

Работа со статическими ресурсами

Spring Boot предоставляет автоматическую конфигурацию для отдачи статических ресурсов — например, JS, CSS или картинок. По умолчанию они автоматически отдаются по запросу из директории *src/main/resources/static*. Содержимое этой директории может выглядеть так:

- *src/main/resources/static/images/logo.png*
- *src/main/resources/static/css/style.css*
- *src/main/resources/static/js/app.js*

Если предположить, что сервер стартует на 8080 порту, то получить доступ к этим файлам можно по следующим ссылкам:

- <http://localhost:8080/images/logo.png>
- <http://localhost:8080/css/style.css>
- <http://localhost:8080/js/app.js>

Или то же самое внутри HTML-файлов:

```

<link rel="stylesheet" href="/css/style.css" />
<script src="/js/app.js"></script>
```

Если ресурс не найден, то Spring Boot автоматически возвращает ошибку 404. Страницу с ошибкой можно кастомизировать, для этого необходимо создать файл *404.html* в директории *src/main/resources/public/error*.

index.html

Входную страницу сайта или сервиса на Spring Boot можно получить двумя способами:

- С помощью обработчика маршрута главной страницы <http://localhost:8080>
- С помощью статического файла *index.html* добавленного в директорию *src/main/resources/public*. Spring Boot отдает этот файл автоматически при запросе на <http://localhost:8080>. Это самый распространенный способ при наличии фронтенд-части, создаваемой отдельно от Spring Boot

Рассмотрим пример файла:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home Page</title>
  <link rel="stylesheet" href="/css/style.css" />
</head>
<body>
  <h1>Welcome to Hexlet!</h1>
  
  <script src="/js/app.js"></script>
</body>
</html>
```

Далее →