**ОТЧЕТ**

Для создания веб-приложения был выбран язык **Java** за счет своей универсальности и кросс-платформенности. В нем содержатся все необходимые библиотеки, для создания веб интерфейса. Но кроме библиотек, существует фреймворк, Spring, прямое назначение которого – создание веб приложений. Он включает в себя множество модулей, для быстрой и удобной разработки.

В качестве среды программирования будем использовать **Intelij Idea 2018**. Это свободно-распространяемая среда разработки, она содержит множество полезных функций, таких как поиск синтаксических ошибок, автоматическое подключение нужных библиотек, подсветка синтаксиса, быстрая генерация классов, генерация шаблонных функций и многое другое. Скачать ее можно на оффициальном сайте JetBrains - <https://www.jetbrains.com/ru-ru/idea/download/#section=windows>

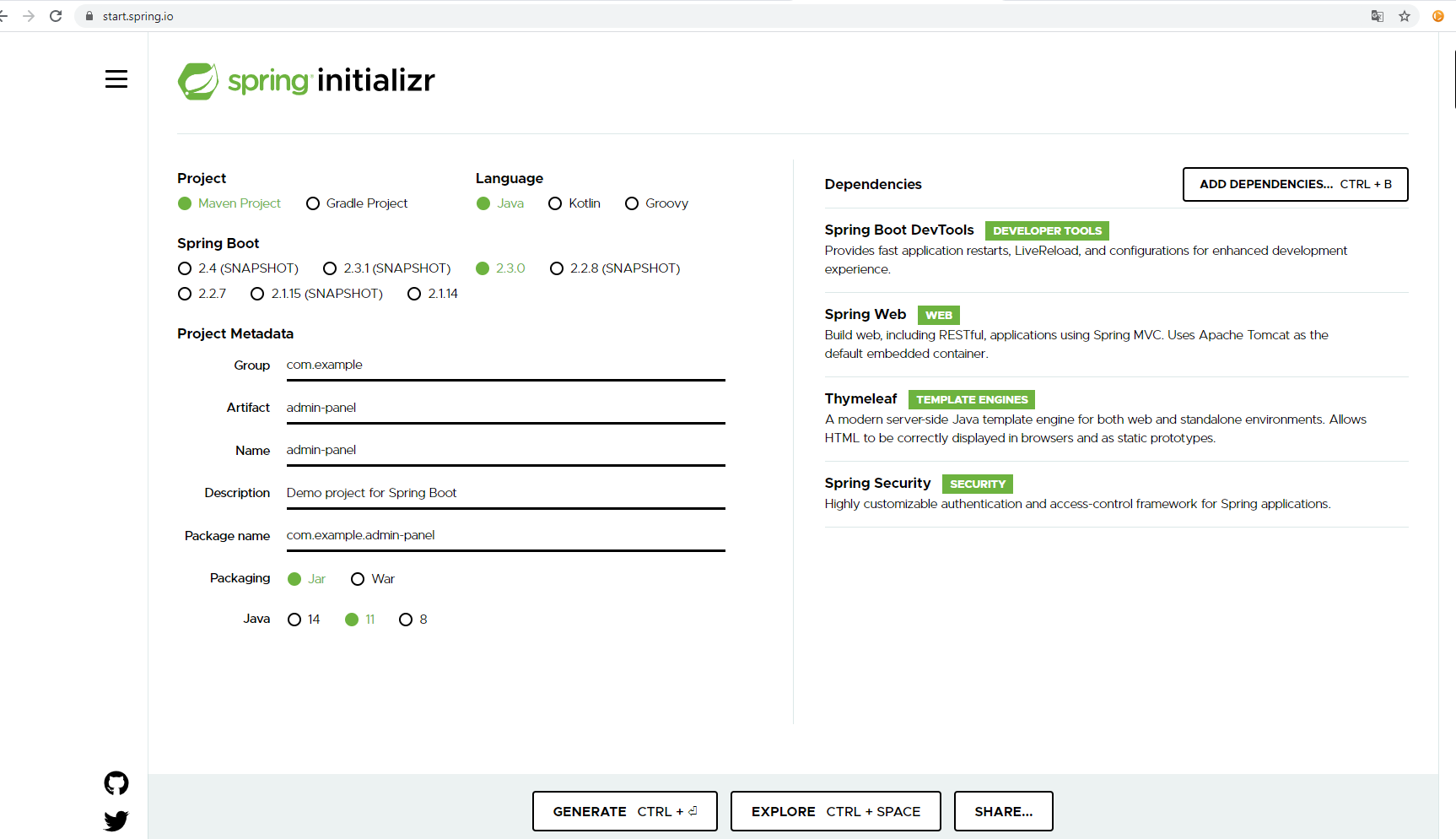
После скачивания и установки можно непосредственно приступать к разработке программ, все необходимые библиотеки и инструменты доступны “из коробки”, ничего дополнительного скачивать и устанавливать не нужно.

**Spring Framework** (или коротко Spring) — универсальный фреймворк с открытым исходным кодом для Java-платформы.

Spring стал широко распространённым в Java-сообществе главным образом как альтернатива и замена модели Enterprise JavaBeans. Spring предоставляет бо́льшую свободу Java-разработчикам в проектировании; кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба.

Между тем, особенности ядра Spring применимы в любом Java-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на Java Enterprise платформе. По этим причинам Spring приобрёл большую популярность и признаётся разработчиками как стратегически важный фреймворк.

Для создания проекта на Spring-е необходимо перейти на сайт <https://start.spring.io/> Инициализатор Spring проекта



Здесь необходимо выбрать сборщик проекта (Maven или Gradle)

Apache Maven — фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM, являющемся подмножеством XML. Проект Maven издаётся сообществом Apache Software Foundation, где формально является частью Jakarta Project.

Gradle — система автоматической сборки, построенная на принципах Apache Ant и Apache Maven, но предоставляющая DSL на языках Groovy и Kotlin вместо традиционной XML-образной формы представления конфигурации проекта.

В качестве сборщика был выбран Maven, тк он используется в большинстве проектов, он является более стабильным и имеет понятный интерфейс, кроме того, в большинстве документаций по спрингу описан именно сборщик Maven.

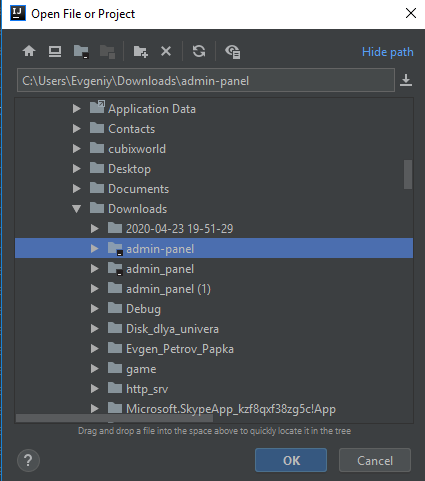
Далее необходимо выбрать название проекта, название артефакта, название пакета и т.д (никаких логических функций это не несет, просто названия, которые нужны лишь для идентифицирования проекта)

После выбора названий, идет выбор упаковки приложения в Jar или War, отличия заключаются в том, что jar -–это так называемый “толстый клиент” т.е проект компилируется вместе со всеми библиотеками, вместе с Tomcat сервером и запускается лишь одной командой (java –jar project.jar) а War – это “тонкий клиент”, в нем содержится лишь необходимые для проекта библиотеки, но чтобы его запустить, необходимо скачать и установить сервер Apache Tomcat и перенести скомпилированный war-файл в директорию сервера.

