ключ таблицы
)
```

Валидация данных при вводе и обновлении:

```
//метод исправляет данные при добавлении и обновлении
public function news_validate(){
$arr = [
'public_ins'=>['name'=>'tags','text'=>'tags','statusf'=>'tags'], //при добавлении с публичной части приложения
'ins'=>['name'=>'tags','text'=>'tags','statusf'=>'tags'], //при добавлении с административной части приложения****
'up'=>['name'=>'tags'] //при обновлении с административной части приложения
];
return $arr;
}
//метод проверяет данные при добавлении и обновлении
public function news_is(){
$arr = [
'ins'=>['name'=>'0','text'=>'0','statusf'=>'0'], //при добавлении с административной части приложения*****
'up'=>['name'=>'0','text'=>'0','statusf'=>'0'] //при обновлении с административной части приложения
];
return $arr;
}
```

****значениями массива могут быть:
html — эквивалент htmlspecialchars
pass — создает md5 хеш
text — оставляет только печатные символы
medium — оставляет только русские и латинские символы и пробел
abcd — оставляет только русские и латинские символы, цифры и пробел
abc - оставляет только латинские символы

abv - оставляет только русские символы
same — оставляет данные без изменения
int – оставляет только цифры
phone — оставляет цифры ± и скобки
tags — удаляет все теги
plus — аналог метода abs
mini – аналог метода floor
maxi - аналог метода ceil
nul – всегда выдает null
tabs – удаляет все пробельные символы

*****значениями массива могут быть:
M – mail
D – домен
F – число float
IP – IP адрес
R – регулярное выражение
U – URL адрес
I – число integer
0 – непустое значение
T – строка

Перейдем в административную часть сайта в меню должен появится пункт Контент с подразделами Новости и статусы, давайте заполним несколько статусов (опубликована-pub, на модерации-moderation, отклонена-no,новая-new)
и 2 новости с любым содержанием.

Создаем контроллер

Перейдем в каталог /app/controller/site/ и создадим файл controller_news.php со следующим содержимым:

```
<?php
class Controller_news extends Model_news{
```

```
public function mvc_page(){
        return $this->core->getlist(['news','where'=>['=statusf'=>'pub']]);
}
public function json_page(){
        return $this->core->getlist(['news','where'=>['=statusf'=>'pub'],'format'=>'J']);

}
public function xml_page(){
        return $this->core->getlist(['news','where'=>['=statusf'=>'pub'],'format'=>'X']);

}
}
```

* первый параметр аргумента метода news – название таблицы, where = массив для отбора элементов для вывода, может содержать следующие первые символы ключей - = → равно,пустое → равно, ! → не равно,%->LIKE,^ → начало строки,$ → конец строки, < → меньше,> → больше, далее в методах json и xml мы добавляем поле format – формат вывода

Создаем представление

Перед созданием представления давайте создадим несколько полей данных для вывода перейдя по ссылке вашдомен/interface_list/html/exfield с названием mail и tel, если они уже существуют, можете просто обновить их значения на свои.

Теперь перейдем в каталог /app/view/site/ и создадим файл news_view.tpl со следующим содержимым:

```
<header>
<p>@@@tel@@@</p>
<a href='mailto:@@@mail@@@'>@@@mail@@@</a>
</header>
<main>
<div class="news-list">
@for:{$data['mvc_page'] as $news}
<div class="news-item">
```

```
<p>@{news['name']}</p>
<img src="/app/view/@{GLOBALS['sitedir']}/@{news['photo']}">
<div class="content">@{news['text']}</div>
</div>
@:@
</div>
</main>
```

Также давайте создадим файл /app/view/json/site/news_view.tpl со следующим содержимым:
@{data['json_page']}
Также создадим файл /app/view/xml/site/news_view.tpl со следующим содержимым:
@arr{data['xml_page']}
Теперь перейдем в административную панель и обновим кеш шаблонов нажав на значок обновления шаблонов:

Либо через консоль зайдя в корневую директорию вашего сайта введите команду#: php foton up
Затем давайте перейдем по адресу вашсайт/news/, должен отобразится список опубликованных новостей, стили для страницы вы можете добавить создав файл: /app/view/site/css/news.css, js для страницы вы можете добавить создав файл по адресу: /app/view/site/js/news.js
Также перейдя на страницу вашсайт/news.json мы увидим отображение наших новостей в json формате.
Перейдя на страницу вашсайт/news.xml мы увидим отображение наших новостей в xml формате.

Полностью код для тестирования по файлам:

Файл /app/model/site/model_news.php:

