**МИНОБРНАУКИ РОССИИ**

**САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

**ОТЧЕТ**

**по лабораторным работам №5**

**по дисциплине «Программирование интернет-приложений на языке Java»**

**Тема: «Аутентификация и авторизация пользователей Web-приложений»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 7308 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Фомченкова Н.А. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Павловский М.Г. |

Санкт–Петербург

2021

**Цель работы**

Знакомство со способами реализации аутентификации и авторизации пользователей Web-приложения.

**Задание**

* Настройте Web-сервер на базовую аутентификацию.
* Запустите приложение, разработанное в лабораторной работе № 3, с указанием имени пользователя и пароля.
* Настройте Web-сервер на протокол SSL и запустите приложение, используя защищенный порт.

**Ход работы**

Настроим базовую аутентификацию.

Добавим в файл tomcat-user.xml пользователей. Определим их роли и зададим пароли.

<role rolename=" manager "/>

<role rolename=" admin "/>

<role rolename=" tomcat "/>

<role rolename="manager-gui"/>

<user username="tomcat" password="123456" roles="manager-gui"/>

<user username="admin" password="123456" roles="manager, admin " />

<user username="myname" password="123456" roles="tomcat" />

Далее вносим изменения в файл web.xml. Указываем способ аутентификации, защищаемые ресурсы, имена ролей, которые позволят обращаться к этим ресурсам.

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"

