

Дубенков Семен, 9308

Цель работы: знакомство со способами реализации аутентификации и авторизации пользователей Web-приложения

Настройка базовой аутентификации

Для подключения к серверу используем базовую аутентификацию (через окошко сверху): таким способом легко логиниться, однако для разлогирования требуется перезагрузить браузер

Добавляем в web.xml следующие строки

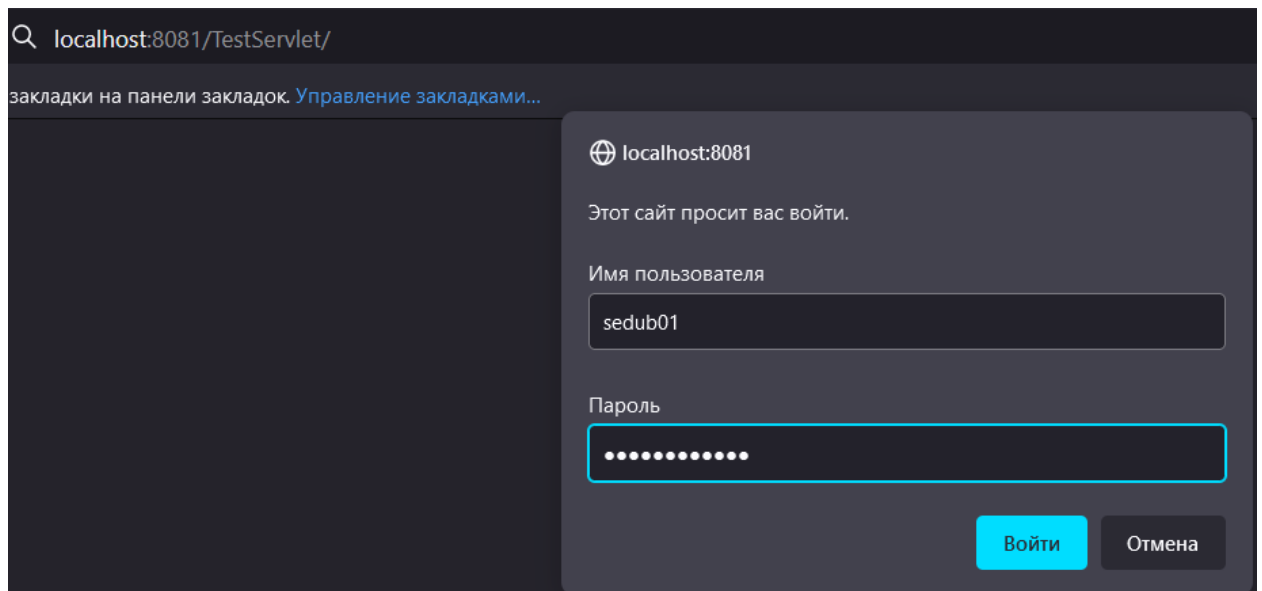
```
<security-role>
  <role-name>admin</role-name>
</security-role>
<security-role>
  <role-name>user</role-name>
</security-role>
<!-- Описание защищаемых ресурсов -->
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Список футболистов</web-resource-name>
    <url-pattern>/TeamTitle.jsp </url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>admin</role-name>
    <role-name>user</role-name>
  </auth-constraint>
</security-constraint>
<!-- Определение вида аутентификации -->
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Write Team List</realm-name>
</login-config>
```

Есть два вида пользователей: админ и обыватель. Пока что им присваиваются одинаковые роли из-за скудности функционала, однако если в auth-constraint убрать тег с user, то вход ему будет запрещен

Пользователей надо добавить в файл tomcat-users.xml, который, в моем случае, находится почему-то в пространстве проектов ебаного eclipse:
C:\Users\sedub01\eclipse-workspace\Servers\Tomcat v10.0 Server at localhost-config (пользователи eclipse поймут)

```
<role rolename="user"/>
<role rolename="admin"/>
<user username="sedub01" password="admin2001!!!" roles="admin"/>
<user username="newbie" password="iamnoob" roles="user"/>
```