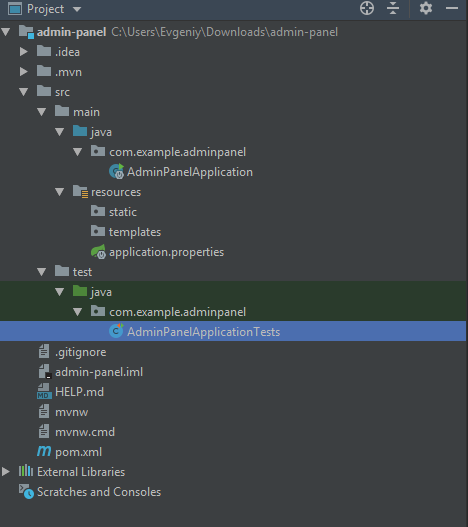
Далее идет выбор версии Java (выбираем 11-ю, т.к она новее 1.8, содержит больше функций чем в более старых версиях, и стабильнее чем последние версии, например 14)

В правой части сайта идет выбор подключаемых модулей. Для данного проекта необходимо выбрать основные модули, такие как *Spring Boot Starter* – содержит большинство основных библиотек, классов и функций Spring-а. *Spring Web –* в нем содержатся биюлиотеки для создания веб приложений. *Thymeleaf* –шаблонизатор, предназначен для связывания html страниц и бэк-енда, содержит функции, которые переносят контент из бэк-энда на html страницу. *Spring Security* – модуль для безопасности веб-приложений, содержит множество инструментов, фильтров, которые позволяют защитить приложения от большинства атак.

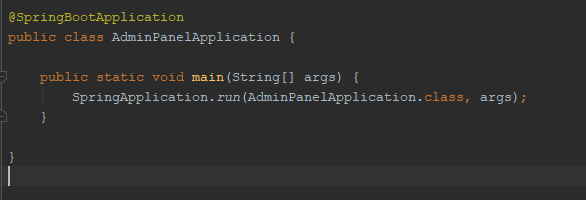
После того как настроили приложение, выбрали все необходимые модули, необходимо его сгенерировать, нажать на кнопку Generate. После нажатия на данную кнопку, скачается zip архив с проектом, разархивируем его и открываем в Intelij Idea. (рисунок ниже)



Листинг содержимого проекта представлен на рисунке ниже

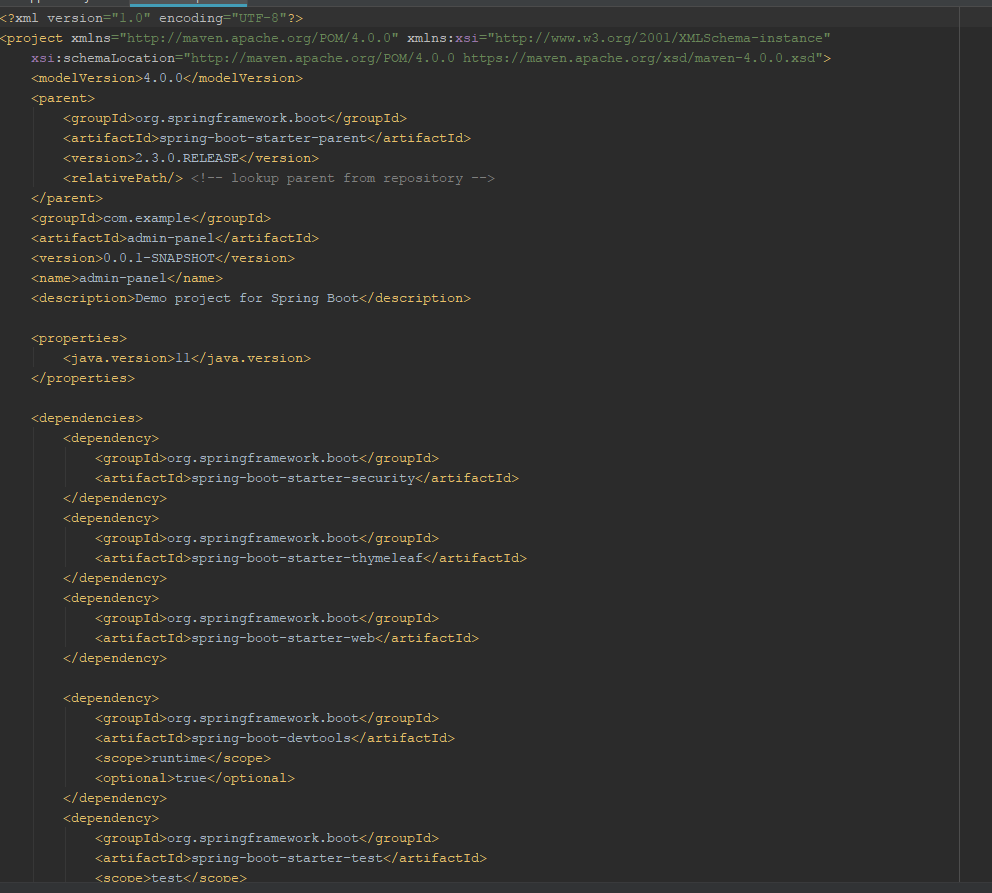


Основные файлы здесь – это AdminPanelApplication – главный класс, точка входа в программу. Класс, содержащий метод main, в котором выполняется запуск приложения (SpringApplication.run())



Далее, файл pom.xml – он содержит в себе настройки сборщика Maven, функции для упаковки приложения, зависимости приложения и другие настройки, необходимые для правильной сборки

Скриншот pom.xml представлен ниже



Файл application.properties содержит конфигурацию спринг приложения, он создан пустым, но далее туда будет добавлены такие параметры как – порт, на котором будет запущен сервер приложения.

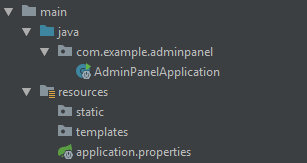
Так же докачаем необходимую библиотеку для загрузки файлов на сервер. Библиотека **Bootstrap File Input**

Ссылка: <https://plugins.krajee.com/file-input#installation>

Скачиваем zip архив, разархивируем и перемещаем папки js и css в папку static проекта

**ВСЕ ЧТО ПРЕДСТАВЛЕНО ВЫШЕ – ЭТО СТАРТОВАЯ УСТАНОВКА – НАСТРОЙКА ПРИЛОЖЕНИЯ, ДАЛЕЕ БУДЕТ ПРЕДСТАВЛЕН САМ ПРОЦЕСС РАЗРАБОТКИ**

Для начала разберемся в структуре проекта.



Все необходимое в процессе разработке будет добавляться в папку main. Как видно на рисунке выше, в main есть два подкаталога – это java и resources.

В java хранится весь исходный код, файлы с расширением .java. Здесь будет описан бэк-энд приложения, классы, методы для управления веб-приложением.

