Молдавский Государственный Университет

Факультет Математики и Информатики

Специальность: Прикладная Информатика

Предмет: Framework. p/u Apl.WEB

Лабораторная работа №1(Symfony)

Выполнил: Грицюк Артем, IA1902

Руководитель: S.Bodrug

Кишинев, 2021

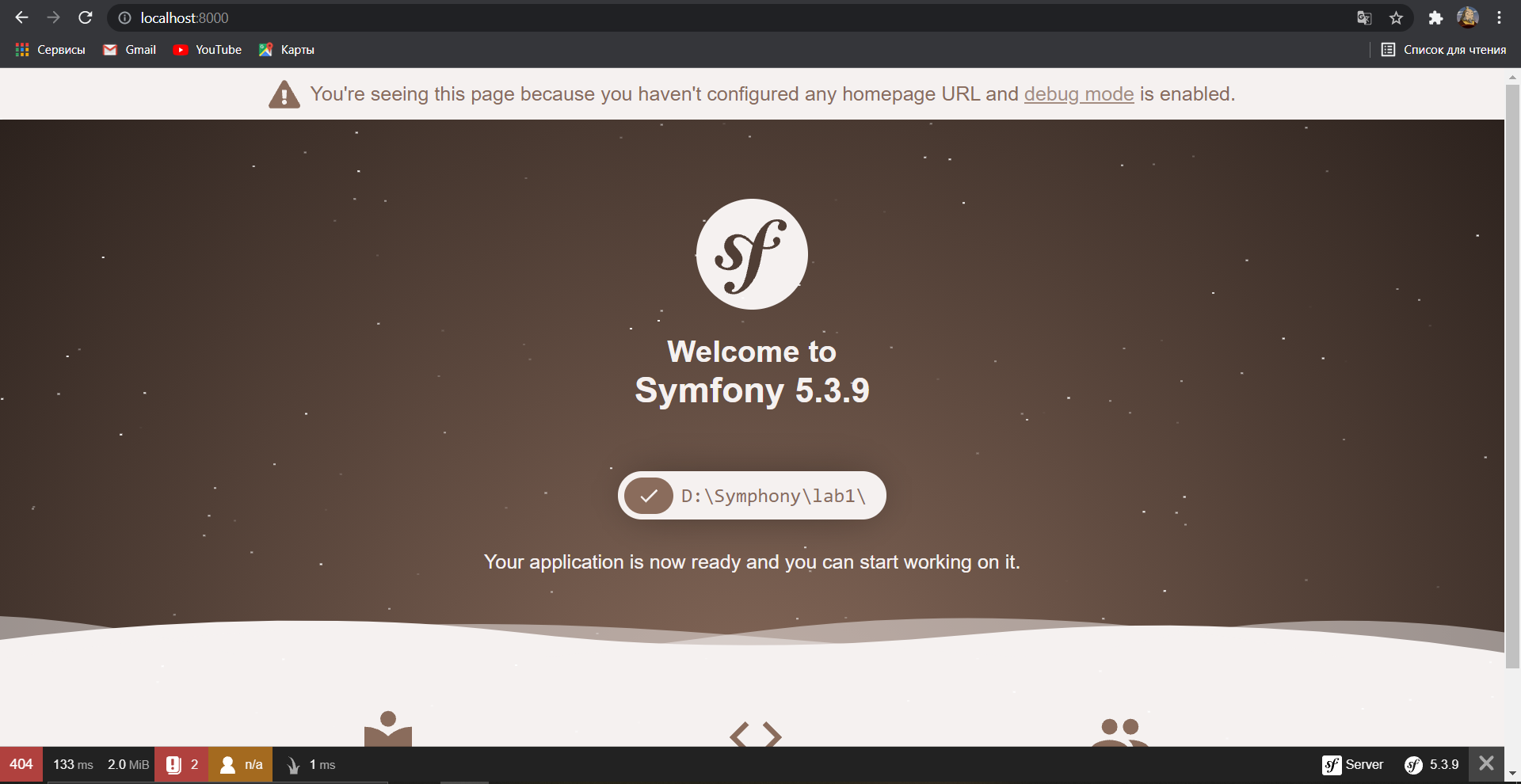
**Создание приложений Symfony**

**Команды:**

symfony new my\_project

cd my\_project

symfony server:start



# **Создайте свою первую страницу в Symfony**

**Код:**

<?php

// src/Controller/LuckyController.php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

class LuckyController

{

public function number()

{

$number = random\_int(0, 100);

return new Response(

'<html><body>Lucky number: '.$number.'</body></html>'

