WEB разработка

CI Models

Съдържание

CI Query Builder

CI Query Builder

- Тъй като реализира MVC pattern, CI позволява да работим с различни бази данни. Така без да има нужда от промяна на кода можем да мигрираме от MySQL БД към PostgreSQL БД например.

- Query Builder класът също така прави автоматичен escaping на зявките.

Конфигуриране и връзка към БД в CI

Конфигуриране и връзка с БД ...

Конфигуриране – config/database.php

```
$db['default']['hostname'] = "localhost";
$db['default']['username'] = "root";
$db['default']['password'] = "";
$db['default']['database'] = "database_name";
```

Конфигуриране и връзка с БД … - 2

Връзка към БД – config/autoload.php

```
$autoload['libraries'] = array('database');
```

CI CRUD / Read

CI CRUD / Read

MODEL - документация

model->controller->view

1. *Дефинираме метод със заявка към БД ->*
2. *Извикваме метода в контролера->*
3. *Отпечатваме получената информация от БД*

За пример използваме БД - oophp.sql

CI CRUD / Read - 2

Задачи:
Да се извлече информация за имената/таблица names/ от БД и да се покаже по следния начин -
Таблица 1 - Номериран списък с имената.
Дом. На всеки ред в таблица 1 да има бутон Show, който показва в нова страница информацията за конкретното име - име, значение, пол и бутон за връщане към общата таблица.

Alice - noble, light - girl

CI CRUD / Read - 3

MODELS

class Names_model extends CI_Model{

……..

}

Заявка за извличане на всички имена.

models/names_model.php

```
public function get_all_names()
    {
        $q = $this->db->get('names');
        return $q->result_array();
    }
```

CI CRUD / Read - 4

CONTROLLER

controllers/names.php

```
public function show_all_names()
{
    $this->load->model('names_model');
    $data['all_names'] = $this->names_model->get_all_names();
    $this->load->view('names/all_names', $data);
}
```

CI CRUD / Read - 5

VIEW

Отпечатване на таблица с имената

views/names/all_names.php

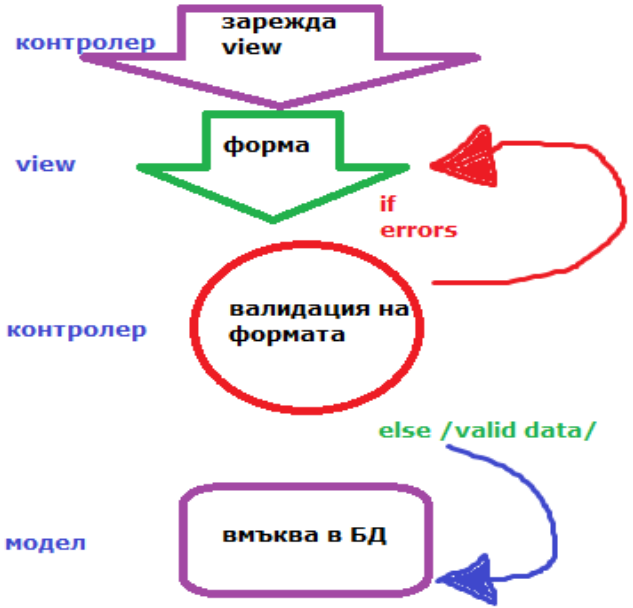
```
<?php
    $i = 1;
    echo <table border="1">
    foreach($all_names as $key => $value)
        {
            echo ‘<tr>’;
            echo " <td> $i++ </td>”;
            echo " <td> $value[‘name’] </td>”;
            echo ‘</tr>’;
        }
                echo </table>
```

CI CRUD / Create

CI CRUD /Create

CI Tutorial, Create News Item

Form Validation Library



Form Helper

form_open()

Creates an opening form tag with a base URL **built from your config preferences**. It will optionally let you add form attributes and hidden input fields, and will always add the attribute accept-charset *based on the charset value in your config file.* The main benefit of using this tag rather than hard coding your own HTML is that it permits your site to *be more portable* in the event your URLs ever change. …..

echo form_open('email/send'); ……

CI CRUD /Create - 2

За показване на валидационни грешки

$this->load->library('form_validation')

controllers/names.php

```
public function show_add_name()
{
    $this->load->library('form_validation');
    $this->load->view('names/add_name_view');
}
```

CI CRUD /Create - 3

echo validation_errors() – показва валидационни грешки

views/add_name_view.php

```
<?php
echo validation_errors();
echo form_open('name/insert_name');
echo form_input('name');
……
echo form_submit('submit', 'Save New Name');
echo form_close();
```

CI CRUD /Create - 4

CONTROLLER validation

controllers/names.php

```
public function insert_name()
{
	$this->load->model('names_model');
	$this->load->helper('form');
	$this->load->library('form_validation');
	$this->form_validation->set_rules('name', 'Име', 'required');

		if($this->form_validation->run() === FALSE)
		{
			$this->show_add_name();
		}
		else
		{
			$this->names_model->add_name();
			echo "Successfully inserted a new name in DB!";
		}
  }
```

CI CRUD /Create - 5

MODEL - добавяне на запис в БД

models/names_model.php

```
public function add_name()
{
      $name = array('name' => $this->input->post(‘name’),
                  '????' => $this->input->post(‘???’),
                  …
                  );

                  return $this->db->insert('cars', $car);
}
```

CI CRUD /Create - 6

CONTROLLER

names/

Всеки метод, който работи с модел, трябва да има достъп до съответния модел - в случая -

```
$this->load->model('names_model');
```

Трябва да присъства в дефиницията на всеки метод.

CI CRUD /Create - 7

CONTROLLER __constructor - документация

names/

За да избегнем множеството повторения на кода -

```
public function __construct()
        {
                parent::__construct();
                 $this->load->model('names_model');
        }
```

Ресурси

Codeigniter Features
Query Builder
Database Configuration
Connecting to your Database
News section
Create news items