Теперь можно авторизоваться



После чего доступ разрешен



Вот мои футболисты

Имя Фамилия	Специализация	Город	Зарплата
Билли Херрингтон	Вратарь	Махачкала	15000
Антон Чехов	Нападающий	Санкт-Петербург	30000
Илья Антонов	Полузащитник	Екатеринбург	25000
Андрей Сачков	Защитник	Вологда	19000

ru ▼ Submit

SSL-авторизация

Для защищенного доступа к странице необходимо создать ключ

```
C:\Program Files (x86)\Apache Software Foundation\Tomcat 10.0\conf>keytool -genkey -alias test2 -keystore mystore2 -validity 365 -keyalg RSA -keysize 2048
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: Semyon Dubenkov
What is the name of your organizational unit?
[Unknown]: FKTI
What is the name of your organization?
[Unknown]: LETI
What is the name of your City or Locality?
[Unknown]: SPb
What is the name of your State or Province?
[Unknown]: Leningradskaya state
What is the two-letter country code for this unit?
[Unknown]: ru
Is CN=Semyon Dubenkov, OU=FKTI, O=LETI, L=SPb, ST=Leningradskaya state, C=ru correct?
[no]: yes
```

Стоит отметить, что мне необходимо было создать ключ размером не менее 1024, иначе браузер выбрасывал ошибку из-за ненадежности соединения (а оно действительно ненадежное, 512 битовый размер лет 10 назад использовали)

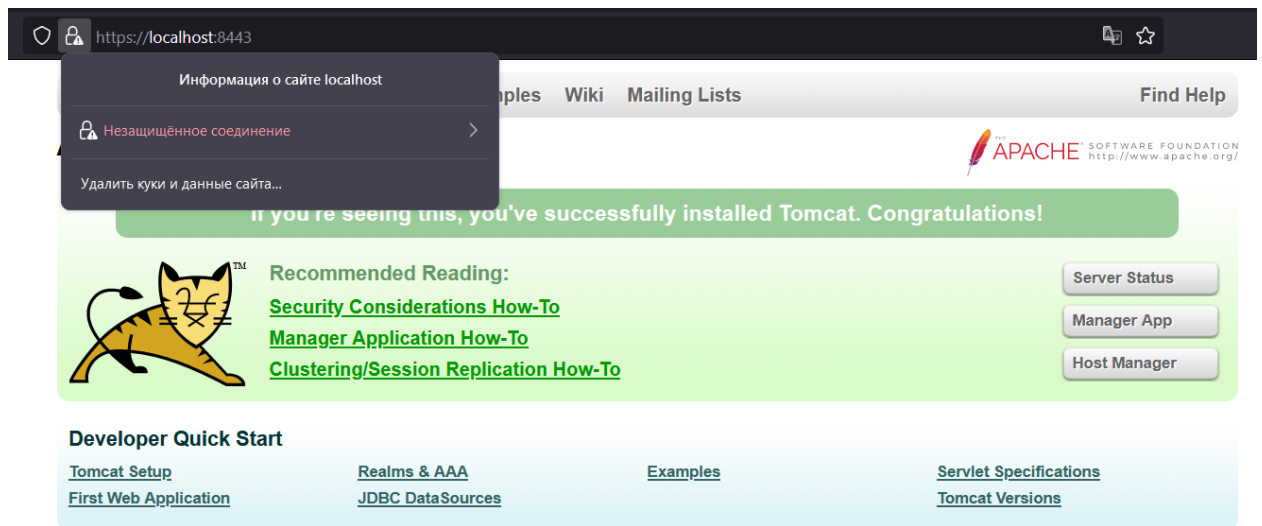
Созданный ключ следовало бы поместить в папку /conf в корневой директории сервера Tomcat и там же отредактировать server.xml, однако так можно протестировать только сервер, но не само приложение

Для приложения следует разместить ключ и измененный server.xml в папку C:\Users\sedub01\eclipse-workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\conf (ну или в другую папку, посмотрите по логам)

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150" SSLEnabled="true">
    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
        <Certificate certificateKeystoreFile="conf/mystore2"
            certificateKeystorePassword="sedub01"
            type="RSA" />
    </SSLHostConfig>
</Connector>
```

Как можно видеть, вставленные строки тоже не из методички, так как Connector org.apache.coyote.http11.Http11Protocol был убран еще в Tomcat 8.5

Еще один нюанс: в переменной JAVA_HOME у меня стоит JDK, в то время как для тестирования защищенного соединения нужно подключать именно JRE, а в eclipse поставить JRE так, чтобы сервер не выебывался, невозможно (в письме Разумовскому Г.В. подробно объяснил почему), так что для демонстрации работы защищенного соединения покажу только сервер Tomcat



Стоит обратить внимание, что соединение обеспечивается по https-протоколу, никак иначе