

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

Практический пример №6
Spring Security. Простейшая форма аутентификации.

Составил: Старший преподаватель кафедры «Информатика»
Черниговский Алексей Сергеевич

Красноярск 2020

Содержание

1 Создание Maven проекта.....	3
2 Зависимости.....	3
3 Структура проекта.....	4
4 Контроллер главной страницы.....	5
5 security.xml конфигурация.....	5
6 dispatcher.xml конфигурация.....	6
7 web.xml конфигурация.....	7
8 home.ftl страница.....	7
9 privatePage.ftl страница.....	8
10 Описание.....	8

1 Создание Maven проекта

Создаем новый Maven проект в IDE Eclipse из архетипа webapp

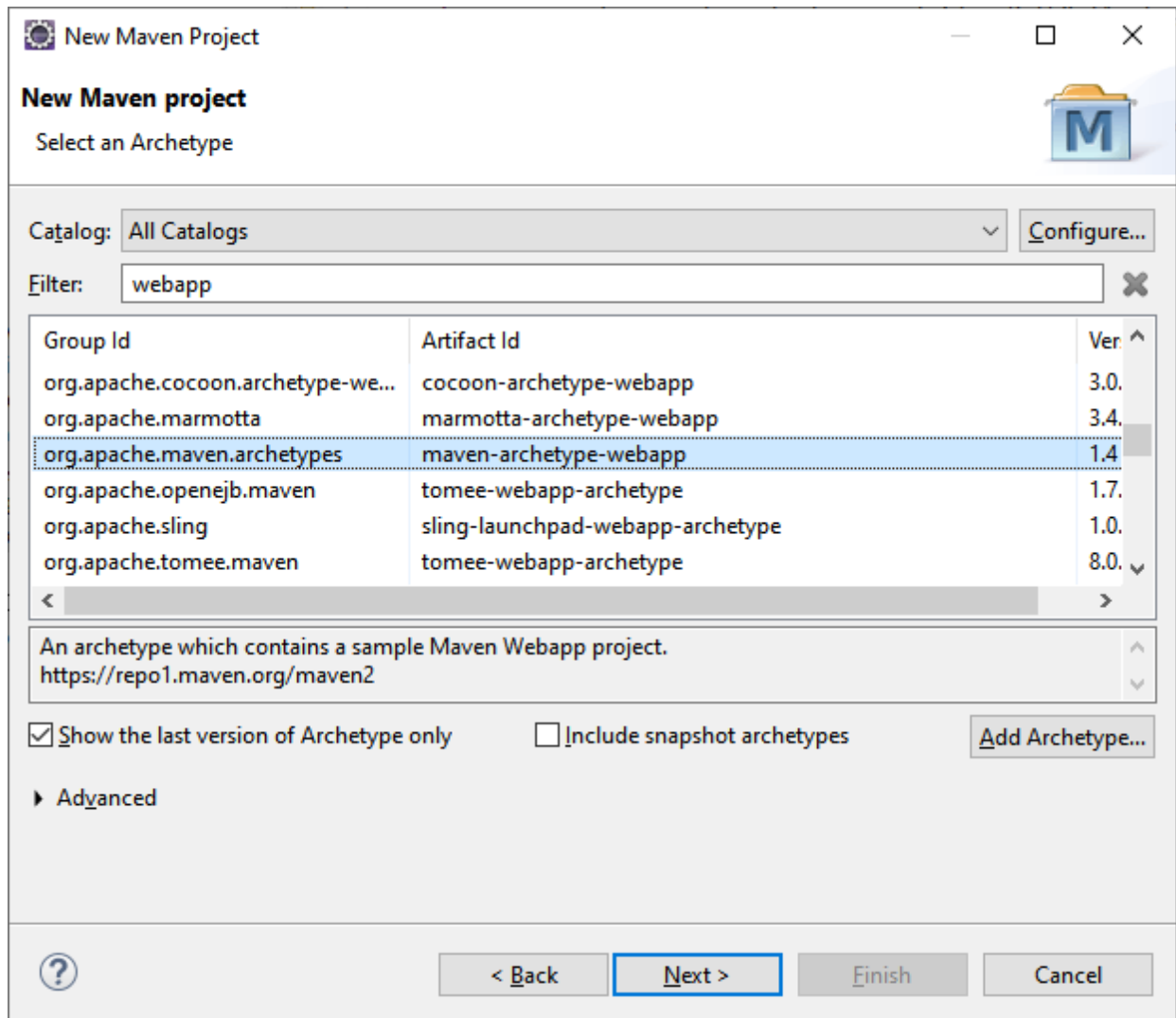


Рисунок 1 — webapp artifact

Зададим ему Artifact Id: security.

2 Зависимости

Добавим в pom.xml зависимости Spring Core, Spring Context, Spring Web, Spring Web MVC, Spring Data JPA, Apache Freemarker (можно было сделать и на jsp, но воспользуемся наработками прошлого примера), Spring Context Support, Spring Security, Spring Security Config и Spring Security Web и некоторые другие, которые будут необходимы позже. В случае если каких-либо зависимостей не хватает и вы их не можете найти — обратитесь к преподавателю.