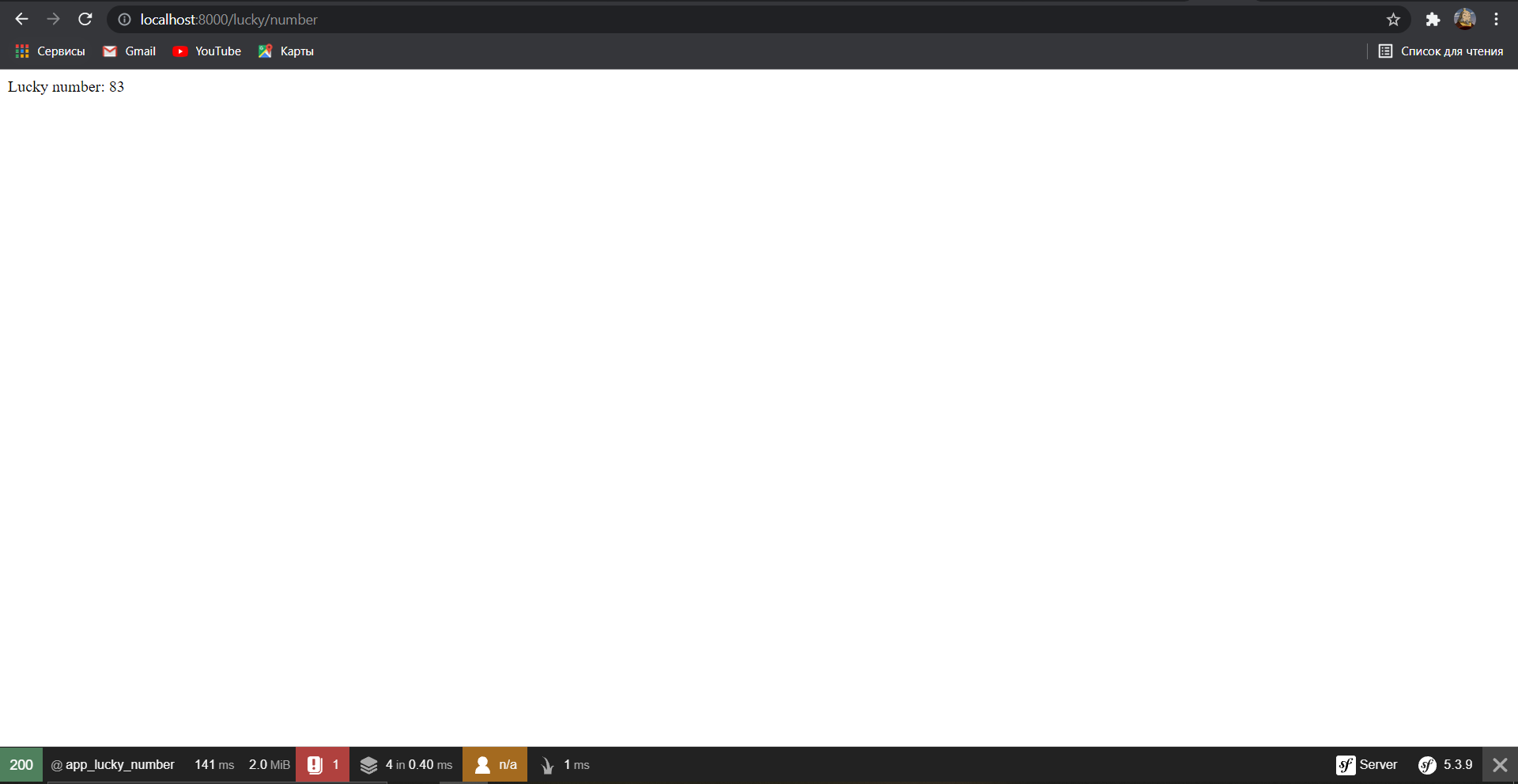
); }}

**Теперь вам нужно связать эту функцию контроллера с общедоступным URL (например /lucky/number), чтобы number()метод выполнялся, когда пользователь просматривает его.**

app\_lucky\_number:

path: /lucky/number

controller: App\Controller\LuckyController::number



**Маршруты аннотаций**

<?php

// src/Controller/LuckyController.php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

class LuckyController

{

    public function number(): Response

    {

        $number = random\_int(0, 100);