```
<?php
```

```
 class Model_news extends Model{
public function nameinclude(){
    return 'Контент'; //название раздела в админке
}
public function names(){
    $arr = array('news'=>'Новости','status'=>'Статусы'); //название таблицы и
подраздела таблицы в админ. панели
    return $arr;
}
public function news_chmod(){
    return '1,2'; //id ролей, которым доступна таблица
}
public function status_chmod(){
    return '1,2'; //id ролей, которым доступна таблица
}
public function interfaces(){
   $arr = array(
"news" => array(
"field" => array('id','name','text','statusf','photo'), //поля таблицы в базе
"name" => array('id','Название','Содержание новости','Статус','Фото'), //описание
полей, для описание полей в админке
"format" => array('int','text','text','text','text'), //формат полей в базе данных*
"format_select" => array("ids","input","textarea",'select:status','img'), //формат типов
данных
"key" => "id" //ключ таблицы
),
"status" => array(
"field" => array('id','name','text'), //поля таблицы в базе
"name" => array('id','Название','Код'), //описание полей, для описание полей в
админке
"format" => array('int','text','text'), //формат полей в базе данных*
"format_select" => array("ids","input",'input'), //формат типов данных**(***)
"key" => "id" //ключ таблицы
));
return $arr;
```

```
}
//метод исправляет данные при добавлении и обновлении
public function news_validate(){
$arr = [
'public_ins'=>['name'=>'tags','text'=>'tags','statusf'=>'tags'], //при добавлении с публичной части приложения
'ins'=>['name'=>'tags','text'=>'tags','statusf'=>'tags'], //при добавлении с административной части приложения
'up'=>['name'=>'tags'] //при обновлении с административной части приложения
];
return $arr;
}
//метод проверяет данные при добавлении и обновлении
public function news_is(){
$arr = [
'ins'=>['name'=>'0','text'=>'0','statusf'=>'0'], //при добавлении с административной части приложения
'up'=>['name'=>'0','text'=>'0','statusf'=>'0'] //при обновлении с административной части приложения
];
return $arr;
}
}
?>
```

Файл /app/controller/site/controller_news.php:

```
<?php
class Controller_news extends Model_news{
public function mvc_page(){
        return $this->core->getlist(['news','where'=>['=statusf'=>'pub']]);
}
public function json_page(){
        return $this->core->getlist(['news','where'=>['=statusf'=>'pub'],'format'=>'J']);
}
```

```
public function xml_page(){
        return $this->core->getlist(['news','where'=>['=statusf'=>'pub'],'format'=>'X']);
}
}
```

Файл /app/view/site/news_view.tpl:

```
<header>
<p>@@@tel@@@</p>
<a href='mailto:@@@mail@@@'>@@@mail@@@</a>
</header>
<main>
<div class='news-list'>
@for:{$data['mvc_page'] as $news}
<div class='news-item'>
<p>@{news['name']}</p>
<img src='/app/view/@{GLOBALS['sitedir']}/@{news['photo']}'>
<div class='content'>@{news['text']}</div>
</div>
@:@
</div>
</main>
```

файл /app/view/json/site/news_view.tpl со следующим содержимым:
@{data['json_page']}

файл /app/view/xml/site/news_view.tpl со следующим содержимым:
@arr{data['xml_page']}

Просмотреть логи ошибок и предупреждений вы можете в консоли вашего браузера.
Для корректной работы отладки в режиме json необходимо указать дополнительный get параметр, например так: /news.json?format=api_json