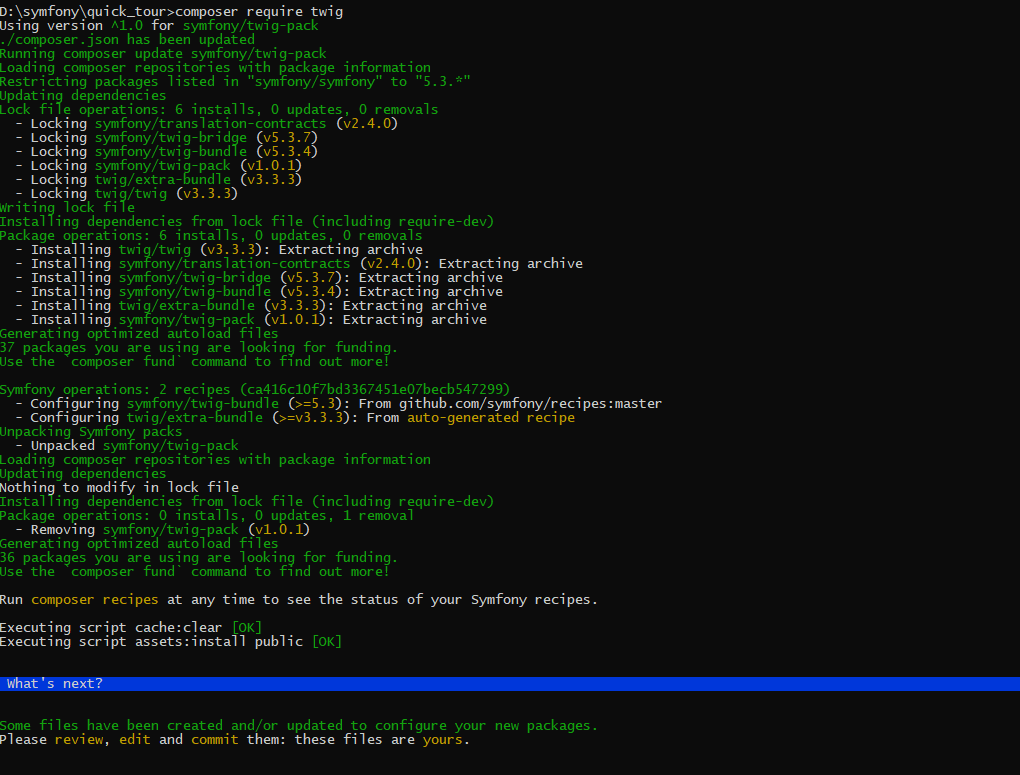
Выполнил Кадомцев Алексей

Рецепты и дополнительные имена Flex

Twig: Отображение шаблона

Благодаря Flex, после одной команды вы можете незамедлительно начать использовать Twig:

<?php

namespace App\Controller;

use Symfony\Component\Routing\Annotation\Route;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class DefaultController extends AbstractController

{

    /\*\*

     \* @Route("/hello/{name}")

     \*/

public function index($name)

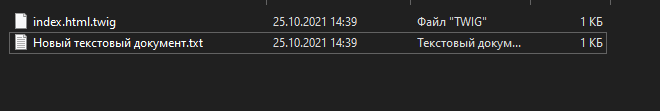
{

return $this->render('default/index.html.twig',['name'=>$name,]);

}

}

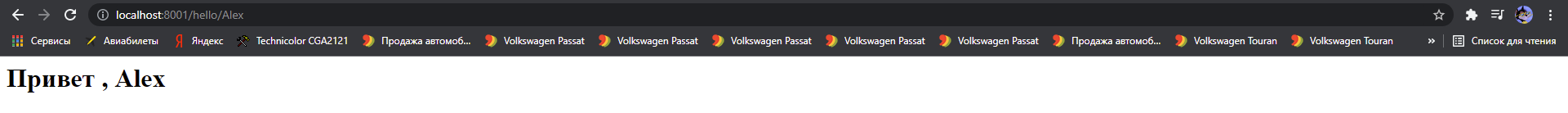
Расширив AbstractController, у вас теперь есть доступ к перечню методов сокращения и инструментов, вроде render(). Создайте новый шаблон:



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

<h1>Привет , {{ name }}</h1>

Вот и всё! Синтаксис {{ name }} отобразит переменную name, которая передаётся из контроллера. Если вы новичок в Twig, то добро пожаловать! Вы узнаете больше о его синтаксисе и силе позже.