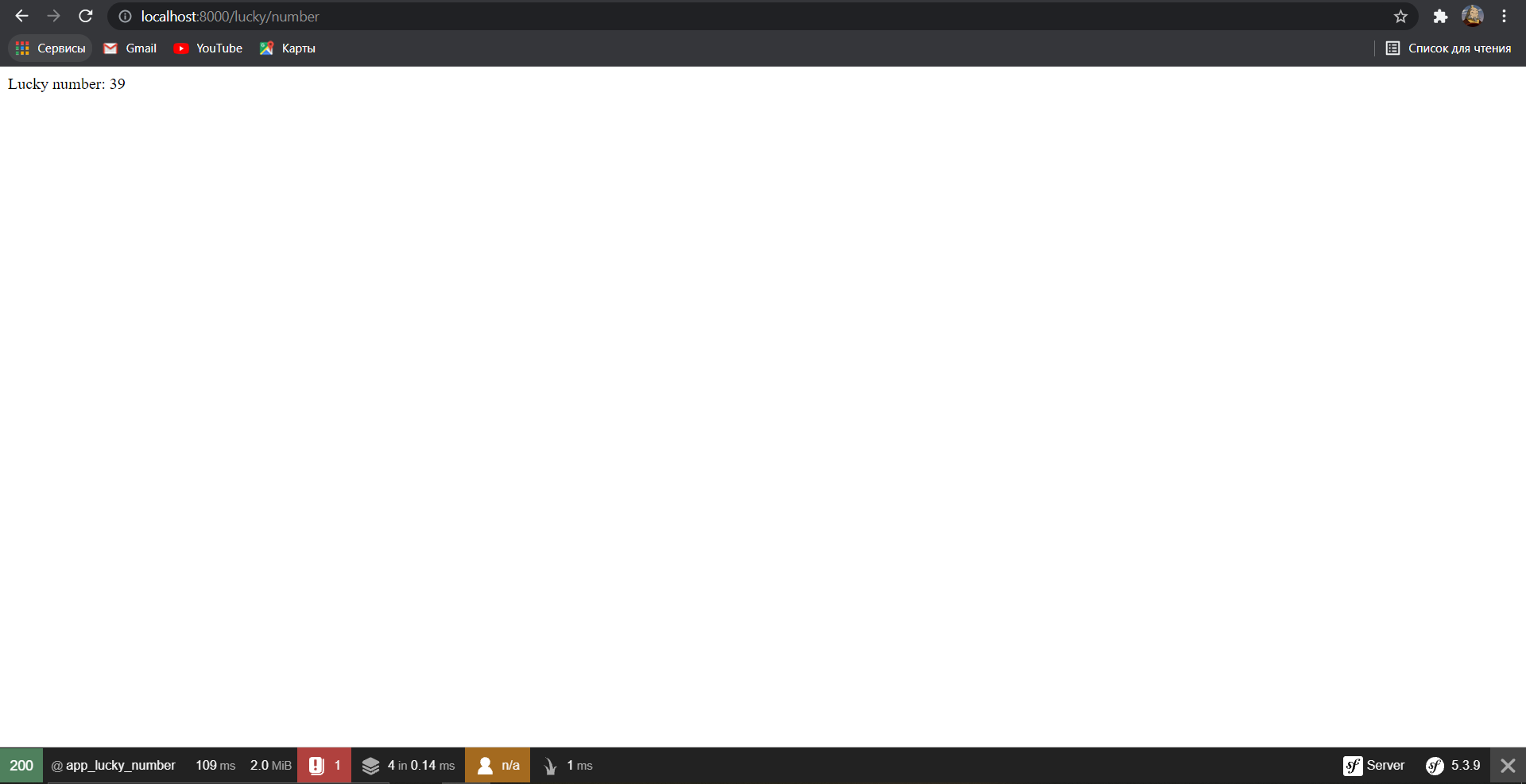
        return new Response(

            '<html><body>Lucky number: '.$number.'</body></html>'

