



# Symfony

Damian Łukasik  
Młodszy programista php  
Provider Sp. z o.o.

# O czym będzie mowa?

- Symfony
- MVC
- Twig – system szablonów
- Aplikacja
  - Struktura
  - Rozwój (praktyczny przykład tworzenia kontrolera wraz z widokiem + commit do repozytorium)

# Symfony

- Framework PHP
- Aplikacje webowe
- MVC
- Pierwsza wersja - Październik 2005
- **Sensio**Labs
- Aktualna wersja 4.3.4
- Wieloplatformowość
- Licencja X11
- Wolne oprogramowanie



Symfony



# Symfony



- Cele
  - Przyspieszenie procesu tworzenia aplikacji internetowych
  - Konserwacja aplikacji internetowych
  - Zastąpienie powtarzających się zadań przy pisaniu kodu
  - Budowanie niezawodnych aplikacji dla przedsiębiorstw
  - Zapewnienie programistom pełnej kontroli nad konfiguracją
  - Testowanie, debugowanie i dokumentowanie projektów poprzez dostarczanie dodatkowych narzędzi



# Symfony



- Cechy

- Bazuje na wzorcu projektowym MVC
- Programowanie zorientowane obiektowo
- Łatwość w instalacji oraz konfiguracji na większości platform
- Niezależność od systemu bazodanowego
- Zgodność z najlepszymi standardami oraz wzorcami budowy aplikacji internetowych
- Łatwość rozbudowy oraz możliwość integracji z innymi bibliotekami
- Wbudowana ochrona przed atakami CSRF oraz XSS
- Walidacja formularzy i treści
- Zarządzanie sesjami
- Wykorzystanie technologii scaffoldingu.
- Wbudowana internacjonalizacja (i18n)

```
<?php
class User {
    private $name;
    public $gender;
    protected $age;
}
```

# Symfony

- Framework Symfony był silnie inspirowany projektem **Spring Framework**
- Wykorzystuje istniejące projekty open source PHP jako część frameworka, w tym:
  - **Doctrine** – obsługa komunikacji z bazą danych
  - **PDO** – Warstwa abstrakcji bazy danych (DBAL)
  - **PHPUnit** – testy jednostkowe
  - **Twig** – system szablonów
  - **Swift mailer** – wysyłanie maili
- Korzysta także z własnych komponentów, które są dostępne w witrynie *Symfony Components*
  - Symfony Yaml
  - Symfony Event Dispatcher
  - Symfony Dependency Injector



