

# Architetture MVC e standard PSR-7

Corso Backend System Integrator  
Modulo **Programmazione PHP**

Docente: Dott. Enrico Zimuel

in collaborazione con:



REGIONE  
PIEMONTE

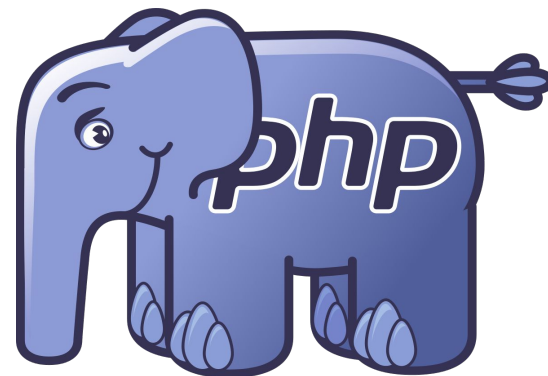
per una crescita intelligente,  
sostenibile ed inclusiva

[www.regione.piemonte.it/europa2020](http://www.regione.piemonte.it/europa2020)

INIZIATIVA CO-FINANZIATA CON FSE

# Programma

- Model-View Controller
- PSR-7
- Simple-MVC

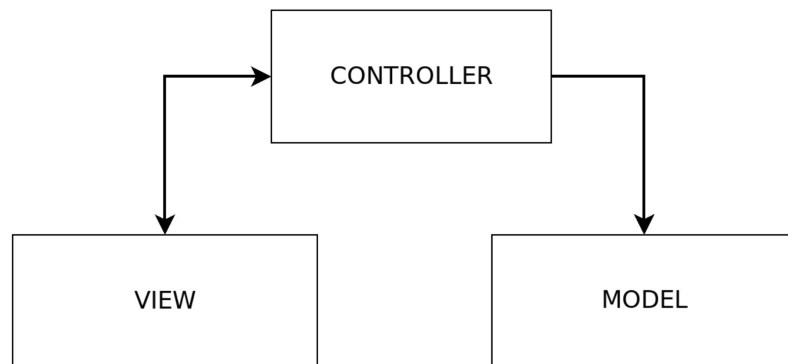


# Model-View Controller

- **Model-View-Controller** (MVC) è un design pattern di tipo architetturale che suddivide un'applicazione in tre insiemi:
  - **Model**, business logic (es. accesso al db)
  - **View**, l'output di un'applicazione (es. HTML)
  - **Controller**, logica di routing

# MVC

- In un'applicazione web la richiesta HTTP è gestita dal Controller
- Il Controller richiama il Model per determinare quali dati passare alla View
- La risposta (HTML) è il contenuto della View renderizzato con i dati (variabili) passati dal Controller



# Esempi di framework MVC

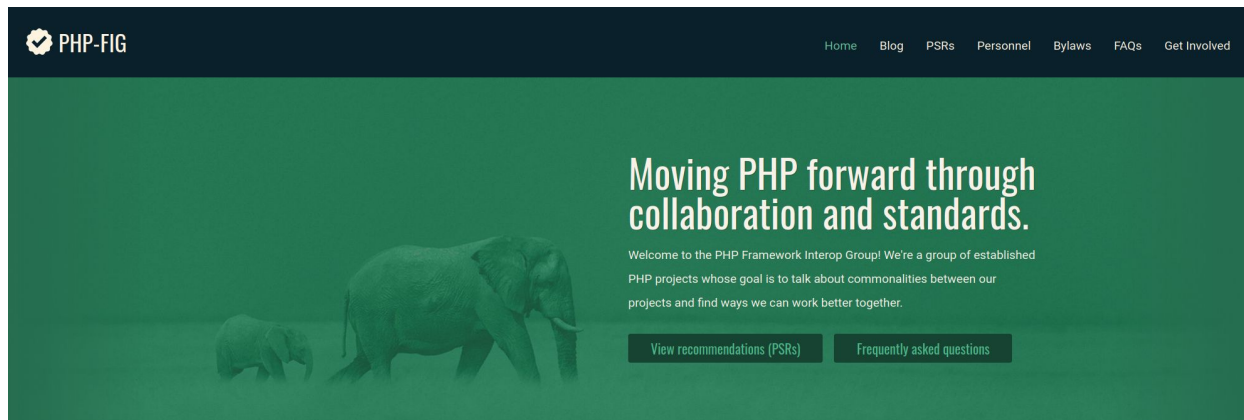
- [CodeIgniter](#)
- [Laravel](#)
- [Slim](#)
- [Symfony](#)
- [Laminas](#) (ex Zend Framework)



Symfony

# PSR-7

- Standard del progetto [PHP Framework Interop Group](#) (PHP-FIG) per definire delle interfacce comuni per la gestione dei messaggi HTTP



# Interfacce PSR-7

- Psr\Http\Message\MessageInterface
- Psr\Http\Message\RequestInterface
- Psr\Http\Message\ResponseInterface
- Psr\Http\Message\ServerRequestInterface
- Psr\Http\Message\StreamInterface
- Psr\Http\Message\UploadedFileInterface
- Psr\Http\Message\UriInterface
- Github: [psr/http-message](https://github.com/php-fig/http-message)

# Esempio

```
// Restituisce un array vuoto se non trovato:  
$header = $request->getHeader('Accept');  
  
// Restituisce una stringa vuota se non trovato:  
$header = $request->getHeaderLine('Accept');  
  
// Verifica la presenza di un header:  
if (! $request->hasHeader('Accept')) {}  
  
// Restituisce i parametri in query string  
$query = $request->getQueryParams();
```