В папке resources хранится весь фронт-энд, т.е html файлы разметки, css, js, медиа файлы и всё, что связано с отображением на html страницах. В resources так же содержится файл конфигурации приложения, и две папки – static и templates.

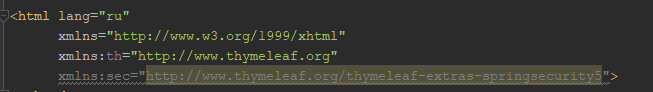
В static хранятся все статичные html страницы, js файлы, css файлы, файлы библиотек , медиа-файлы, и все, что будет храниться без использования шаблонов. А в папке templates будут храниться html страницы, с применением шаблонов thymeleaf

Для начала давайте пропишем в конфигурации проекта порт, на котором будет висеть приложение, т.к стандартные 80 или 8080 могут быть заняты другими приложениями, во избежание конфликтов, пусть порт приложения будет 8888. Для этого в файл application.properties необходимо добавить следующую строчку: server.port=8888. Теперь приложение будет работать на 8888 порту.

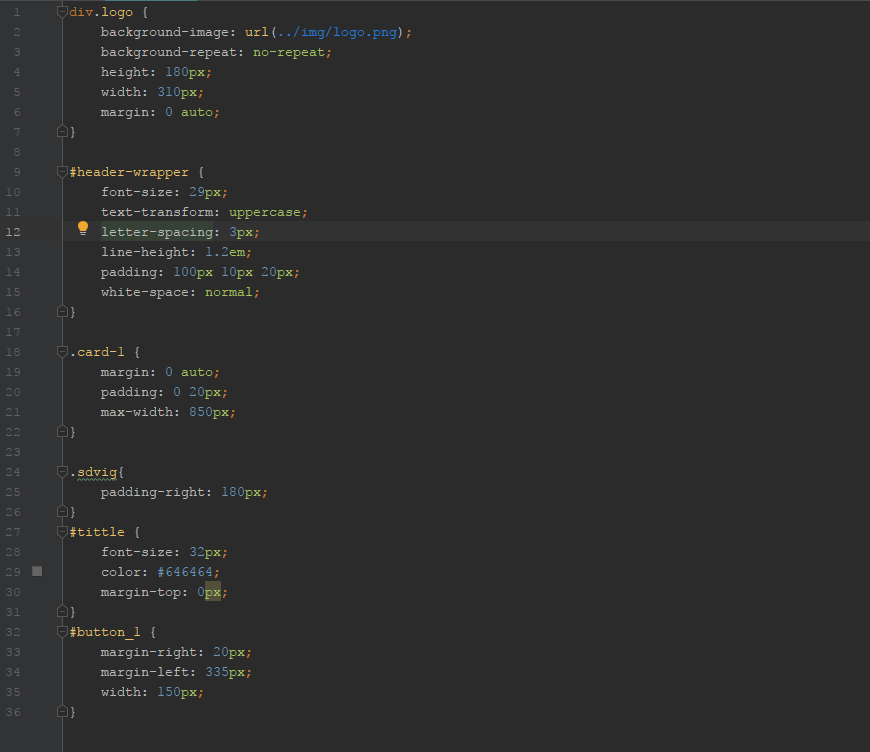
Далее необходимо разобраться с тем, как будет выглядеть страница авторизации пользователей. Она должна состоять из минимум 3-х вещей, это поле для указания логина пользователя, поле для указания пароля, и кнопка “Войти”, при нажатии на которую, будет произведен непосредственно вход в приложение.

Создадим файл login.html, в котором будет разметка страницы логина.

Для начала, подключим thymeleaf шаблонизатор, это стандартное действие и будет применяться для всех последующих html – файлов. Для того, чтобы подключить thymeleaf необходимо в тэге html указать следующие параметры: (рисунок ниже)



Далее, создадим файл таблицы стилей style.css в папке static/css, в котором пропишем стили для страницы логина:



Далее на странице login.html применим все необходимые css стили и js скрипты, а именно, наш style.css, и фреймворк bootstrap (содержащий в себе библиотеки в виде js и css файлов) и jquery, рисунок ниже. Эти действия так же будут применены для всех последующих html страниц

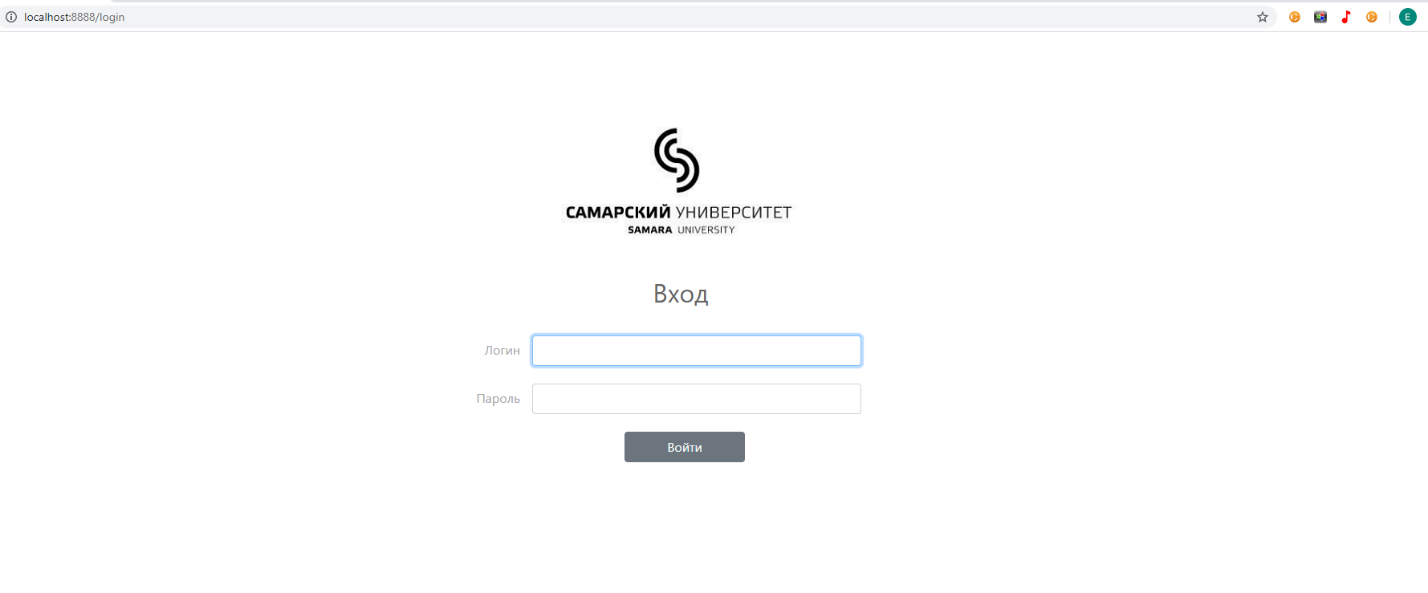


После этого создадим файл logo.png в папке static/img, это будет логотип Самарского университета.