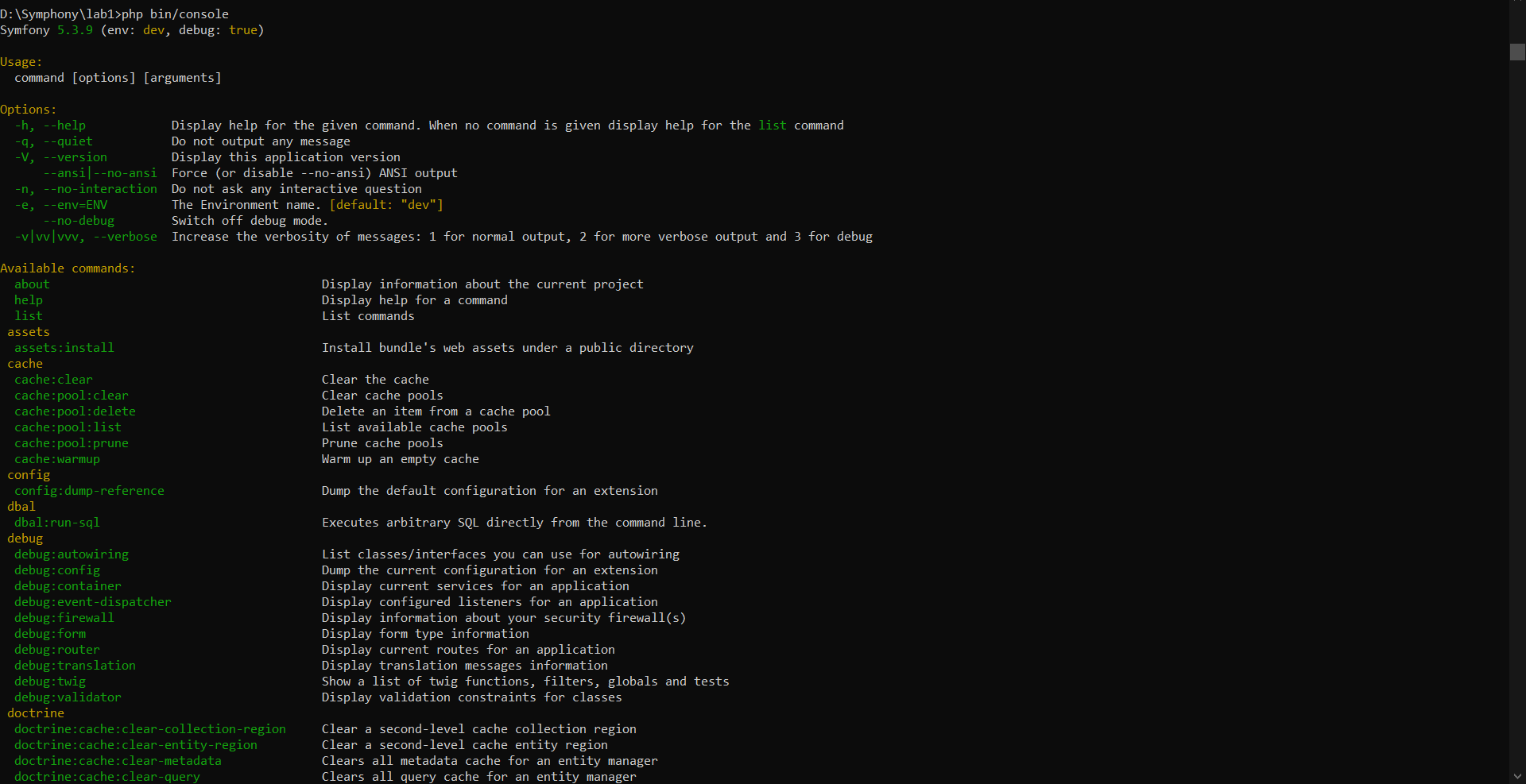
        );