# MVC

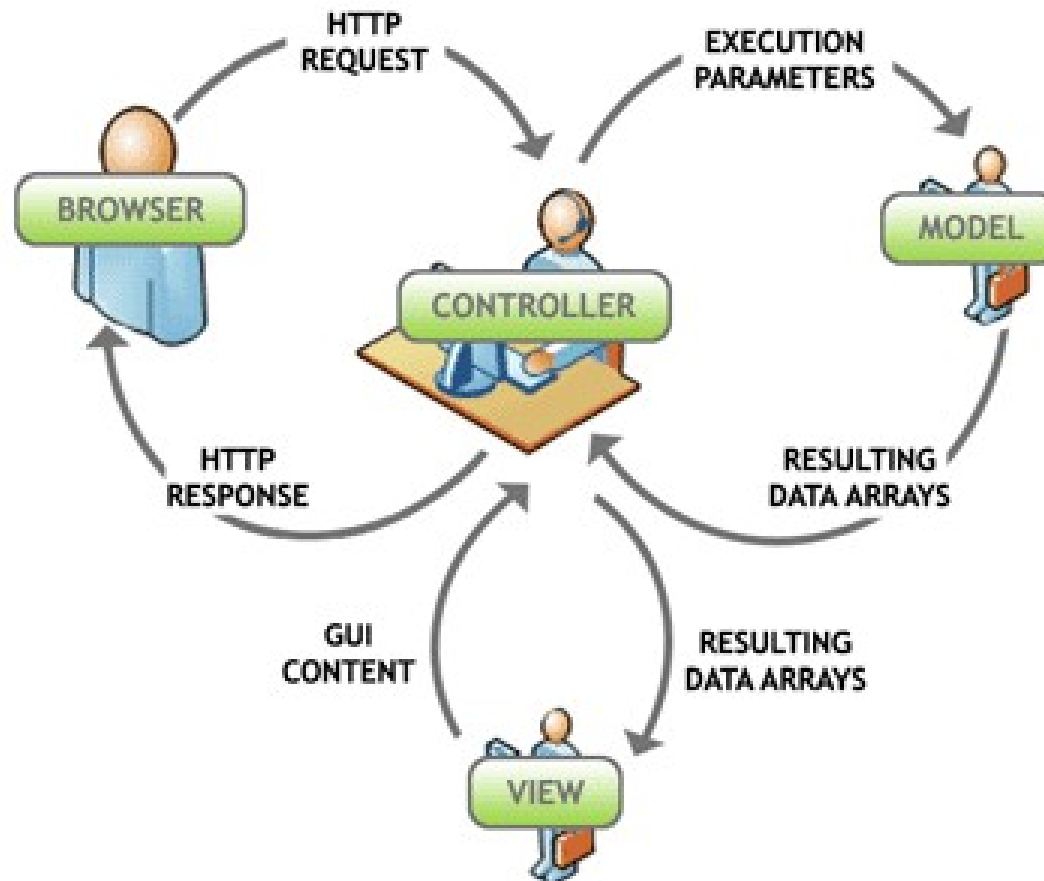


- **Model-View-Controller**

- Wzorzec projektowy
- **Model** – jest pewną reprezentacją problemu bądź logiki aplikacji.
- **Widok** – opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika. Może składać się z podwidoków odpowiedzialnych za mniejsze części interfejsu.
- **Kontroler** – przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania, zarządzając aktualizacje modelu oraz odświeżenie widoków.

# MVC

- **Model-View-Controller**





# Twig

- Twig
  - System szablonów
  - Generowanie kodu html
  - Oddzielenie logiki biznesowej od warstwy prezentacji aplikacji.



# Inne systemy szablonów

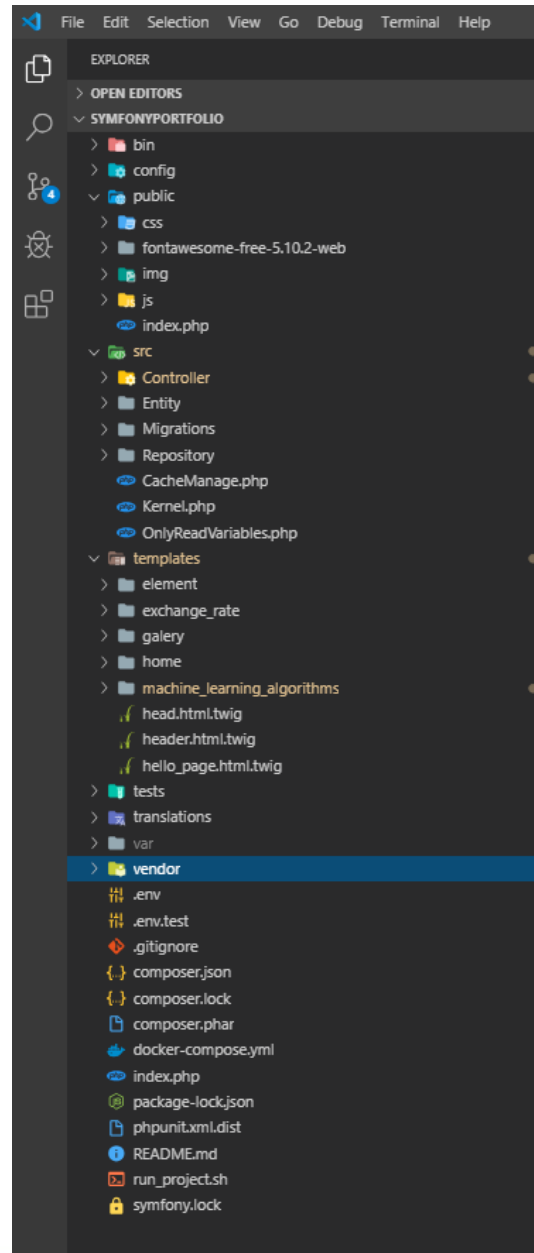
A yellow pencil and a pink eraser are positioned in the top right corner of the slide, appearing as if they are on a piece of paper.