version="4.0">

<welcome-file-list>

<welcome-file>MedicineList.jsp</welcome-file>

</welcome-file-list>

<!-- Описание защищаемых ресурсов -->

<security-constraint>

<web-resource-collection>

<web-resource-name> Список врачей </web-resource-name>

<url-pattern> /MedicineList.jsp </url-pattern>

<http-method> GET </http-method>

<http-method> POST </http-method>

</web-resource-collection>

<auth-constraint>

<role-name> admin</role-name>

<role-name> tomcat </role-name>

</auth-constraint>

</security-constraint>

<!-- Определение вида аутентификации -->

<login-config>

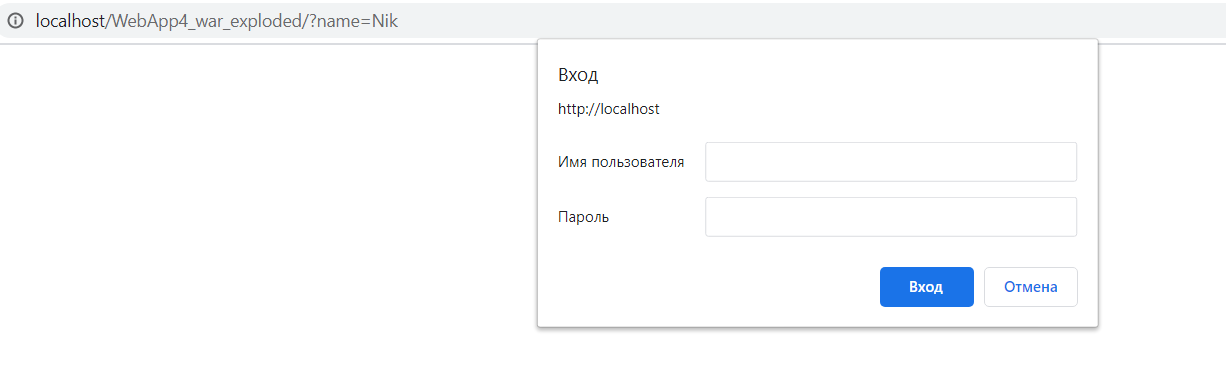
<auth-method>BASIC</auth-method>

<realm-name> Write Medicine List </realm-name>

</login-config>

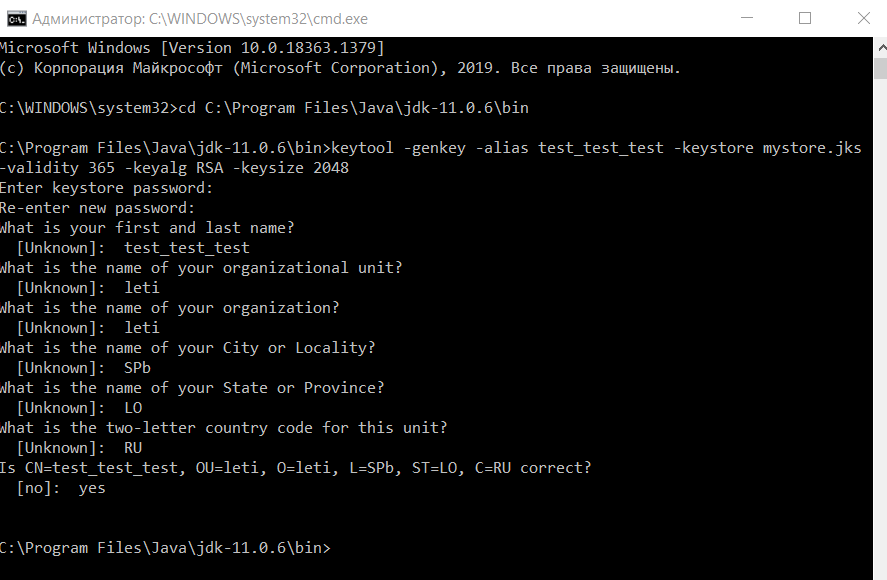
</web-app>

Далее запустим программу. У нас просят ввести логин и пароль. Вводим, все работает.

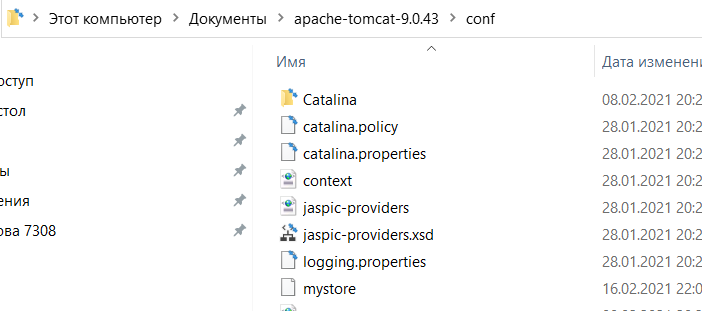


Настройка SSL-протокола.

Создадим ключ и хранилище.



Поместим его в корневую директорию tomcat.



Внесем изменения в файл server.xml.

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" SSLEnabled="true"

compression= "on" scheme="https" secure="true"

keystoreFile="conf/mystore.jks"

keystorePass="123456789"

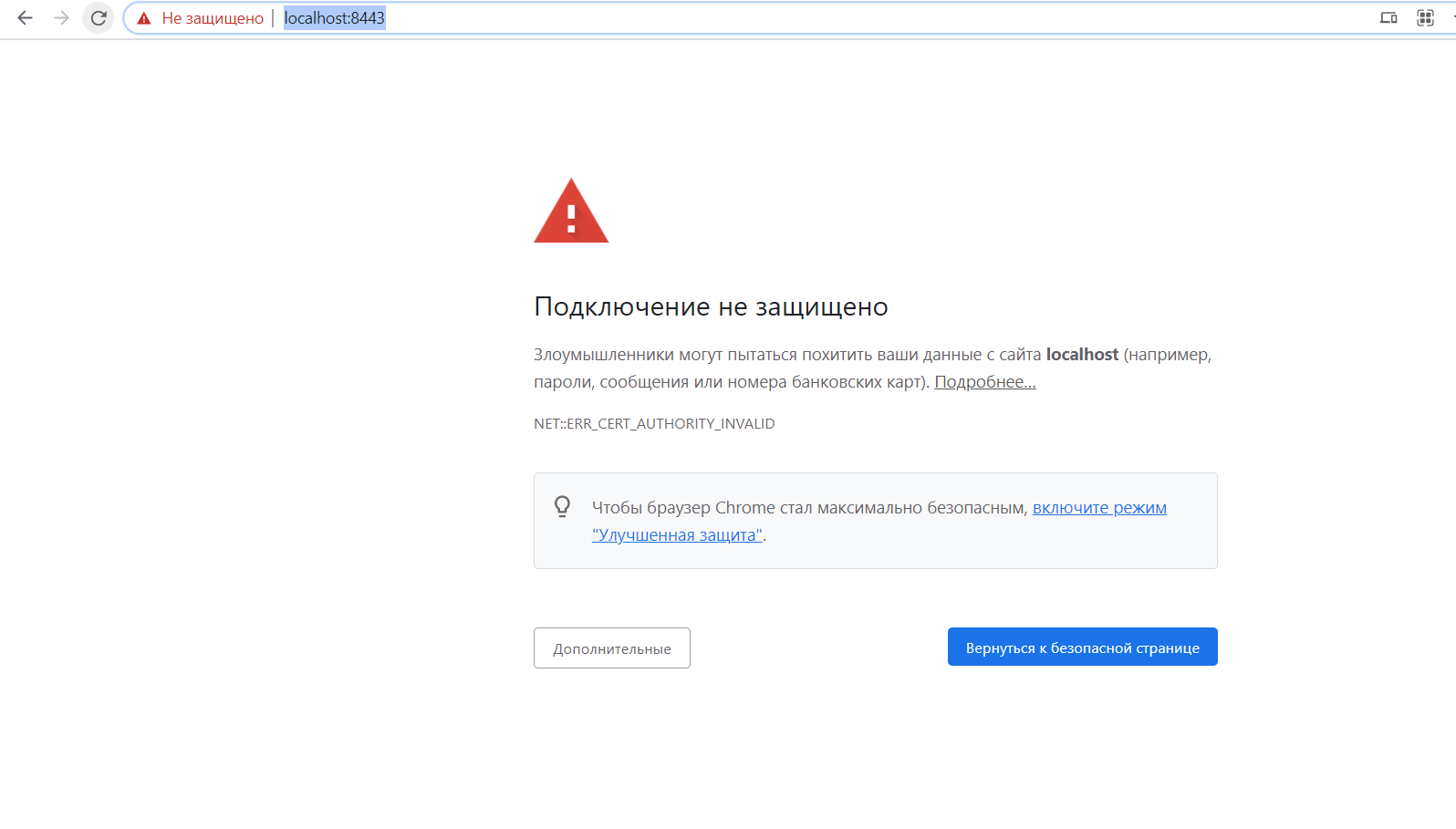
SSLVerifyClient="none"

