

Diario di lavoro

Luogo	Balerna
Data	13.05.2020

Lavori svolti

Nella giornata di oggi mi sono occupato di completare i seguenti capitoli della documentazione:

- Implementazione
 - o Gestione versioni
 - o Gestore di pacchetti
 - o Database
 - Account di accesso
 - Implementazione banca dati
 - Interrogazione database
 - o Applicativo web
 - Struttura
 - Model View Controller (MVC)
 - Representational State Transfer (REST)
 - Routing delle richieste
 - Configurazione

Come da pianifica mi sono occupato di configurare il framework php-rest. Ho dunque creato un progetto attraverso la creazione di un file "composer.json" nel quale ho specificato le dipendenze del progetto.

```
{
  "name": "filippo/ritardi-web",
  "description": "Gestione Web Ritardi Allievi SAMT",
  "authors": [
    {
      "name": "Filippo Finke",
      "email": "filippo.finke@samtrevano.ch"
    }
  ],
  "minimum-stability": "dev",
  "autoload": {
    "psr-4": {
      "FilippoFinke\\": "src/"
    }
  },
  "require": {
    "filippofinke/php-rest": "dev-master"
  }
}
```

Una volta creato il file composer ho creato la struttura di cartelle del progetto:

```

├── assets
├── src
│   ├── Controllers
│   ├── Libs
│   ├── Middlewares
│   ├── Models
│   └── Views
└── vendor

```

Successivamente ho creato il file .htaccess che verrà utilizzato per il routing delle richieste:

```

RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.+)$ index.php [QSA,L]

```

Ho creato inoltre un file di configurazione che verrà utilizzato per ricavare le credenziali di connessione al database di esempio chiamato "config.sample.php":

```

<?php
/**
 * Config.php
 * File nel quale risiede la configurazione del progetto.
 *
 * @author Filippo Finke
 */
/**
 * Configurazioni riguardanti la banca dati.
 */
// Indirizzo di connessione del database.
define("DB_HOST", "127.0.0.1");
// Il nome del database.
define("DB_NAME", "ritardi");
// L'username dell'utente da utilizzare per collegarsi al database.
define("DB_USERNAME", "USERNAME");
// La password da utilizzare per il collegamento al database.
define("DB_PASSWORD", "PASSWORD");

```

Una volta creato il file di configurazione ho creato due viste di prova che ho inserito nella cartella "Views". La prima vista chiamata "test.php" la quale contiene solo un messaggio di test e la seconda chiamata "error.php" la quale verrà caricata in caso di errori. Per caricare la vista di test ho dunque creato un controller anch'esso di prova, il codice è il seguente:

```
<?php
namespace FilippoFinke\Controllers;
use FilippoFinke\Request;
use FilippoFinke\Response;
class Test {
    public static function index(Request $req, Response $res) {
        return $res->render(__DIR__ . '/../Views/test.php');
    }
}
```

Il codice in questione carica semplicemente la vista “test.php”. In fine ho creato il file principale del progetto nel quale verrà eseguito il routing delle richieste, ovvero “index.php”. Essendo che utilizzerò il server web di PHP per testing ho aggiunto come prima cosa un controllo per i file statici presenti nel progetto:

```
if (php_sapi_name() == 'cli-server') {
    $path = $_SERVER["REQUEST_URI"];
    if (strpos($path, '/assets/') !== false) {
        return false;
    }
}
```

In questo modo i file statici verranno serviti dal web server senza passare dal router di php-rest. Successivamente ho aggiunto un controllo per la presenza del file di configurazione:

```
// Controllo del file di configurazione.
if (!file_exists("config.php")) {
    $response = new Response();
    $info = array(
        "title" => "File di configurazione mancante!",
        "message" => "Impossibile caricare il file di configurazione (config.php)!<br><br>Se stai installando l'applicativo puoi trovare un file di configurazione di esempio chiamato config.sample.php che puoi rinominare config.php"
    );
    $response->render(__DIR__ . "/src/Views/Error/error.php", $info);
    exit;
}
require __DIR__ . '/config.php';
```

In questo modo se il file di configurazione non è presente verrà mostrata la vista “error.php” alla quale verranno passate le informazioni presenti nell’array `$info`.

Dopo il controllo del file di configurazione ho creato la connessione al database con i dati ricavati da esso:

```
// Imposto indirizzo del server MySQL.
```

```

Database::setHost(DB_HOST);
// Imposto del database da utilizzare.
Database::setDatabase(DB_NAME);
// Imposto del nome utente per accedere al database.
Database::setUsername(DB_USERNAME);
// Imposto della password per accedere al database.
Database::setPassword(DB_PASSWORD);
// Controllo connessione alla banca dati.
try {
    Database::getConnection();
} catch (PDOException $e) {
    $response = new Response();
    $info = array(
        "title" => "Impossibile stabilire la connessione con il database!",
        "message" => "Errore: " . $e->getMessage()
    );
    $response->render(__DIR__ . "/src/Views/Error/error.php", $info);
    exit;
}

```

Ed in fine ho registrato dei percorsi di prova:

```

$router = new Router();
$router->setNotFound(function (Request $req, Response $res) {
    return $res->redirect("/");
});
$router->get("/", "FilippoFinke\Controllers\Test::index");
$router->start();

```

Ho inoltre aggiunto un campo "type" alla tabella "SETTING" all'interno dello schema ER che verrà utilizzato per verificare il tipo di dato dell'impostazione.

Problemi riscontrati e soluzioni adottate

Non ho riscontrato nessun problema nella giornata di oggi.

Punto della situazione rispetto alla pianificazione

Essendo che ho terminato la configurazione della base REST e lo sviluppo della banca dati mi trovo due giorni in anticipo rispetto alla pianifica iniziale.

Programma di massima per la prossima giornata di lavoro

Iniziare lo sviluppo della gestione degli accessi all'interno del sito web.