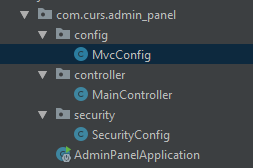
И наконец, основная часть:



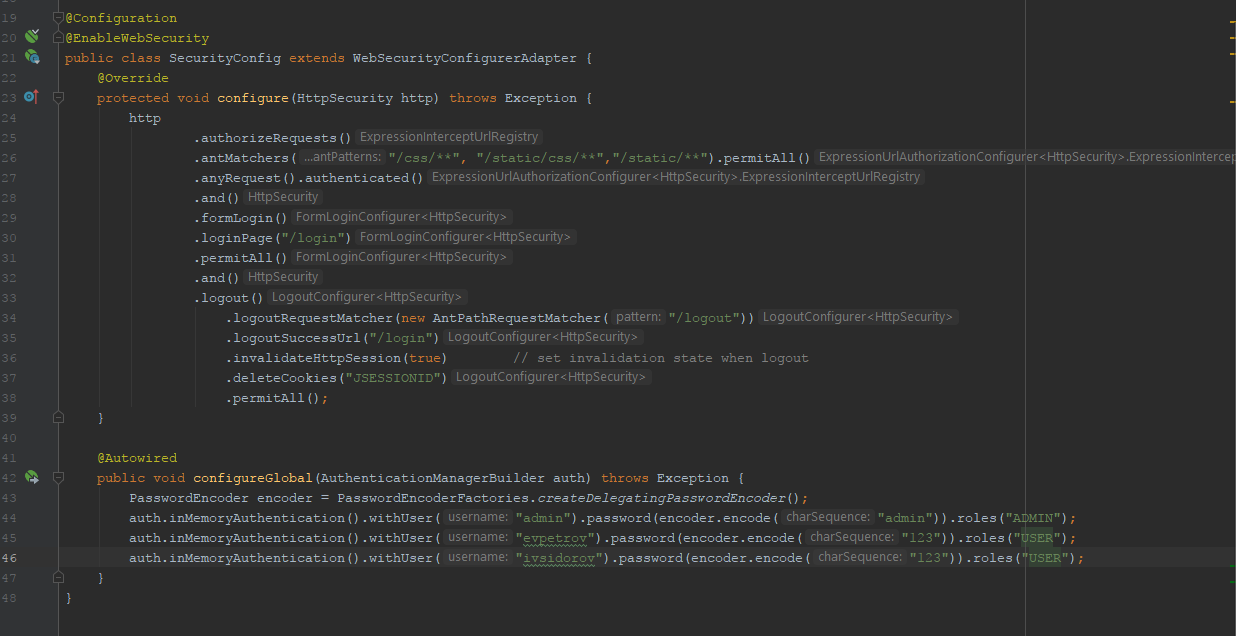
Как видно из рисунка выше, в данной html странице есть форма <form th:action="@{/login}" method="post"> которая будет отсылать post запрос на страницу /login и в качестве параметров указывать username и password, которые пользователь вводит в соответствующих полях. В итоге, наша страница логина выглядит следующим образом:



Но это лишь отображение, разметка, для того, чтобы страница логина выполняла свои функции, необходимо добавить несколько классов в бэк-энде.



Класс SecurityConfig – содержит настройки безопасности, листинг представлен ниже

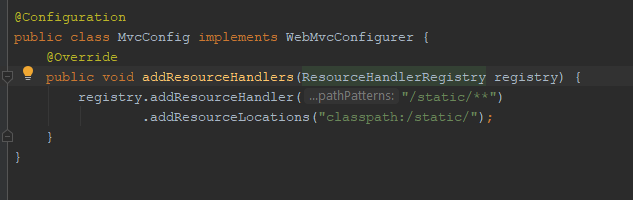


Метод configure отвечает за конфигурацию приложения, в нем описано, какие страницы будут доступны неавторизованным пользователям - .antMatchers("/css/\*\*", "/static/css/\*\*","/static/\*\*").permitAll(). Остальные страницы будут доступны только авторизованным пользователям.

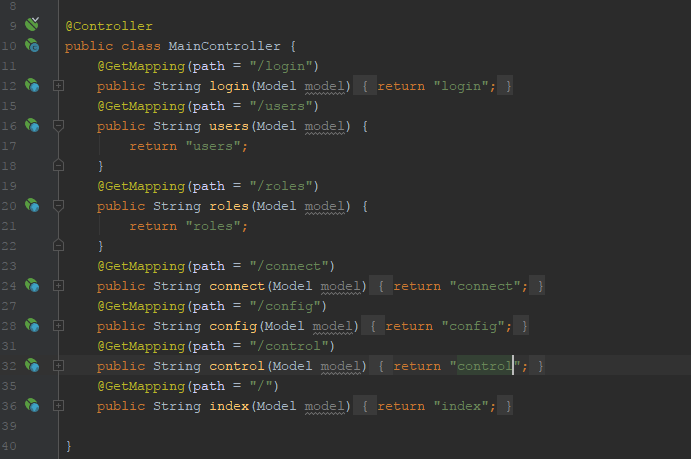
Далее указывается страница для авторизации .loginPage(“/login”).permitAll(), это та страница, на которую будут посылаться post запросы после нажатия кнопки “Войти”, обрабатывается автоматически Spring Security модулем. Так же указываются действия, которые будут произведены при logout-е (.logout()) а именно, страница, на которую должен быть послан запрос, чтобы выйти из пользователя, страница, на которую пользователь будет перемещен после выхода (.logoutSuccesUrl(“/login”)) и удаление cookies с идентификатором “JSESSIONID”

Второй метод configureGlobal добавляет стандартный кодировщик паролей (по умолчанию пароли будут храниться как хэш-суммы MD5) и добавляет трех пользователей с паролями auth.inMemoryAunthetication(…)

Далее файл MVCconfig – в нем указано, в какой папке будут храниться ресурсы приложения, т.е чтобы Spring видел где лежат статичные файлы, в папке /static



И самый важный файл – это файл контроллера MainController. Он содержит все обработчики, и действия, которые должны выполняться при переходе пользователя на те или иные URL-адреса.

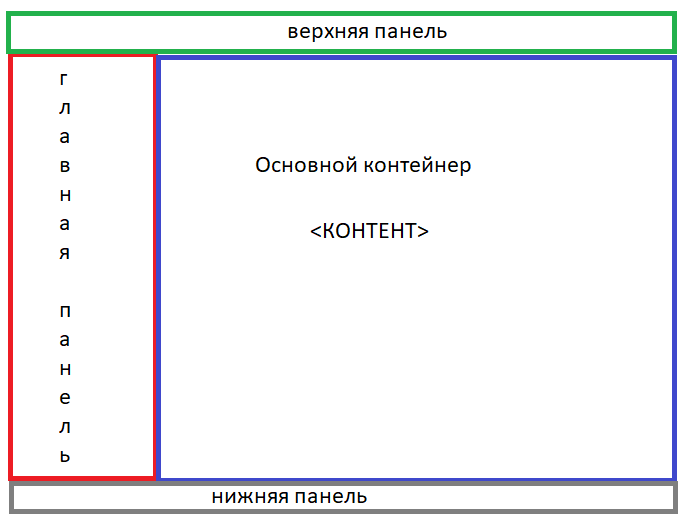


В нем заранее укажем обработку на все страницы, которые будут доступны пользователям.