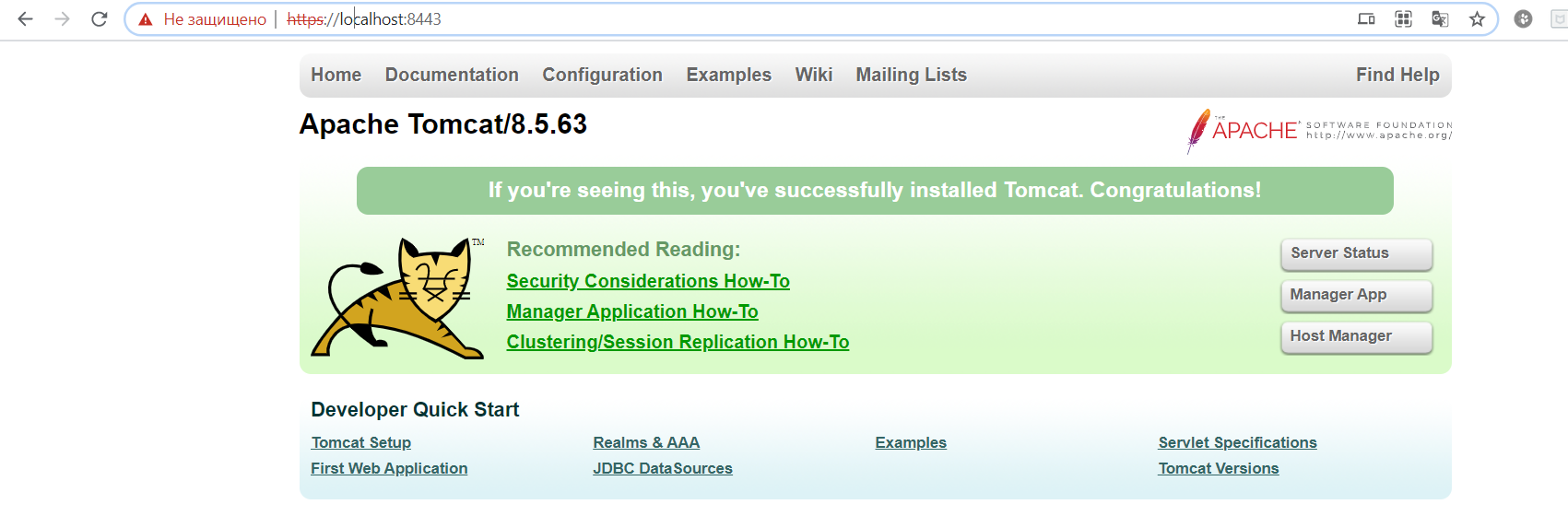
sslProtocol="TLS"

useBodyEncodingForURI="true"