    }

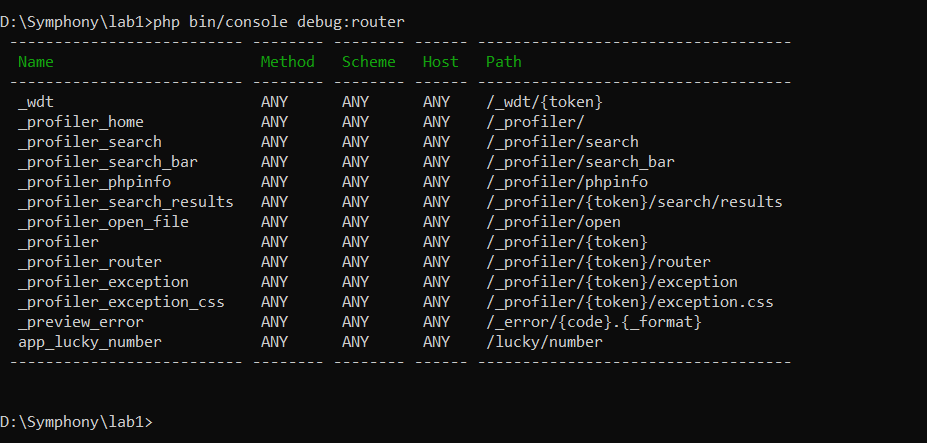
}



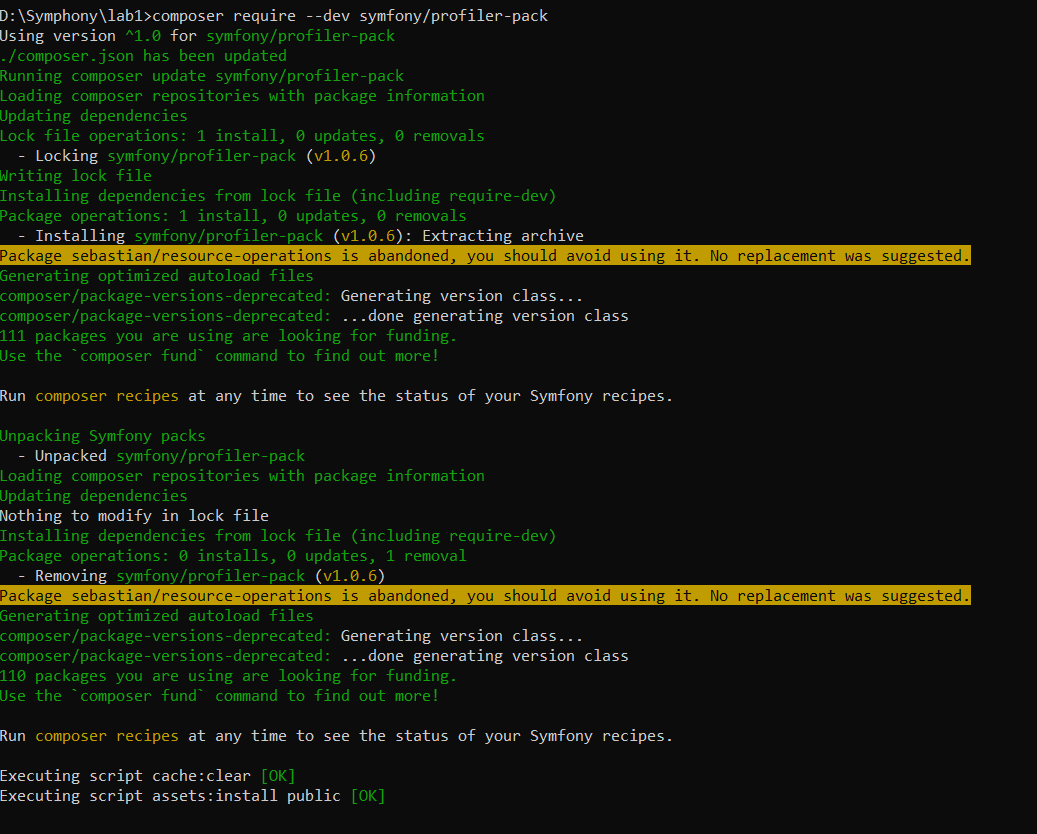
**php bin / console**

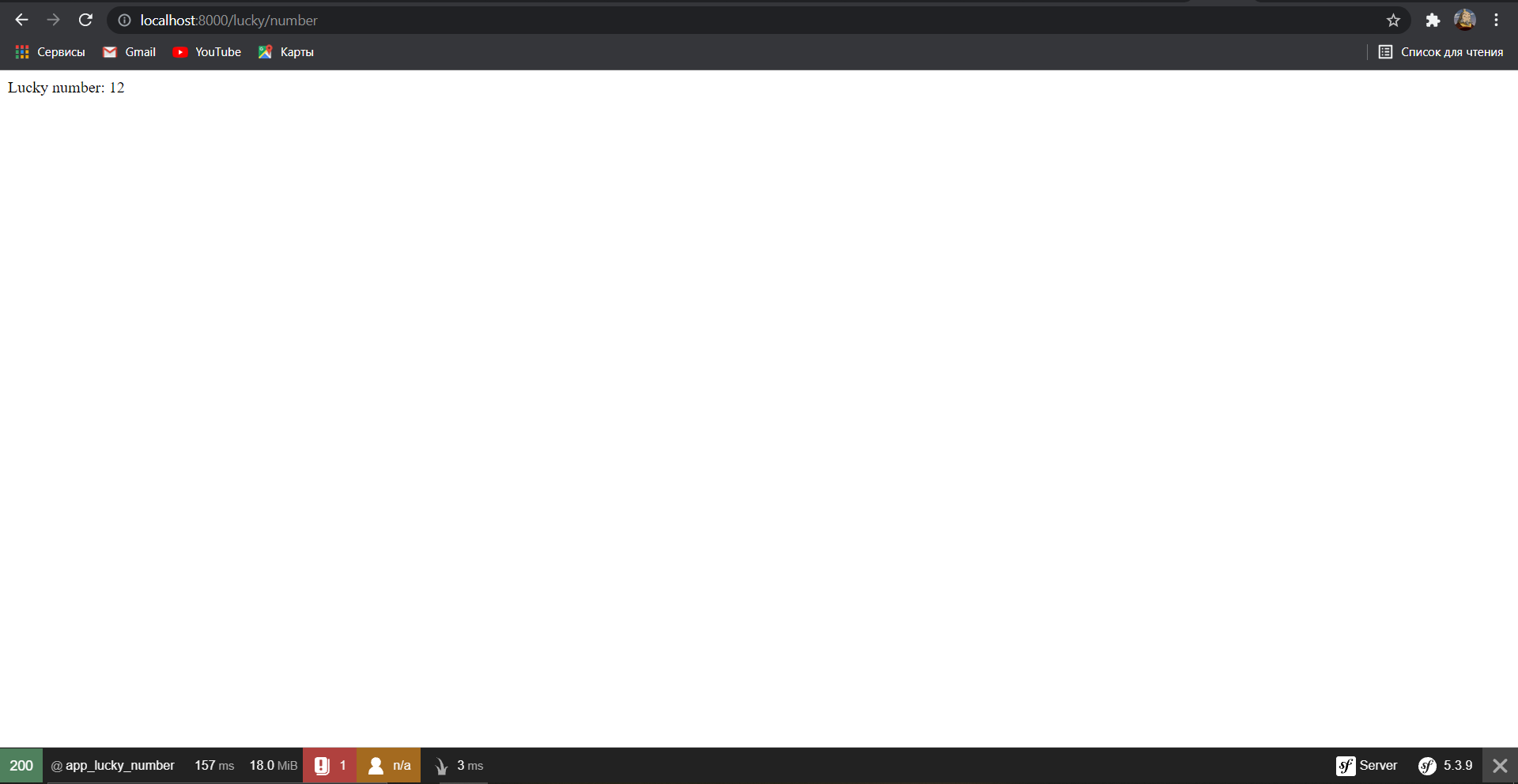


**php bin/console debug:router**



**composer require --dev symfony/profiler-pack**





**Убедитесь, что это LuckyController расширяет базовый [AbstractController](https://github.com/symfony/symfony/blob/master/src/Symfony/Bundle/FrameworkBundle/Controller/AbstractController.php" \o "Symfony\\Bundle\\FrameworkBundle\\Controller\\AbstractController) класс Symfony :**

<?php

// src/Controller/LuckyController.php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class LuckyController extends AbstractController

{

    /\*\*

     \* @Route("/lucky/number")

     \*/

    public function number()

    {

        $number = random\_int(0, 100);

        return $this->render('lucky/number.html.twig', [

            'number' => $number,

        ]);