@GetMapping(path = …) - это аннотация отвечает за обработку Get – запросов к url, значению параметра path, далее идет публичный метод, который возвращает String значение, return “login” – это значение из папки resources/templates/login.html, оно должно совпадать с существующим шаблоном html страницы т.е если возвращаем “login” то пользователь, перейдя по соответствующему url увидит страницу login.html

**РАЗРАБОТКА ОСНОВНОГО ИНТЕРФЕЙСА СЕТЕВОГО УСТРОЙСТВА**

Интерфейс схематично можно представить следующим образом:



Как видно из рисунка, интерфейс разделен на 4 части.

**Верхняя панель** – здесь будет надпись “Панель управления” и некоторые декоративные детали.

**Главная панель** – это панель, в которой будут располагаться все вкладки приложения, это навигационная панель, для перехода на соответствующие страницы сайта.

**Нижняя панель** – тут будут данные об разработчике

**Основной контейнер**– здесь будет находиться весь контент, т.е это то место, куда будут подставляться те или иные функции для управления устройством.

Определим вкладки, которые будут доступны в главной панели:

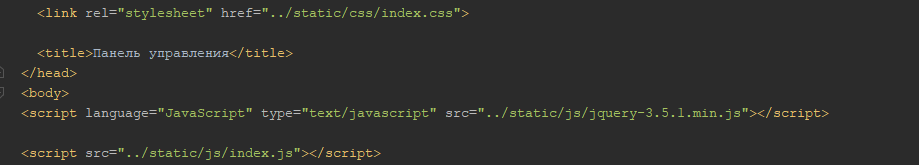
* Главная страница – это приветственная страница, на которую будет переслан пользователь, после авторизации
* Страница информации – здесь будет краткая информация по сетевому устройству
* Настройки – здесь будут страницы, которые отвечают за настройку сетевого устройства
* Файлы – отображение файлов конфигурации, загруженных на устройство
* Администрирование – здесь будет управление пользователями и группами, а так же кнопка отката до заводских настроек.
* Выход – страница logout

Определим роли пользователей в нашей системе

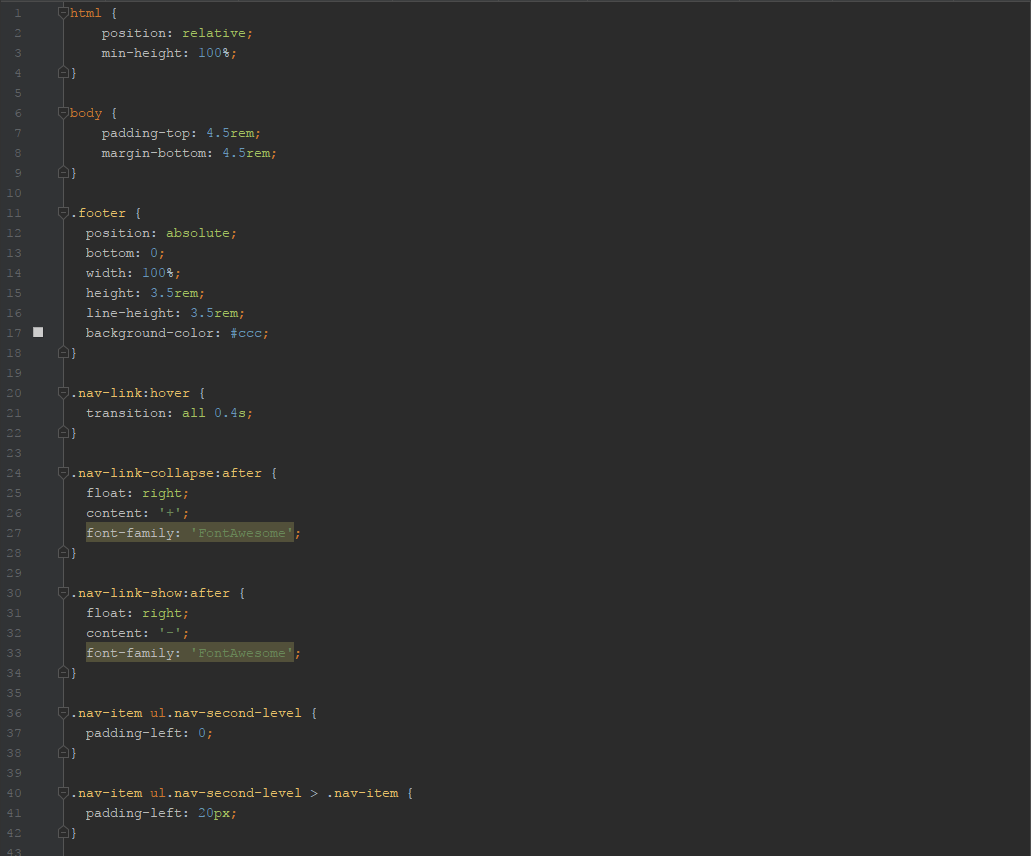
* **Администратор** – доступны все возможности системы, главная роль.
* **Пользователь** – доступны вкладки – Главная страница, Страница информации, настройки, Файлы и Выход (запрещена страница Администрирование)
* **Аноним** – доступны вкладки – Главная страница, Страница информации, Выход

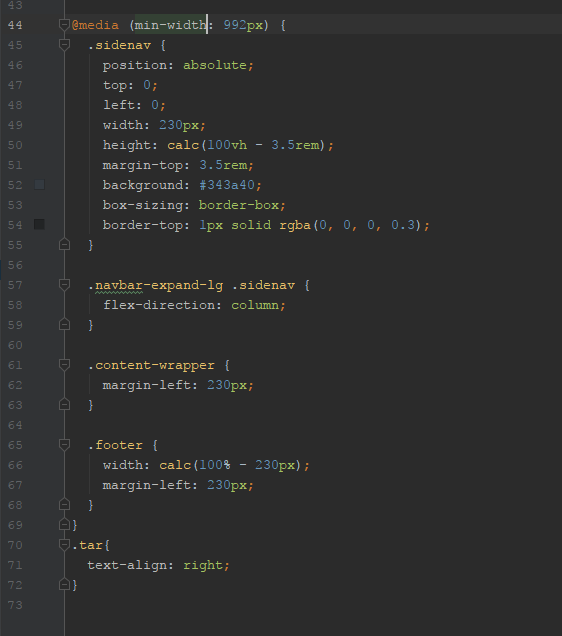
Начнем с разработки главной страницы веб приложения – index.html

Выполним стандартные действия, которые были описаны выше (для страницы login.html) а так же добавим еще таблицу стилей index.css и скрипт, отвечающий за отображение вкладок в главной панели index.js

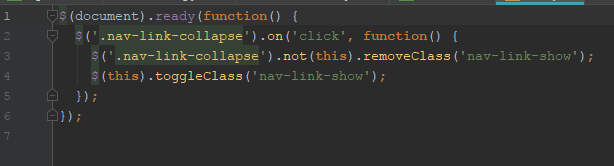


Листинг файла index.css:

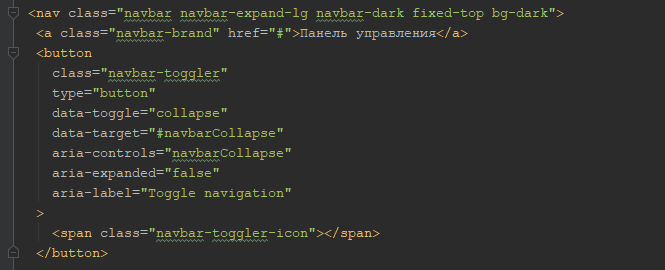




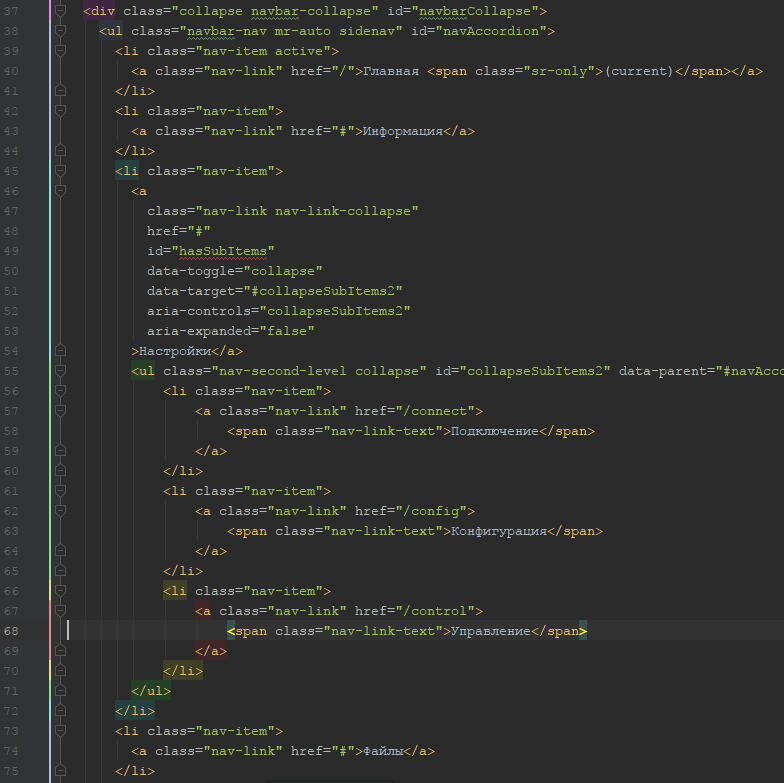
Листинг файла index.js:

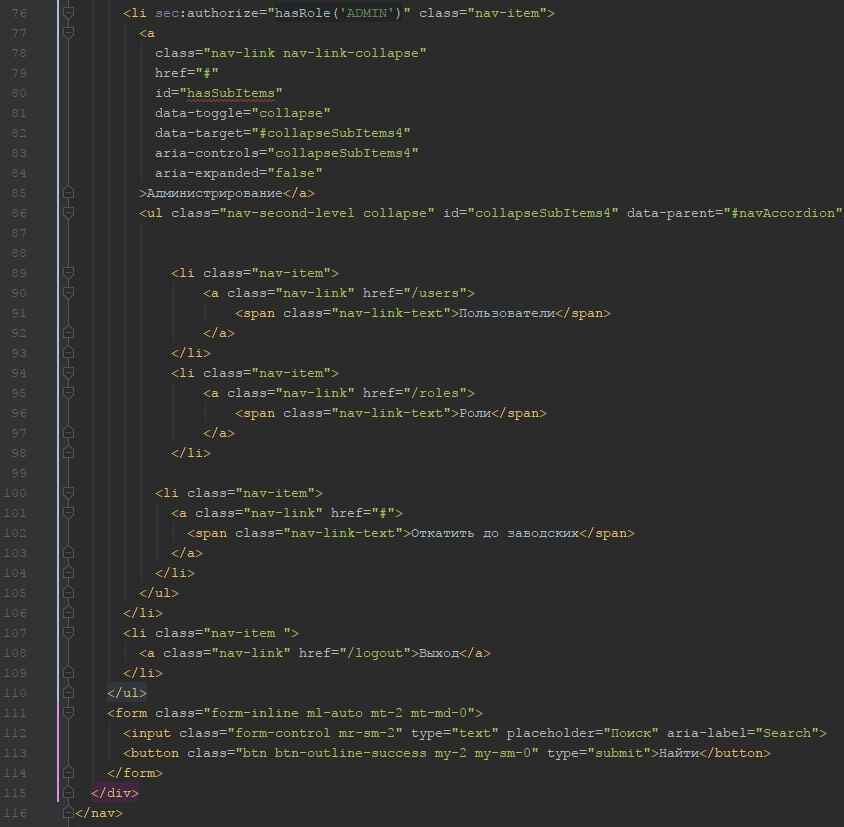


Далее опишем навигационную панель, она будет общая для всех страниц. Итак, **верхняя панель:**



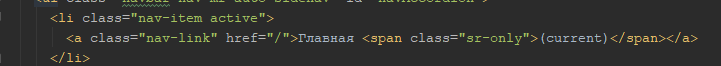
**Главная панель:**





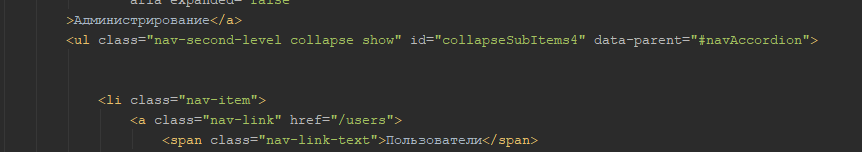
Здесь рассмотрим подробнее, в чем будет отличие у различных страниц.

Index.html – это главная страница, поэтому в навигационной панели – она должна быть выделена, для этого необходимо в тэге <li> добавить класс active



Сама панель состоит из списка ссылок. Ссылок на конкретные страницы например /users / roles /index /logout и другие.

Чтобы открыть подсписки, например во вкладке Администрирование – есть так же вкладки Пользователи, Роли и Откатить до заводских, необходимо добавить класс show в тэг <li> как на рисунке ниже

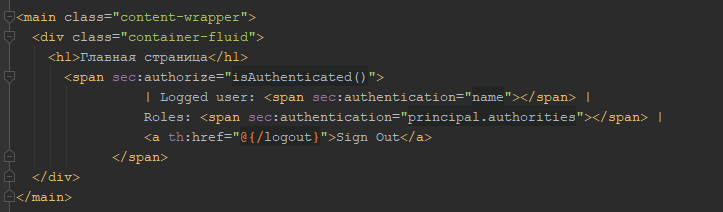


Еще обратим внимание на то, что некоторые вкладки будут доступны только определенной группе пользователей, например вкладка Администратор доступна только пользователям, с группой ADMIN. Для этого в тэг <li> необходимо добавить следующую строку



Теперь этот список будет досупен только пользователям с ролью Администратор. Аналогично можно сделать для пользователей USER или гостей ANNONIMUS.