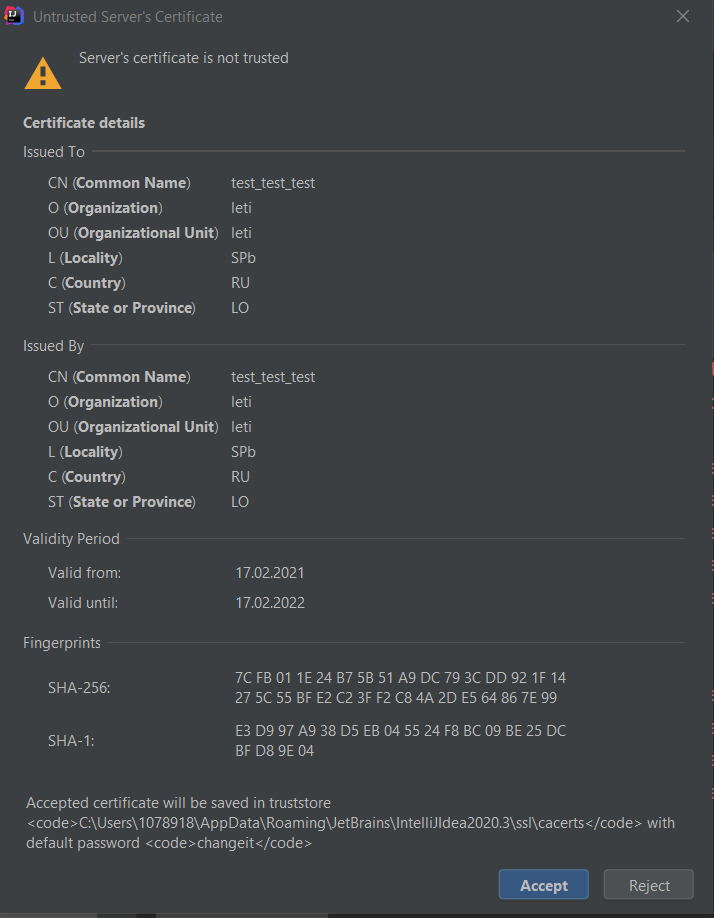
/>

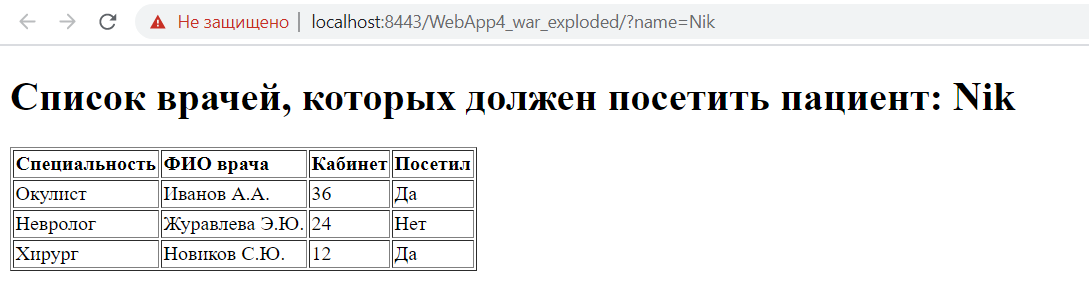
Запустим сервер и смиримся с рисками. Все работает.

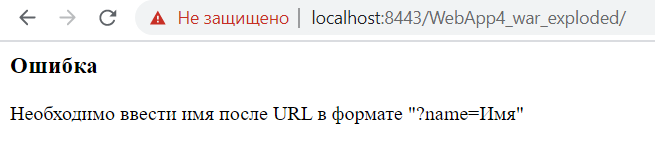




Запустим наше приложение. Все работает.







**Вывод:** в данной л/р я узнала, несколько способов реализации аутентификации и авторизации пользователя в Web-приложении, написанном на языке Java. Первые два пункта было достаточно легко выполнить, в то время как создание сертификата было достаточно проблемной задачей. Но в конечном итоге поставленная задача была решена.