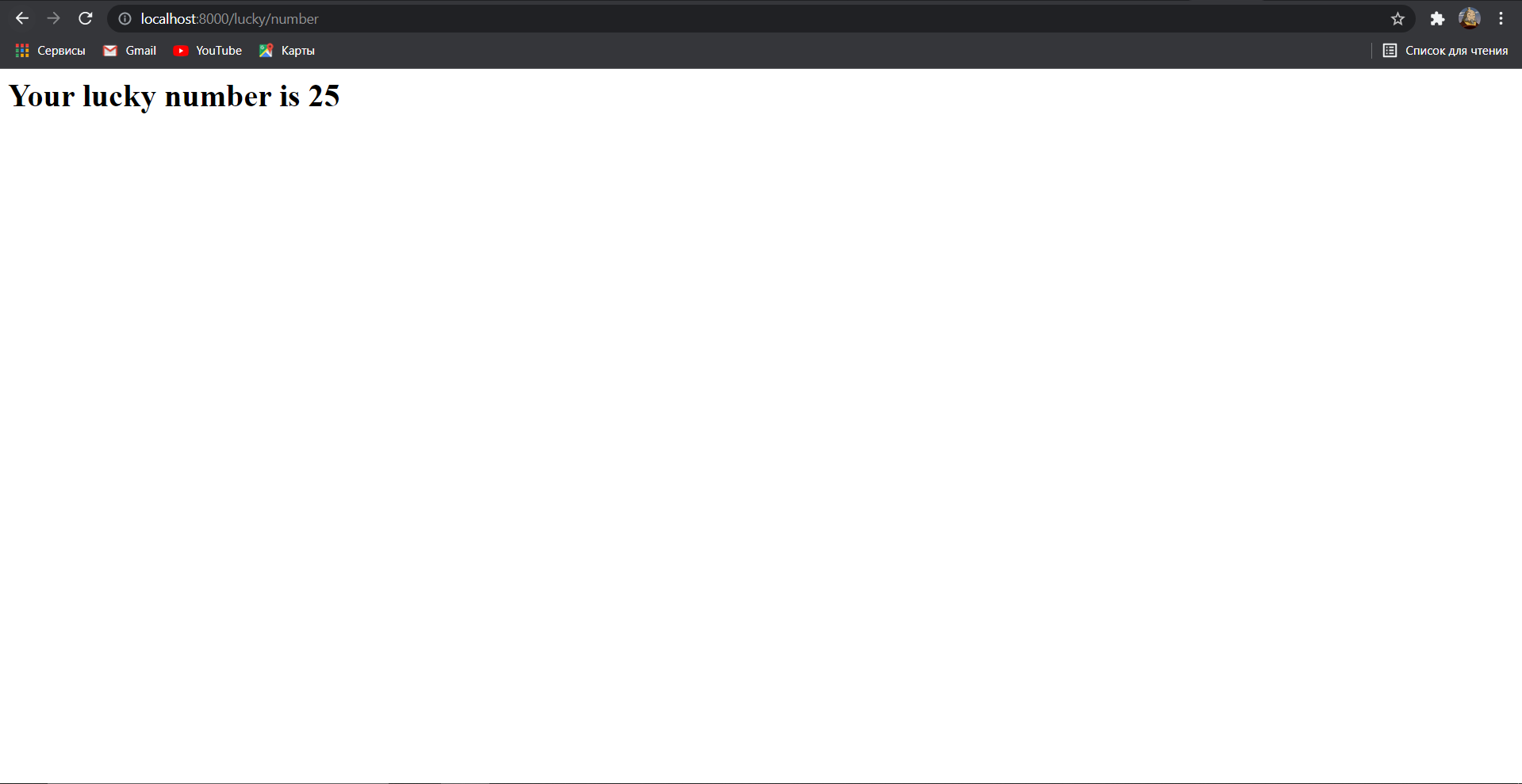
    }

}

**Файлы шаблонов находятся в templates/каталоге, который был создан для вас автоматически при установке Twig. Создайте новый templates/lucky каталог с новым number.html.twig файлом внутри:**

**{# templates/lucky/number.html.twig #}**

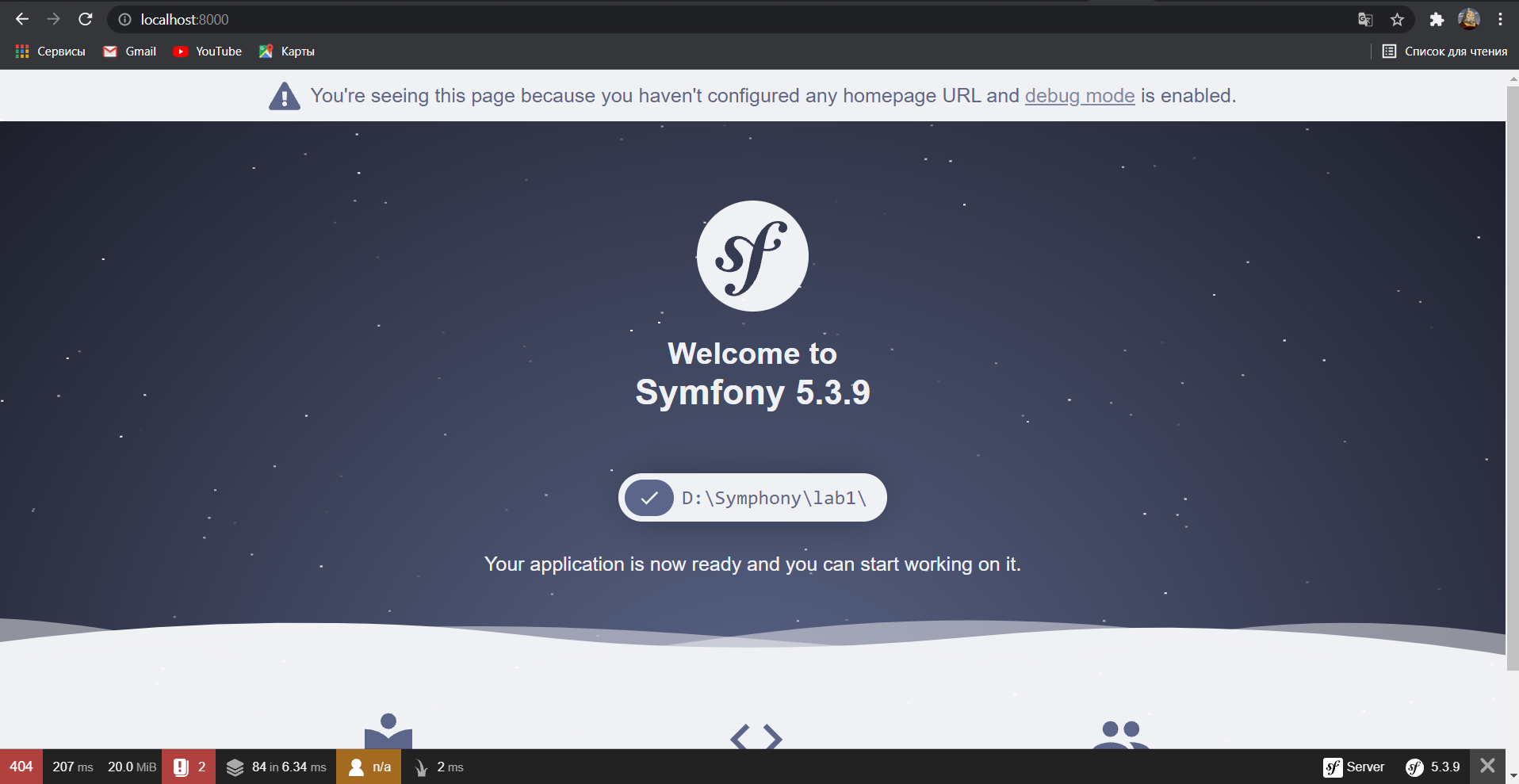
**<h1>Your lucky number is {{ number }}</h1>**