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>5.2.8.RELEASE</version>
</dependency>
```

```

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>5.2.8.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>5.2.8.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>5.2.8.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.freemarker</groupId>
  <artifactId>freemarker</artifactId>
  <version>2.3.30</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>5.2.8.RELEASE</version>
</dependency>

dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-config</artifactId>
  <version>5.2.7.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-core</artifactId>
  <version>5.2.7.RELEASE</version>
</dependency>

```

3 Структура проекта

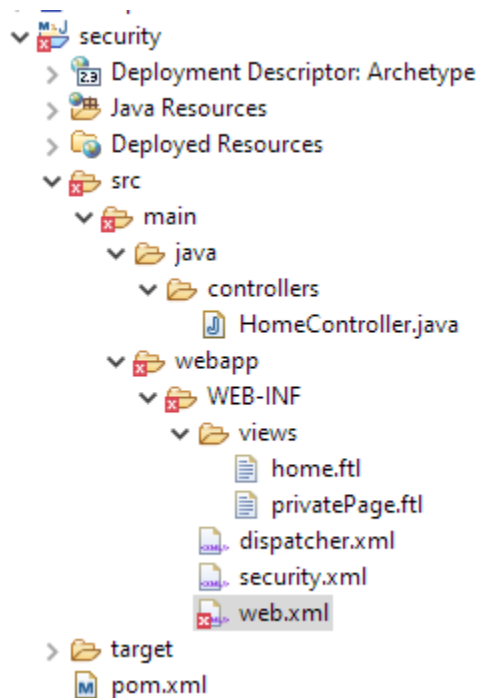


Рисунок 2 — структура проекта

4 Контроллер главной страницы

package controllers;

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```

```
@Controller
```

```
public class HomeController {
```

```
    @RequestMapping(value="/", method=RequestMethod.GET)
```

```
    public String home(){
        return "home";
    }
}
```

```
    @RequestMapping(value="/admin", method=RequestMethod.GET)
```

```
    public String privateHome() {
        return "privatePage";
    }
}
```

5 security.xml конфигурация

```
<beans:beans xmlns="http://www.springframework.org/schema/security"
xmlns:beans="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
```

```

        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/security
        http://www.springframework.org/schema/security/spring-security-
5.2.xsd">

```

```

        <http auto-config="true" use-expressions="true">
            <intercept-url pattern="/admin" access="hasRole('ROLE_ADMIN')" />
        </http>
        <authentication-manager>
            <authentication-provider>
                <user-service>
                    <user name="admin"
password="$2y$10$dZPGF39gkohmiVwAZQw.muH8W2anLtKwqMmGqWDbQIcdB30qF/NKe"
authorities="ROLE_ADMIN" />
                </user-service>
                <password-encoder hash="bcrypt"/>
            </authentication-provider>
        </authentication-manager>

    </beans:beans>

```

6 dispatcher.xml конфигурация

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
    <mvc:annotation-driven />
    <context:component-scan base-package="controllers"/>

    <mvc:annotation-driven/>

    <bean id="freeMarkerConfig"
class="org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer">
        <property name="templateLoaderPath" value="/WEB-INF/views"/>
        <property name="defaultEncoding" value="UTF-8"/>
        <property name="freemarkerSettings">
            <props>
                <prop key="default_encoding">UTF-8</prop>
            </props>
        </property>
    </bean>

    <bean id="templateResolver"
class="org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver">
        <property name="suffix" value=".ftl"/>
        <property name="contentType" value="text/html; charset=utf-8"/>
        <property name="cache" value="false"/>
    </bean>

```

</bean>

</beans>

7 web.xml конфигурация

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >
<web-app>
  <display-name>Archetype Created Web Application</display-name>

  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-class>org.springframework.web.servlet.
      DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>
        /WEB-INF/dispatcher.xml
      </param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
      /WEB-INF/security.xml
    </param-value>
  </context-param>

  <servlet-mapping>
    <servlet-name>dispatcher</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <listener>
    <listener-class>org.springframework.web.context.
      ContextLoaderListener</listener-class>
  </listener>

  <filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-class>
      org.springframework.web.filter.DelegatingFilterProxy
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

8 home.ftl страница

```
<html>
<head>
```

```

        <meta content="text/html; charset=UTF-8">
            <title>Home</title>
    </head>
    <body>
        <h2>Welcome!</h2>
    </body>
</html>

```

9 privatePage.ftl страница

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Admin</title>
</head>
<body>
    Hello Admin
</body>
</html>

```

10 Описание

Приложение представляет собой простейшую авто генерацию страницы аутентификации.

По адресу `/security/login` будет располагаться автоматически сгенерированная страница аутентификации. В приложении настроен (`<user-service>`) один пользователь с правами `ADMIN_ROLE` в файле `security.xml`

Spring Security 5.0 обязывает хранить пароли в захешированном виде, поэтому пароль хранится в виде `bcrypt`-хэша. В Spring по-умолчанию «rounds» установлен 10. (<https://bcrypt-generator.com>) Также в качестве энкодера пролей установлен использующий `bcrypt`-хэш (`<password-encoder hash="bcrypt"/>`).

После того как будут введены верные `login`-пароль пользователю станет доступна страница `/admin` (`intercept-url pattern="/admin" access="hasRole('ROLE_ADMIN')"` `</>`).

Это единственная цель данного примера и приложения — сделать доступной админскую страницу после ввода логина-пароля, который хранится в конфигурации. Об остальном читайте в конспекте лекции.