

Dependency injection

Corso Backend System Integrator
Modulo **Programmazione PHP**

Docente: Dott. Enrico Zimuel

in collaborazione con:



REGIONE
PIEMONTE

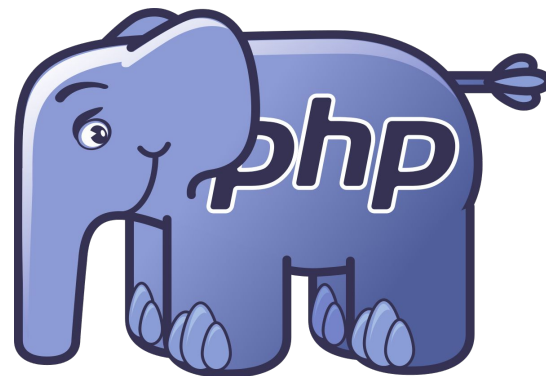
per una crescita intelligente,
sostenibile ed inclusiva

www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

Programma

- Dependency injection
- PHP-DI
- Autowiring
- Configurazione del container



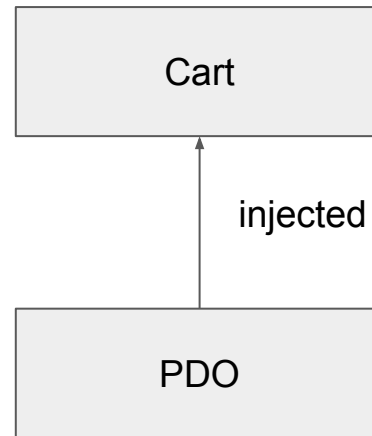
Dependency injection

- La definizione esplicita delle dipendenze di una classe con il passaggio delle stesse in costruzione o tramite setter/getter

Esempio

```
class Cart
{
    public function __construct(PDO $pdo)
    {
        $this->pdo = $pdo;
    }
    // ...
}

$pdo = new PDO(/* parametri */);
$cart = new Cart($pdo);
```



Dependency Injection Container (DIC)

- I **DIC** consentono di archiviare tutte le dipendenze in un unico repository, denominato **Container**
- Condividendo questo unico repository (classe) in tutto il progetto si facilita la fase di sviluppo
- I DIC possono offrire la **risoluzione automatica delle dipendenze**, ossia risolvere tutte le dipendenze tra le classi

Librerie in PHP

- Alcune librerie che implementano un DIC:
 - [Aura.Di](#)
 - [Illuminate Container](#)
 - [PHP-DI](#)
 - [Pimple](#)
 - [The PHP League Container](#)
 - [Nette Di](#)
 - [Laminas-di](#)

PHP-DI

- **PHP-DI** = the dependency injection container for humans
- Funzionalità di **autowire**: l'abilità di un container di creare e iniettare automaticamente le dipendenze
- Utilizzo della funzionalità [Reflection](#) di PHP: reverse-engineering di classi
- Configurazione del container tramite annotazioni o codice PHP

Installazione

```
composer require php-di/php-di
```


Esempio: autowire

```
class Foo {  
    private $bar;  
    public function __construct (Bar $bar)  
    {  
        $this->bar = $bar;  
    }  
}  
  
class Bar {}  
  
$container = new DI\Container();  
$foo = $container->get('Foo');
```

Esempio: addDefinitions()

```
use Psr\Container\ContainerInterface;  
$builder = new \DI\ContainerBuilder();  
  
$builder->addDefinitions([  
    'dsn' => 'sqlite:db.sq3',  
    'PDO' => function(ContainerInterface $c) {  
        return new PDO($c->get('dsn'));  
    }  
]);  
  
$container = $builder->build();  
$cart = $container->get('Cart');
```

Esempio che utilizza la
classe **Cart** con **PDO**
come dipendenza
esplicita in costruzione

Esempio: configurazione

```
// config.php
use Psr\Container\ContainerInterface;
return [
    'dsn' => 'sqlite:db.sqlite3',
    'PDO' => function(ContainerInterface $c) {
        return new PDO($c->get('dsn'));
    }
];
```

```
// example.php
$builder = new \DI\ContainerBuilder();
$builder->addDefinitions('config.php');
$container = $builder->build();
$cart = $container->get('Cart');
```

Esercizio

Utilizzare la libreria **PHP-DI** per istanziare la classe di **Autenticazione** dell'esercizio della Login tramite **Dependency Injection Container**

Suggerimento: scrivere un file di configurazione per passare i parametri di connessione al MySQL per PDO

Grazie dell'attenzione!

Per informazioni:

enrico.zimuel@its-ictpiemonte.it