Но, прямо сейчас. страница содержит только тег h1. Чтобы предоставить ему макет HTML, расширьте index.html.twig:

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

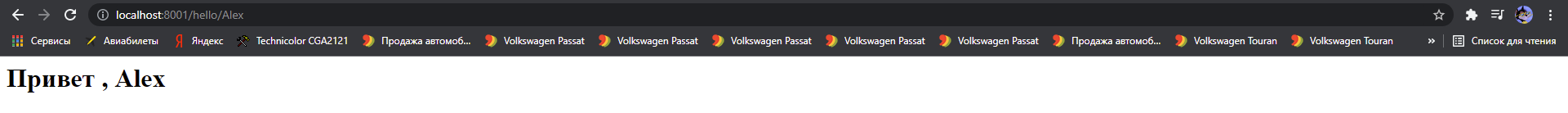
{% extends 'base.html.twig' %}

{% block body %}

<h1>Hello, {{ name }}</h1>

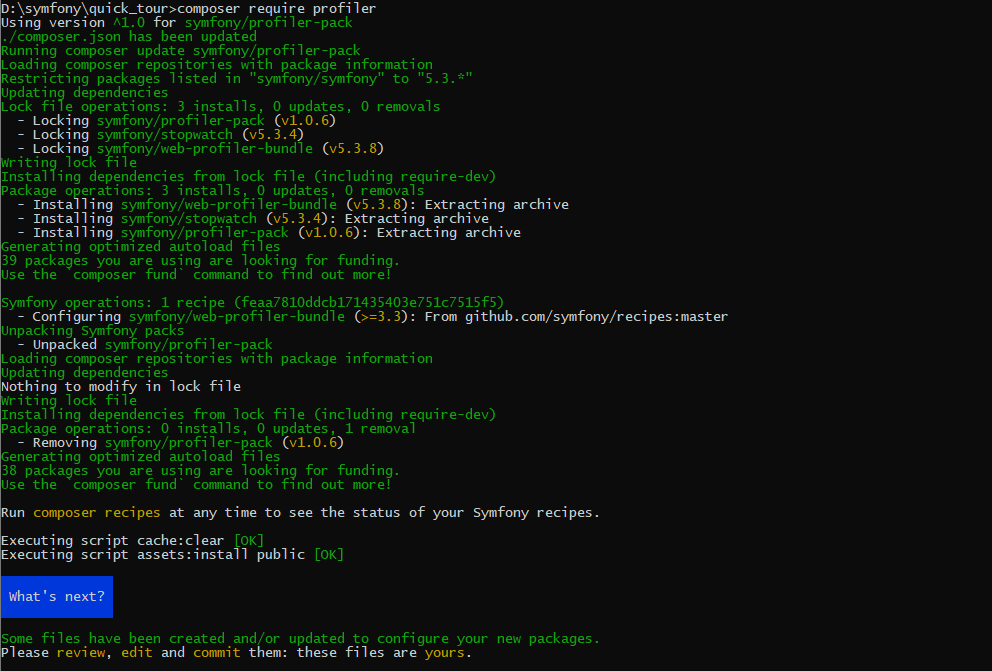
{% endblock %}

Это называется наследование шаблона: наша страница теперь наследует структуру HTML из base.html.twig.