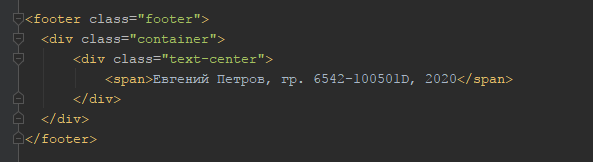
**Основной контейнер:**



Распологается под тэгом <main>

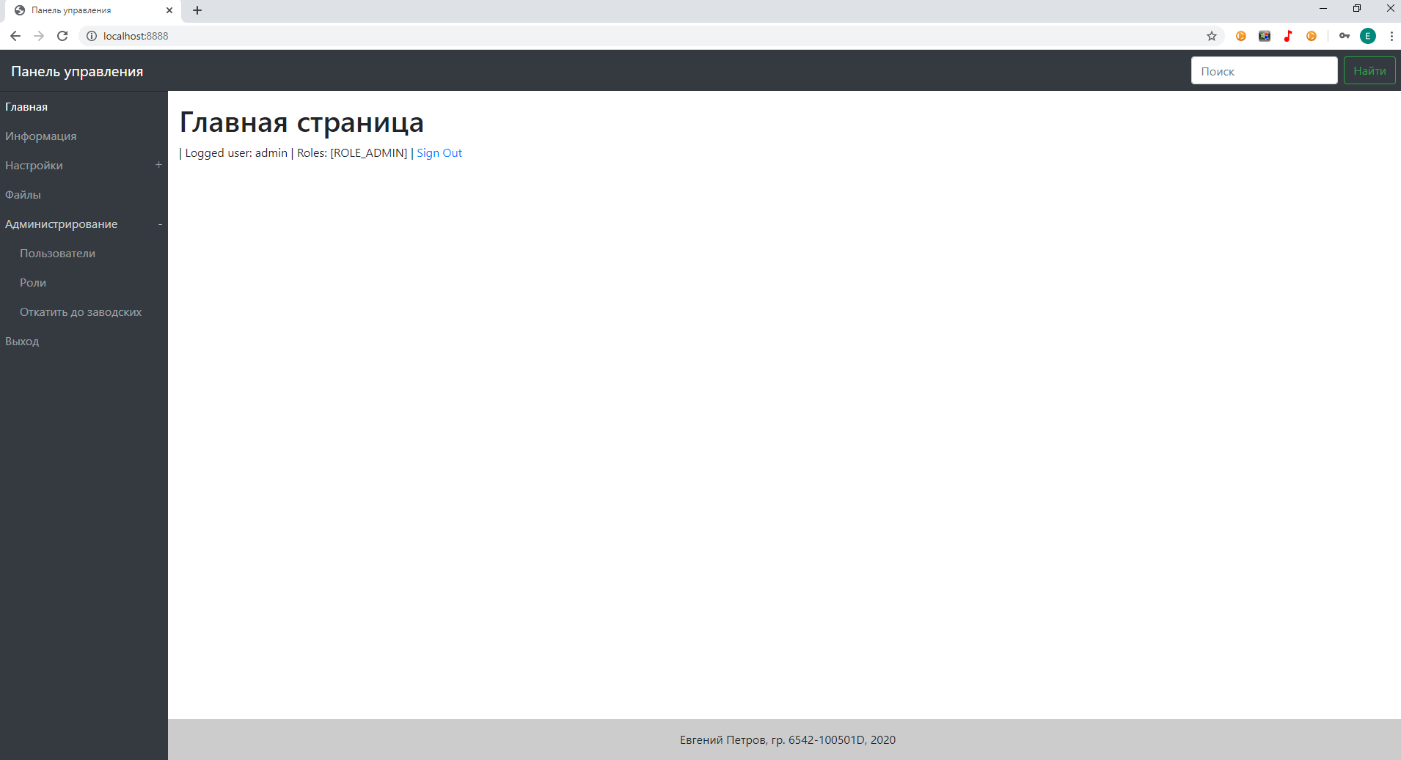
Как видно на рисунке выше, в нем отображаются данные о пользователе – а именно – логин пользователя, роли пользователя и кнопка выхода

**Нижняя панель:**

****

Распологается под тэгом <footer>

В итоге наш index.html выглядит следующим образом:



Все последующие страницы будут различаться лишь *основным контейнером* (тэг main) и выделением соответствующих вкладок в *главной панели*

Ниже будут приведены листинги html страниц, а конкретнее – их основной части main

**Users.html**

<main class="content-wrapper">  
 <div class="container-fluid">  
  
 <div th:if="${(param.add != null) and (param.add[0] == 'true')}">  
 <h1 style="margin-left: 80px;">Добавить пользователя</h1>  
  
 <div class="my-4" style="max-width: 600px;">  
 <form id="user\_form">  
 <div class="form-group row">  
 <label for="name" class="tar col-sm-2 col-form-label">Имя</label>  
 <div class="col-sm-10">  
 <input type="text" class="form-control" id="name" name="name">  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="lastname" class="tar col-sm-2 col-form-label">Фамилия</label>  
 <div class="col-sm-10">  
 <input type="text" class="form-control" id="lastname" name="lastname">  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="login" class="tar col-sm-2 col-form-label">Username</label>  
 <div class="col-sm-10">  
 <input type="text" class="form-control" id="login" name="login">  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="password" class="tar col-sm-2 col-form-label">Пароль</label>  
 <div class="col-sm-10">  
 <input type="password" class="form-control" id="password" name="password">  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="role" class="tar col-sm-2 col-form-label">Роль</label>  
 <div class="col-sm-10">  
 <select class="form-control" id="role" name="role">  
 <option selected></option>  
 <option>Администратор</option>  
 <option>Пользователь</option>  
 <option>Аноним</option>  
 </select>  
 </div>  
 </div>  
  
  
 </form>  
 <form action="/users" id="back"></form>  
 <div style="margin-left: 190px;" class="my-5">  
 <button form="user\_form" type="submit" class="btn btn-success">Сохранить</button>  
 <button form="back" class="ml-3 btn btn-dark" type="submit">Отменить</button>  
 </div>  
  
 </div>  
  
  
 </div>  
  
 <div th:if="${(param.add == null) or (param.add[0] == 'false')}">  
 <h1>Пользователи</h1>  
 <form action="/users">  
 <input hidden type="text" name="add" value="true">  
 <button class="btn btn-secondary my-2" type="submit">Добавить пользователя</button>  
 </form>  
 <table class="table table-hover">  
 <thead class="thead-dark">  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Имя</th>  
 <th scope="col">Фамилия</th>  
 <th scope="col">Username</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <th scope="row">1</th>  
 <td>Евгений</td>  
 <td>Петров</td>  
 <td>evpetrov</td>  
 </tr>  
 <tr>  
 <th scope="row">2</th>  
 <td>Иван</td>  
 <td>Сидоров</td>  
 <td>ivsidorov</td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
  
 </div>  
</main>

**Roles.html**

<main class="content-wrapper">  
 <div class="container-fluid">  
  
 <div th:if="${(param.add != null) and (param.add[0] == 'true')}">  
 <h1 style="margin-left: 80px;">Добавить роль</h1>  
  
