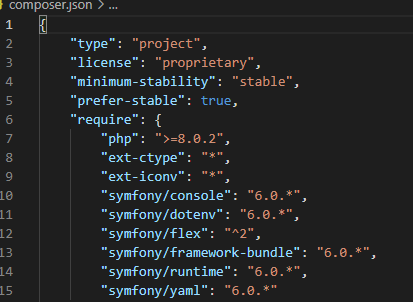
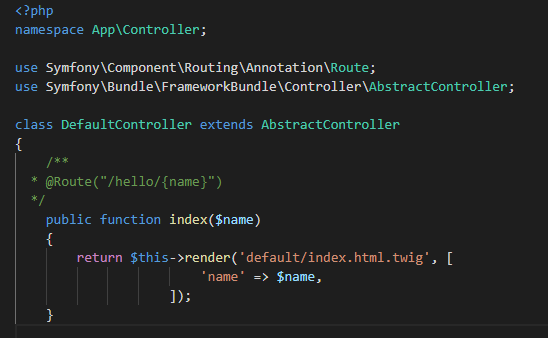
Лабораторная работа по Symfony №2

**Symfony: Начните с микро!**

Кроме случаев, когда вы строите чистый API (об этом позже!), вы скорее всего захотите отобразить HTML. Чтобы сделать это, вы будете использовать Twig. Twig - это гибкий, быстрый и безопасный шаблонизатор для PHP. Он делает ваши шаблоны более читаемыми и компактными; он также делает их более дружественными для веб-дизайнеров.



composer require twig

}

Расширив AbstractController, у вас теперь есть доступ к перечню методов сокращения и инструментов, вроде render().

Создаем новый шаблон:

{# templates/default/index.html.twig #}

<h1>Привет, {{ name }}!</h1>

Добавляем:

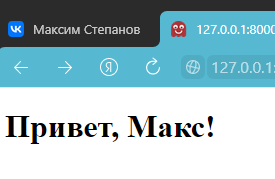
{# templates/default/index.html.twig #}

{% extends 'base.html.twig' %}

{% block body %}

<h1>Привет, {{ name }}!</h1>

{% endblock %}



Профилировщик: Рай отладки

Одна из самых крутых функций Symfony ещё даже не установлена! Давайте это исправим:

$ composer require profiler

Да! Это ещё один союзник! И Flex также устанавливает другой рецепт, который автоматизирует конфигурацию Профилировщика Symfony. Какой результат? Обновите!

## Богатая поддержка API[¶](https://symfony.com.ua/doc/current/quick_tour/flex_recipes.html#api)

Вы создаете API? Вы можете уже с лёгкостью вернуть JSON из любого контроллера:

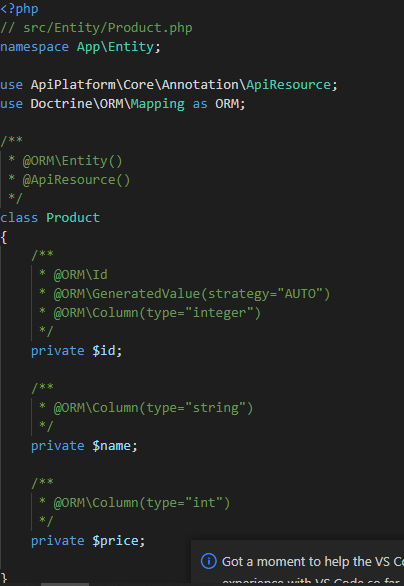


Путь: <http://localhost:8000/api/hello/Makc>

Но для действительно богатого API, попробуйте установить [Платформу Api](https://api-platform.com/):

composer require api

Создайте сущность Product и дайте её аннотацию @ApiResource():



Готово! У вас теперь есть конечные точки для перечисления, добавления, обновления и удаления продуктов!

## Удаление рецептов

Всё ещё не убеждены? Не проблема: удалите библиотеку:

composer remove api

Flex удалит рецепты: удаление файлов и отмена изменений для возвращения вашего приложения обратно в исходное состояние.