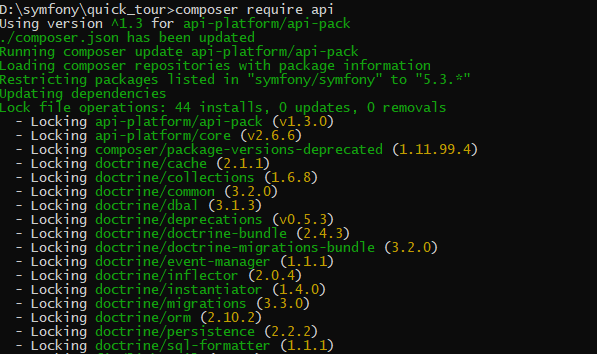


Профилировщик: Рай отладки

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



Богатая поддержка API

Вы строите API? Вы можете уже с лёгкостью вернуть JSON из любого контроллера: Но для действительно богатого API, попробуйте установить Платформу Api:

Это дополнительное имя api-platform/api-pack, которое имеет зависимости в нескольких других пакетах, вроде компонентов Symfony Валидатор и Безопасность, а также Doctrine ORM. На самом деле, Flex установил 5 рецептов!

Но, как обычно, мы можем сразу же начать использовать новую библиотеку. Хотите создать богатый API для таблицы product? Создайте сущность Product и дайте её аннотацию @ApiResource():

<?php

// src/Entity/Product.php

namespace App\Entity;

use ApiPlatform\Core\Annotation\ApiResource;

use Doctrine\ORM\Mapping as ORM;

/\*\*

 \* @ORM\Entity()

 \* @ApiResource()

 \*/

class Product

{

    /\*\*

     \* @ORM\Id

     \* @ORM\GeneratedValue(strategy="AUTO")

     \* @ORM\Column(type="integer")

     \*/

    private $id;

    /\*\*

     \* @ORM\Column(type="string")

     \*/

    private $name;

    /\*\*

     \* @ORM\Column(type="int")

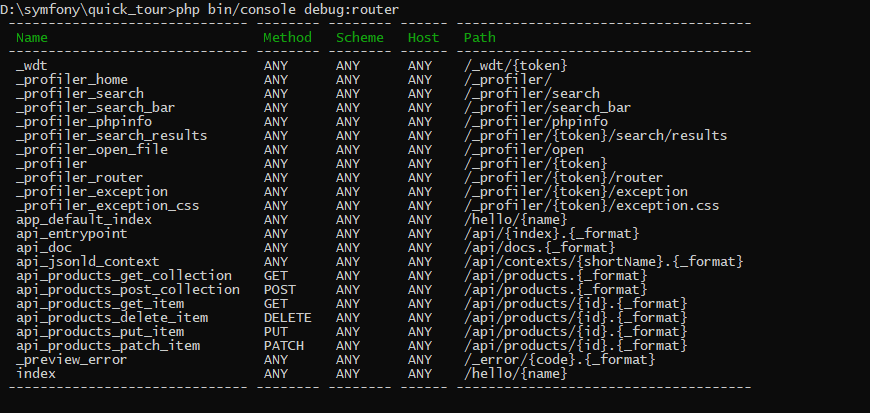
     \*/

    private $price;

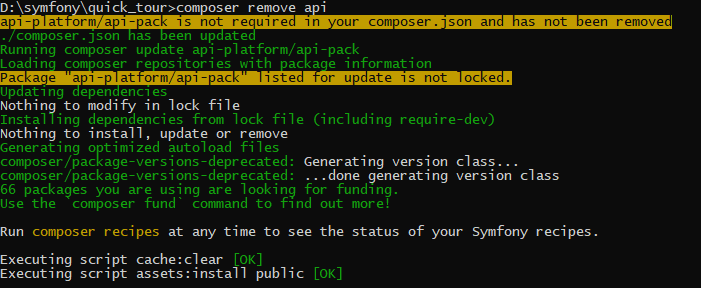
    // ...

}

Прописываем php bin/console debug:router



Легко удаляйте рецепты



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