

Projektowanie aplikacji w modelu MVC opartej o framework CodeIgniter

Konfiguracja i przygotowanie przykładowej aplikacji opartej o model Model-View-Controller

Konfiguracja frameworka

- Zainstaluj na serwerze framework CI w domyślnej konfiguracji,
- Dokonaj modyfikacji pliku w lokalizacji application/config/autoload.php

zmień pozycję: \$autoload['libraries'] = array('');

na: \$autoload['libraries'] = array('database');

- Dokonaj modyfikacji pliku w lokalizacji application/config/routes.php

zmień pozycję: \$route['default_controller'] = welcome;

na: \$route['default_controller'] = 'start';

- Dokonaj modyfikacji pliku w lokalizacji application/config/database.php

Zmień pozycje w pliku na następujące:

```
'username' => 'root',  
'password' => '',  
'database' => 'test',  
'dbdriver' => 'mysqli',
```

Przygotowanie bazy danych do pracy z aplikacją

Utwórz bazę danych o nazwie test.

Utwórz w powyższej bazie tabelę plik o następującej strukturze i przykładowych danych:

```
-- MySQL Administrator dump 1.4
```

```

--
-----
-- Server version 5.5.5-10.1.19-MariaDB

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

--
-- Create schema test
--

CREATE DATABASE IF NOT EXISTS test;
USE test;

--
-- Definition of table `plik`
--

DROP TABLE IF EXISTS `plik`;
CREATE TABLE `plik` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nazwa_pliku` text NOT NULL,
  `sciezka` text NOT NULL,
  `rozmiar` int(10) unsigned NOT NULL,
  `typ` varchar(45) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;

--
-- Dumping data for table `plik`
--

/*!40000 ALTER TABLE `plik` DISABLE KEYS */;
INSERT INTO `plik` (`id`,`nazwa_pliku`,`sciezka`,`rozmiar`,`typ`) VALUES
(1,'01.txt','./',1214,'txt'),
(2,'test.mp3','/home/dane',34434342,'mp3'),
(3,'video_course','c:\dane_archiwalne',4294967295,'avi'),
/*!40000 ALTER TABLE `plik` ENABLE KEYS */;

```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

Projektowanie kontrolera

Utwórz plik Start.php w lokalizacji application/controllers/ o następującej treści:

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
class Start extends CI_Controller {  
    public function index()  
    {  
        echo "Witaj w kontrolerze<br>";  
    }  
}
```

Sprawdź działanie kontrolera, wpisując adres w oknie przeglądarki (zakładamy, że framework CI został zainstalowany w głównym katalogu serwera w podkatalogu ci):

<http://127.0.0.1/ci>

Dodaj do kontrolera następujące metody:

```
public function pobierz_dane()  
{  
    // załaduj model tabeli plik  
    $this->load->model('plik');  
  
    // wykonaj funkcje z klasy modelu  
    $result = $this->plik->pobierz_wszystkie_rekordy();  
  
    // wypisz wyniki  
    echo "Pobranie wszystkich danych z tabeli pliki <br><br>";  
    echo "Struktura wyniku: <br>";  
    print_r($result);  
    echo "<br><br>";  
    echo "Dane w postaci tablicy: <br>";  
    $dane = $result->result();  
    print_r($dane);  
    echo "<br><br>";  
  
    // zapamiętanie ID pierwszego rekordu do zastosowania
```

```

// w kolejnej funkcji
$id = -1;

if (count($dane)>0)
{
    $id = $dane[0]->id; // wskazanie pola tabeli
}

// Pobranie rekordów ograniczonych kryteriami (wybór rekordu na podstawie id)
$result2 = $this->plik->pobierz_dla_id($id);
$dane2 = $result2->result();

echo "Wynik działania drugiej funkcji <br><br>";
print_r($dane2);

$this->plik->dodaj_rekord('brak_pliku','/home','434343433','txt');
}

public function pokaz_tabele()
{
    $header = array('Liczba1','Liczba2','Iloczyn');
    $dane[] = array(2,3);
    $dane[] = array(5,7);
    $dane[] = array(6,9);
    $dane[] = array(3,7);

    $this->load->view('tabela_widok',array('header'=>$header,'dane'=>$dane));
}

```

Projektowanie modelu

Utwórz w katalogu application/models plik o nazwie "plik.php" o następującej treści:

```

<?php

class Plik extends CI_Model {

public function __construct()
{
    parent::__construct();
}

public function pobierz_wszystkie_rekordy()
{

return $this->db->get("plik");
}
}

```

```

}

public function pobierz_dla_id($id)
{
    $data = array(
        'id' => $id
    );

    $query = $this->db->get_where('plik',$data);
    return $query;
}

public function dodaj_rekord($nazwa_pliku,$sciezka,$rozmiar,$typ)
{
    $data = array(
        'nazwa_pliku'=>$nazwa_pliku,
        'sciezka'=>$sciezka,
        'rozmiar' => $rozmiar,
        'typ'=>$typ
    );

    $this->db->insert('plik', $data);
}
/*
    Dodaj implementacje funkcji aktualizuj, odpowiedz: skorzystaj z funkcji:
    $this->db->where(...);
    $this->db->update(...);
}
*/
?>

```

Projektowanie widoku

Utwórz w katalogu application/views plik tabela_widok.php o następującej treści:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>

<h2>Przykładowa tabela z danymi przekazana z kontrolera</h2>
<table style="width: 50%">
<thead>
<?php

echo "<tr style='background: lightgray;'>";
for ($a=0;$a<count($header);$a++)

```

```

{
    echo "<th>";
    echo $header[$a];
    echo "</th>";
}
echo '</tr>';
?>
</thead>
<tbody>
<?php
$ilosc = count($dane);
for ($b=0;$b<$ilosc;$b++):
?>
<tr style="background: lightblue; text-align: center;">
<?php $ilosc2 = count($dane[$b]);
$iloczyn = 1;
for ($c=0;$c<$ilosc2;$c++):
?>
<td>
<?php echo $dane[$b][$c];
$iloczyn*=$dane[$b][$c];
?>
</td>
<?php endfor; ?>
<td>
<?php echo $iloczyn; ?>
</td>
</tr>
<?php endfor; ?>
</tbody>
</table>
</body>
</html>

```

Zapoznaj się logiką funkcjonowania przedstawionej aplikacji wywołując następujące adresy w przeglądarce (zakładamy instalację frameworka w katalogu ci serwera):

- http://127.0.0.1/ci/index.php/start/pobierz_dane
- http://127.0.0.1/ci/index.php/start/pokaz_tabele

Zadania:

1. Utwórz nowy kontroler i widok, który pobiera wszystkie dane z tabeli plik i wyświetla odczytane informacje w postaci tabeli,
2. Dodaj funkcjonalność do powyższego kontrolera, która pozwala na przeprowadzenie operacji CRUD na wskazanej tabeli,
3. Zmodyfikuj interfejs użytkownika przez zastosowanie komponentów biblioteki Bootstrap