Основы: Маршрут, Контроллер, Ответ

Для начала создадим проект

composer create-project symfony/skeleton quick\_tour



**config/routes.yaml:**

#index:

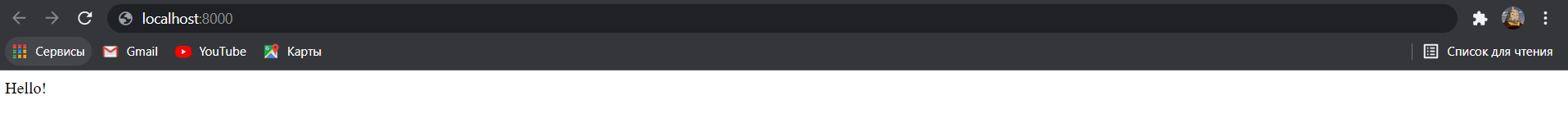
#    path: /

#    controller: App\Controller\DefaultController::index

index:

    path: /

    controller: App\Controller\DefaultController::index



index:

    path: /hello/{name}

    controller: App\Controller\DefaultController::index

<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

class DefaultController

{

    public function index($name)

    {

        return new Response("Привет, $name!");

    }

}

**/hello/\*: {name}**



**Так что давайте установим поддержку аннотаций:**

**composer require annotations**

**Теперь, закомментируйте маршрут YAML**

#index:

#   path: /hello/{name}

#  controller: App\Controller\DefaultController::index

Вместо этого, добавьте маршрут прямо над методом контроллера:

<?php

namespace App\Controller;

use Symfony\Component\Routing\Annotation\Route;

use Symfony\Component\HttpFoundation\Response;

class DefaultController

{

    /\*\*

  \* @Route("/hello/{name}")

  \*/

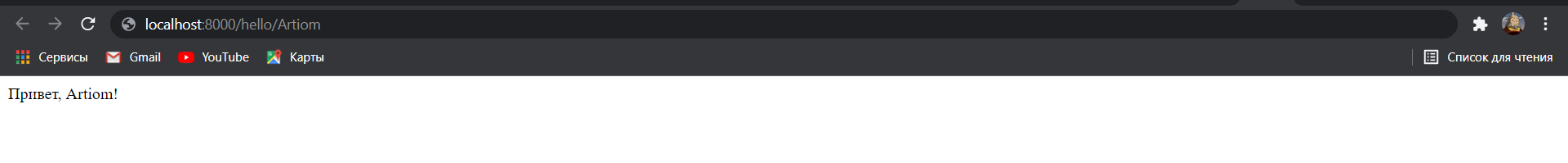
    public function index($name)

    {

        return new Response("Привет, $name!");

    }

}

 **Добавьте ешё один маршрут и метод в DefaultController:**

/\*\*

     \* @Route("/simplicity")

     \*/

    public function simple()

    {

        return new Response(' simple! Легко! Прекрасно!');}