# Immutabilità

- Le richieste PSR-7 sono degli oggetti **immutabili** (immutability)
- Per modificare un dato è necessario creare una nuova istanza dell'oggetto

```
$response = $response->withStatus(418, "I'm a teapot");  
  
$response = $response->withBody($body);
```

- Solo gli oggetti stream di PSR-7 non sono immutabili

# Librerie PSR-7

- [guzzle/psr7](#)
- [laminas/laminas-diactoros](#)
- [slim/slim](#)
- [nyholm/psr7](#)

# SimpleMVC

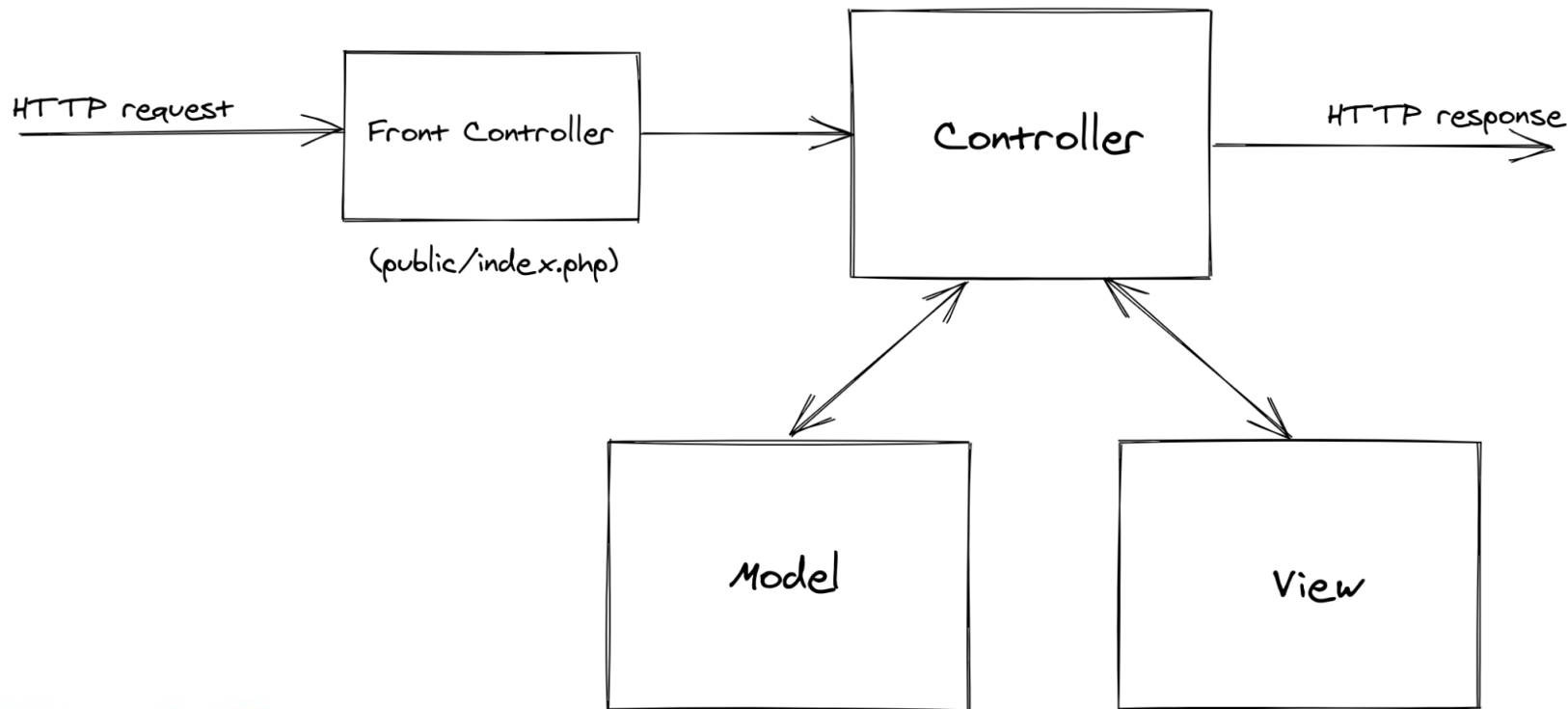
- Microframework open source per scopi didattici
- Disponibile su github: [ezimuel/simplemvc](https://github.com/ezimuel/simplemvc)
- Implementa l'architettura MVC
- Basato su Dependency Injection ([PHP-DI](#)), Routing ([FastRoute](#)), PSR-7 ([nyholm/psr7](#)), Template ([Plates](#))

# Installazione

- L'installazione avviene tramite composer con il seguente comando:

```
composer create-project ezimuel/simple-mvc
```

# Schema MVC



# DEMO

## Esercizio (da consegnare)

- Utilizzando [SimpleMVC](#) realizzare una pagina di Login (email, password) per l'accesso ad un'area riservata del sito (es. una pagina di amministrazione)
- Le credenziali dell'utente devono essere registrate opportunamente in un database MySQL. L'area riservata deve essere gestita con le sessioni PHP.

# Grazie dell'attenzione!

Per informazioni:

[enrico.zimuel@its-ictpiemonte.it](mailto:enrico.zimuel@its-ictpiemonte.it)