- SMARTY
- Savant
- PHPTAL
- Open Power Template

# Aplikacja

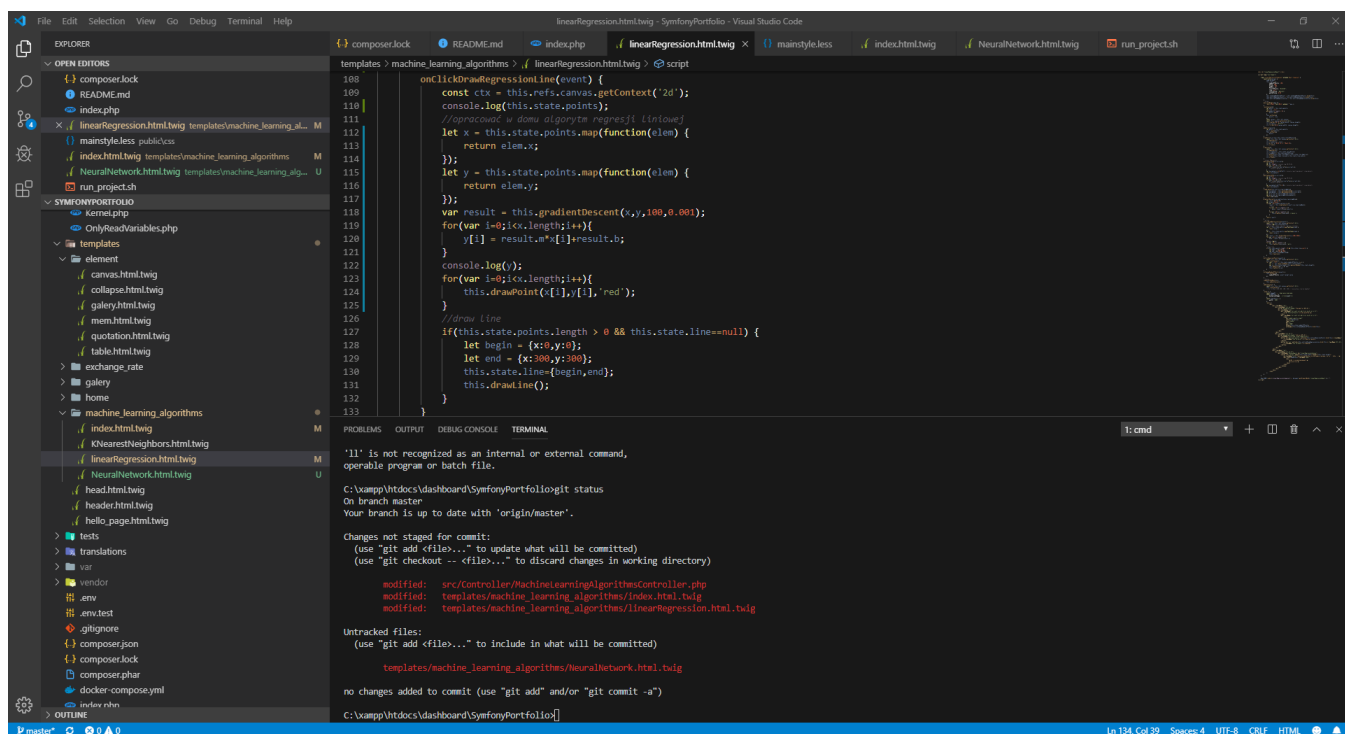
- Struktura

- bin
- config
  - packages
  - routes
- public
- src
  - Controller
- templates
- tests
- translations
- var
  - cache
  - log
- vendor



# Aplikacja

- Prezentacja działającej aplikacji na dockerze w Visual Studio Code



The screenshot displays the Visual Studio Code interface with a project named 'SymfonyPortfolio'. The Explorer sidebar on the left shows the file structure, including a 'machine\_learning\_algorithms' directory. The main editor area shows the 'linearRegression.html.twig' file, which contains JavaScript code for a linear regression model. The code includes a 'onClickDrawRegressionLine' function that calculates the gradient descent and draws a line on a canvas. The terminal at the bottom shows the output of a 'git status' command, indicating that the current branch is up to date with 'origin/master'.

```
108 onClickDrawRegressionLine(event) {
109     const ctx = this.refs.canvas.getContext('2d');
110     console.log(this.state.points);
111     //opracowac w domu algorytm regresji (liniowej)
112     let x = this.state.points.map(function(element) {
113         return element.x;
114     });
115     let y = this.state.points.map(function(element) {
116         return element.y;
117     });
118     var result = this.gradientDescent(x,y,100,0.001);
119     for(var i=0;i<x.length;i++){
120         y[i] = result.a*x[i]+result.b;
121     }
122     console.log(y);
123     for(var i=0;i<x.length;i++){
124         this.drawPoint(x[i],y[i],'red');
125     }
126     //draw line
127     if(this.state.points.length > 0 && this.state.line==null) {
128         let begin = {x:0,y:0};
129         let end = {x:300,y:300};
130         this.state.line={begin,end};
131         this.drawLine();
132     }
133 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

'11' is not recognized as an internal or external command,  
operable program or batch file.

C:\xampp\htdocs\dashboard\SymfonyPortfolio>git status  
On branch master  
Your branch is up to date with 'origin/master'.

Changes not staged for commit:  
(use "git add <file>..." to update what will be committed)  
(use "git checkout -- <file>..." to discard changes in working directory)

modified: src/Controller/MachineLearningAlgorithmsController.php  
modified: templates/machine\_learning\_algorithms/index.html.twig  
modified: templates/machine\_learning\_algorithms/linearRegression.html.twig

Untracked files:  
(use "git add <file>..." to include in what will be committed)

templates/machine\_learning\_algorithms/NeuralNetwork.html.twig

no changes added to commit (use "git add" and/or "git commit -a")

C:\xampp\htdocs\dashboard\SymfonyPortfolio>



Dziękuję za uwagę

<https://www.w3schools.com/>

<https://getcomposer.org/>

<https://twig.symfony.com/>

<https://code.visualstudio.com/>

<https://symfony.com/>