 <div class="my-4" style="max-width: 600px;">  
 <form id="role\_form">  
 <div class="form-group row">  
 <label for="name" class="tar col-sm-3 col-form-label">Наименование</label>  
 <div class="col-sm-9">  
 <input type="text" class="form-control" id="name" name="name">  
 </div>  
 </div>  
 <div class="form-group row">  
 <h4 style="margin-left: 240px;">Разрешения</h4>  
 </div>  
 <div class="form-group">  
 <div class="form-check">  
 <input class="form-check-input" type="checkbox" value="" id="check1">  
 <label class="form-check-label" for="check1">  
 Доступ ко вкладке "Администрирование"  
 </label>  
 </div>  
 <div class="form-check">  
 <input class="form-check-input" type="checkbox" value="" id="check2">  
 <label class="form-check-label" for="check2">  
 Создание пользователей  
 </label>  
 </div>  
 <div class="form-check">  
 <input class="form-check-input" type="checkbox" value="" id="check3">  
 <label class="form-check-label" for="check3">  
 Удаление пользователей  
 </label>  
 </div>  
 <div class="form-check">  
 <input class="form-check-input" type="checkbox" value="" id="check4">  
 <label class="form-check-label" for="check4">  
 Создание ролей  
 </label>  
 </div>  
 <div class="form-check">  
 <input class="form-check-input" type="checkbox" value="" id="check5">  
 <label class="form-check-label" for="check5">  
 Удаление ролей  
 </label>  
 </div>  
 <div class="form-check">  
 <input class="form-check-input" type="checkbox" value="" id="check6">  
 <label class="form-check-label" for="check6">  
 Доступ к файловой системе  
 </label>  
 </div>  
 </div>  
  
  
  
 </form>  
 <form action="/roles" id="back"></form>  
 <div style="margin-left: 190px;" class="my-5">  
 <button form="role\_form" type="submit" class="btn btn-success">Сохранить</button>  
 <button form="back" class="ml-3 btn btn-dark" type="submit">Отменить</button>  
 </div>  
  
 </div>  
  
  
 </div>  
  
 <div th:if="${(param.add == null) or (param.add[0] == 'false')}" style="max-width: 450px;">  
 <h1>Роли</h1>  
 <form action="/roles">  
 <input hidden type="text" name="add" value="true">  
 <button class="btn btn-secondary my-2" type="submit">Добавить роль</button>  
 </form>  
 <table class="table table-hover">  
 <thead class="thead-dark">  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Наименование</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <th scope="row">1</th>  
 <td>Администратор</td>  
 </tr>  
 <tr>  
 <th scope="row">2</th>  
 <td>Пользователь</td>  
 </tr>  
 <tr>  
 <th scope="row">3</th>  
 <td>Аноним</td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
  
 </div>  
</main>

**Connect.html**

<main class="content-wrapper">  
 <h1 style="margin-left: 180px;">Подключение</h1>  
  
 <div>  
 <div class="my-4" style="max-width: 600px;">  
 <form id="user\_form">  
 <div class="form-group row">  
 <label for="ip\_addr" class="tar col-sm-2 col-form-label">IP-адрес</label>  
 <div class="col-sm-10">  
 <input type="text" class="form-control" id="ip\_addr" name="ip\_addr">  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="passw" class="tar col-sm-2 col-form-label">Пароль</label>  
 <div class="col-sm-10">  
 <input type="text" class="form-control" id="passw" name="passw">  
 </div>  
 </div>  
  
  
  
  
 </form>  
 <form action="/users" id="back"></form>  
 <div style="margin-left: 190px;" class="my-5">  
 <button style="width: 110px;" form="user\_form" type="submit" class="btn btn-success">Старт</button>  
 <button style="width: 110px;" form="back" class="ml-3 btn btn-dark" type="submit">Стоп</button>  
 </div>  
  
 </div>  
  
  
 </div>  
  
</main>

**Config.html**

<main class="content-wrapper">  
 <h1 style="margin-left: 300px;">Файлы конфигурации</h1>  
  
 <div>  
 <div class="my-4" style="max-width: 1000px;">  
  
 <div class="form-group row">  
 <label for="file-1" class="tar col-sm-4 col-form-label">Файл конфигурации дерева устройств</label>  
 <div class="col-sm-8">  
 <input id="file-1" name="file1" type="file" class="file add-file" multiple=false>  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="file-2" class="tar col-sm-4 col-form-label">Папка с файлами конфигурации</label>  
 <div class="col-sm-8">  
 <input id="file-2" name="file2" type="file" class="file add-file" multiple=false>  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="file-3" class="tar col-sm-4 col-form-label">Новые конфигурации ОПС</label>  
 <div class="col-sm-8">  
 <input id="file-3" name="file3" type="file" class="file add-file" multiple=false>  
 </div>  
 </div>  
 <div class="form-group row">  
 <label for="file-4" class="tar col-sm-4 col-form-label">Новые конфигурации ОПС</label>  
 <div class="col-sm-8">  
 <input id="file-4" name="file4" type="file" class="file add-file" multiple=false>  
 </div>  
 </div>  
 <script>  
 $(".add-file").fileinput({  
 language: "ru",  
 dropZoneEnabled: **false**,  
 maxFileCount: 1,  
 showPreview: **false** });  
 </script>  
  
 </div>  
  
  
 </div>  
  
</main>

**Control.html**

<main class="content-wrapper">  
 <h1 style="text-align: center;">Управление</h1>  
  
 <div class="">  
 <div class="row">  
 <div class="col-sm">  
 <div class="my-4 mx-4" style="max-width: 600px;">  
 <div class="accordion" id="accordionExample">  
  
  
 <div class="card">  
 <div class="card-header" id="heading1">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse1" aria-expanded="true" aria-controls="collapse1">  
 Двери  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse1" class="collapse show" aria-labelledby="heading1" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div>  
  
 </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading2">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse2" aria-expanded="true" aria-controls="collapse2">  
 Турникеты  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse2" class="collapse show" aria-labelledby="heading2" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading3">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse3" aria-expanded="true" aria-controls="collapse3">  
 Ворота  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse3" class="collapse show" aria-labelledby="heading3" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading4">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse4" aria-expanded="true" aria-controls="collapse4">  
 Считыватели  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse4" class="collapse show" aria-labelledby="heading4" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading5">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse5" aria-expanded="true" aria-controls="collapse5">  
 Выходы  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse5" class="collapse show" aria-labelledby="heading5" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading6">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse6" aria-expanded="true" aria-controls="collapse6">  
 Зоны ОПС  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse6" class="collapse show" aria-labelledby="heading6" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading7">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse7" aria-expanded="true" aria-controls="collapse7">  
 Разделы ОПС  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse7" class="collapse show" aria-labelledby="heading7" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
 <div class="card">  
 <div class="card-header" id="heading8">  
 <h5 class="mb-0">  
 <button style="width: 280px;" class="btn btn-secondary" type="button" data-toggle="collapse" data-target="#collapse8" aria-expanded="true" aria-controls="collapse8">  
 Контроллеры  
 </button>  
 </h5>  
 </div>  
  
 <div id="collapse8" class="collapse show" aria-labelledby="heading8" data-parent="#accordionExample">  
 <div class="card-body">  
 <div class="list-group">  
 <button type="button" class="list-group-item list-group-item-action">Объект 1</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 2</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 3</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 4</button>  
 <button type="button" class="list-group-item list-group-item-action">Объект 5</button>  
 </div> </div>  
 </div>  
 </div>  
  
  
 </div>  
 </div>  
  
  
 </div>  
 <div class="col-sm">  
 <div class="my-4 mx-4" style="max-width: 600px; height: 700px; border-style: groove;">  
 Контент  
 </div>  
  
 </div>  
 </div>  
  
  
 </div>  
  
</main>

Дерево файлов проекта:

