Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт по лабораторной работе № 5
на тему: “Аутентификация и авторизация пользователей web-приложения”
по дисциплине
“Web-программирование”

Выполнил студент гр.9308:	Яловега Н.В.
Проверил:	Павловский М.Г.

Санкт-Петербург, 2021 г.

Оглавление

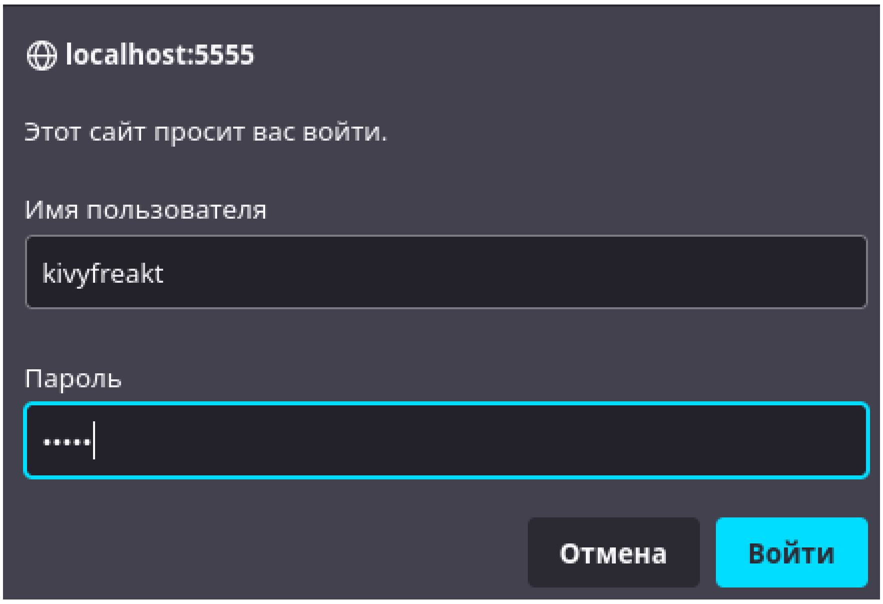
Введение

Целью работы является знакомство со способами реализации аутентификации и авторизации пользователей Web-приложения.

Настройка базовой аутентификации

Для начала необходимо добавить роли и пользователей в файл сервера Tomcat tomcat-users.xml. Далее требуется изменить web.xml проекта. Допустимыми ролями установим только «admin» и «user».

Попробуем авторизироваться за пользователя kivyfreakt группы user:



localhost:5555

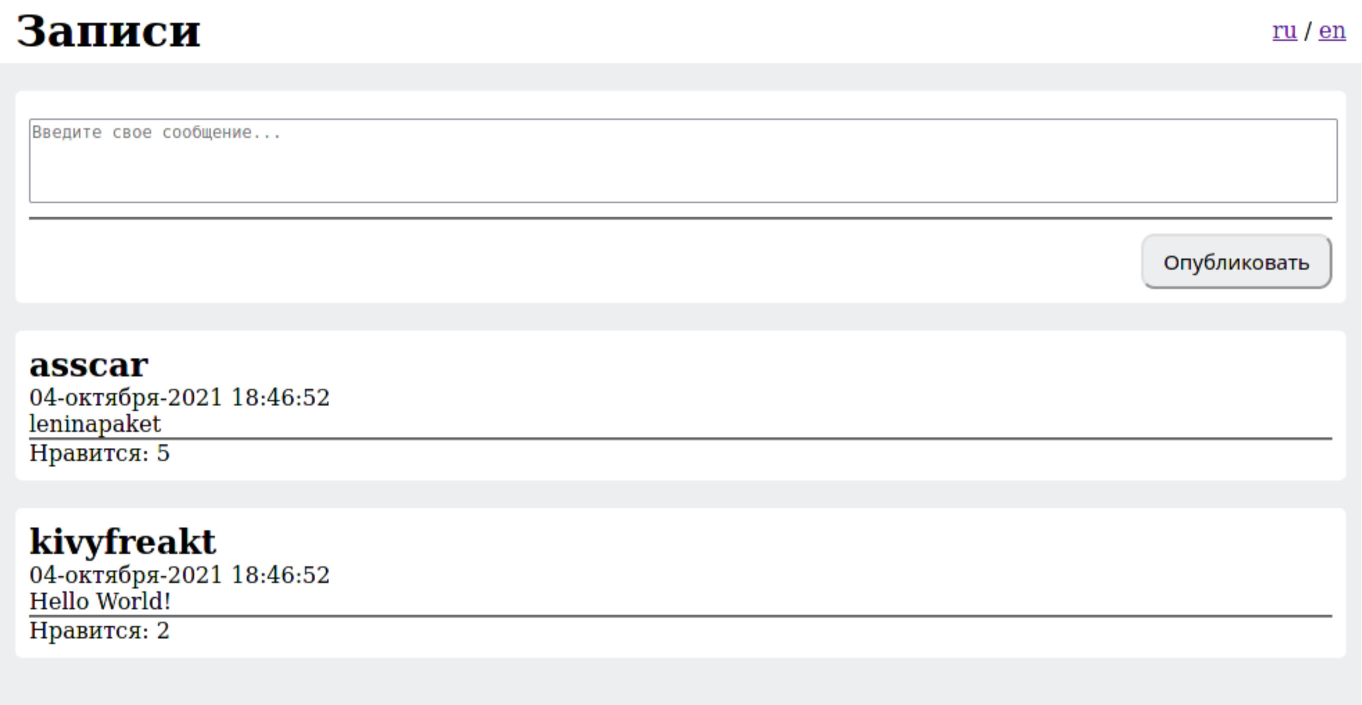
Этот сайт просит вас войти.

Имя пользователя

kivyfreakt

Пароль

•••••

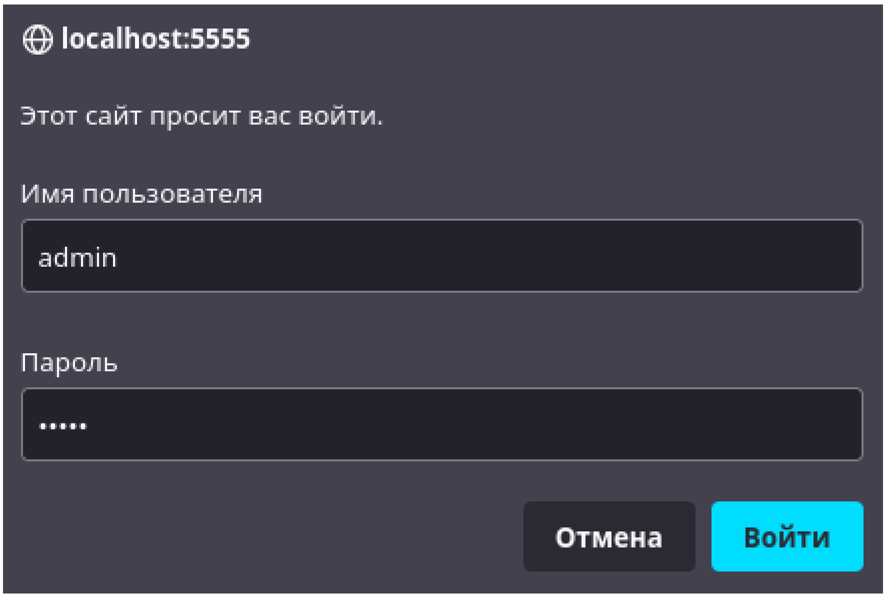


Записи

asscar
04-октября-2021 18:46:52
leninapaket
Нравится: 5

kivyfreakt
04-октября-2021 18:46:52
Hello World!
Нравится: 2

Теперь авторизуемся за пользователя admin с одноименным паролем. Без перезапуска сервера для повторной авторизации потребуется сбросить cookie-файлы браузера, зайти через другой браузер или зайти в режиме инкогнито.

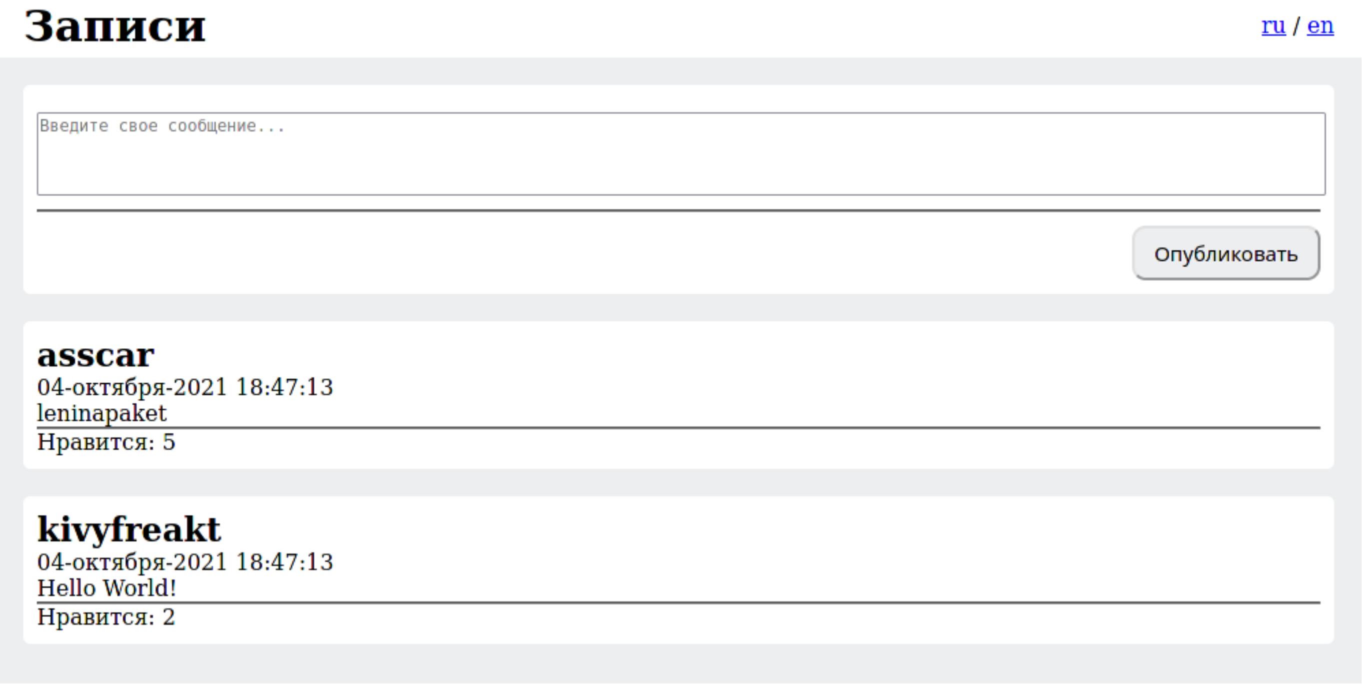


localhost:5555

Этот сайт просит вас войти.

Имя пользователя: admin

Пароль: •••••



Записи

ru / en

Введите свое сообщение...

asscar
04-октября-2021 18:47:13
leninapaket
Нравится: 5

kivyfreakt
04-октября-2021 18:47:13
Hello World!
Нравится: 2

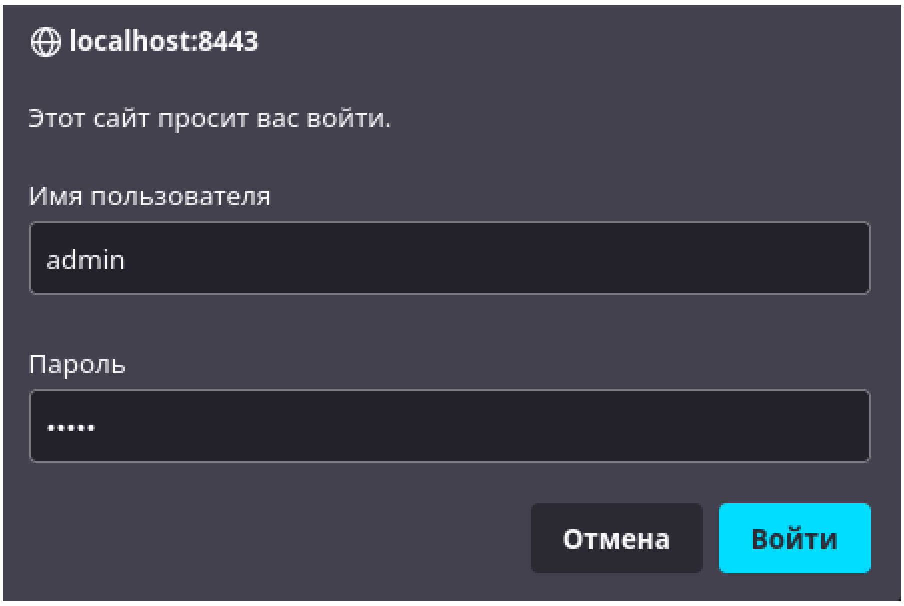
Важно заметить, что мы поставили авторизацию с корневой ссылки “/”, поэтому напрямую перейти по URL мы также не сможем без авторизации.

Настройка SSL-протокола

Сначала сгенерируем хранилище ключей и сам ключ в папке conf в корневой директории сервера Apache Tomcat и отредактируем server.xml для добавления SSL ключа.

Теперь проверим работу защищённого протокола, перейдя по ссылке https://localhost:8443/social

Подтверждаем переход, авторизуемся как пользователь admin:



localhost:8443

Этот сайт просит вас войти.

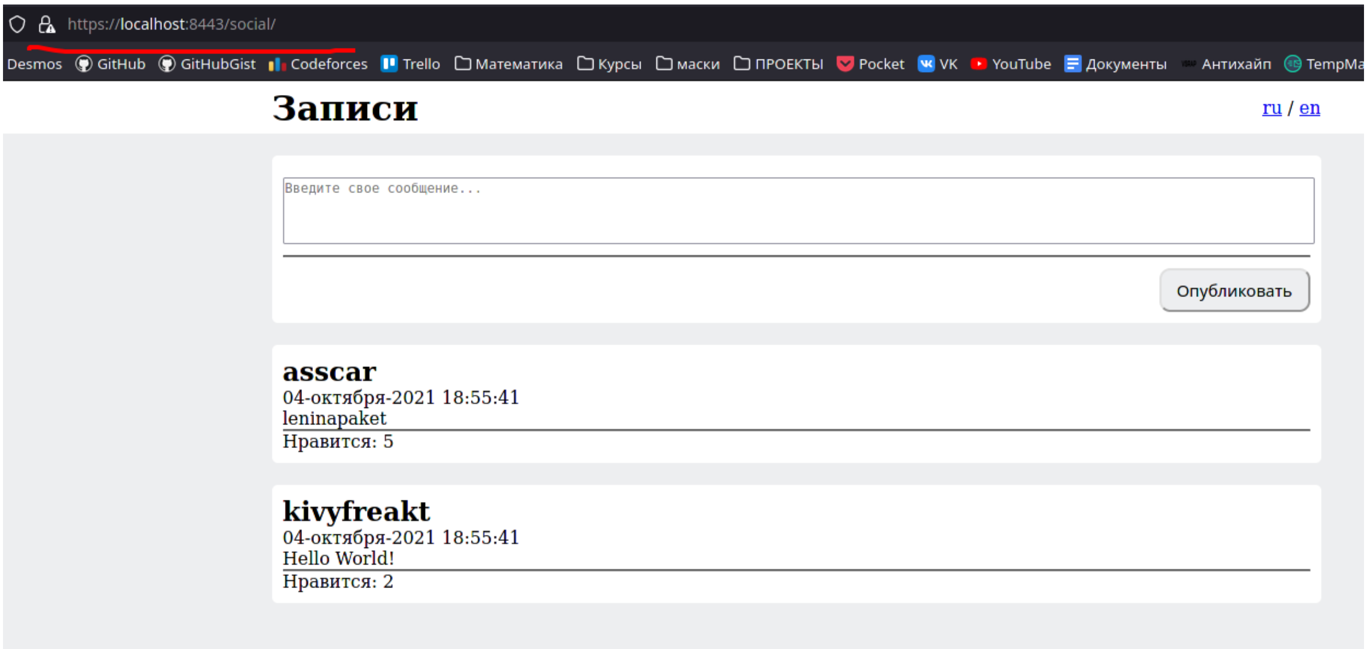
Имя пользователя

admin

Пароль

•••••

Успешно переходим на главную страницу:



Записи

asscar
04-октября-2021 18:55:41
leninapaket
Нравится: 5

kivyfreakt
04-октября-2021 18:55:41
Hello World!
Нравится: 2

Вывод

В ходе выполнения лабораторной работы была изучены технологии аутентификации: базовая, а также при помощи протокола SSL. Было создано хранилище ключей, а также разобран способ защиты содержимого сервера при помощи ролей и прав доступа.

Приложение 1. (tomcat-users.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
--><tomcat-users version="1.0" xmlns="http://tomcat.apache.org/xml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd">
<!--
  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary.

  Built-in Tomcat manager roles:
    - manager-gui    - allows access to the HTML GUI and the status pages
    - manager-script - allows access to the HTTP API and the status pages
    - manager-jmx    - allows access to the JMX proxy and the status pages
    - manager-status - allows access to the status pages only

  The users below are wrapped in a comment and are therefore ignored. If you
  wish to configure one or more of these users for use with the manager web
  application, do not forget to remove the <!.. ..> that surrounds them. You
  will also need to set the passwords to something appropriate.
-->
<!--
  <user username="admin" password="<must-be-changed>" roles="manager-gui"/>
  <user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
```

```
<!--
  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->

<role rolename="admin"/>
<role rolename="user"/>
<user username="admin" password="admin" roles="admin"/>
<user username="kivyfreakt" password="12345" roles="user"/>
<user username="user" password="123" roles="user"/>

</tomcat-users>
```

Приложение 2. (server.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
--><!-- Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/server.html
 --><Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener"/>
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!-- APR library loader. Documentation at /docs/apr.html -->
  <Listener SSLEngine="on" className="org.apache.catalina.core.AprLifecycleListener"/>
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>

  <!-- Global JNDI resources
       Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource auth="Container" description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory" name="UserDatabase"
pathname="conf/tomcat-users.xml" type="org.apache.catalina.UserDatabase"/>
  </GlobalNamingResources>

  <!-- A "Service" is a collection of one or more "Connectors" that share
       a single "Container" Note:  A "Service" is not itself a "Container",
       so you may not define subcomponents such as "Valves" at this level.
       Documentation at /docs/config/service.html
   -->
  <Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread pools-->
    <!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
    -->


    <!-- A "Connector" represents an endpoint by which requests are received
         and responses are returned. Documentation at :
         HTTP Connector: /docs/config/http.html
         AJP  Connector: /docs/config/ajp.html
         Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector connectionTimeout="20000" port="5555" protocol="HTTP/1.1" redirectPort="8443"
useBodyEncodingForURI="true"/>
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2
         This connector uses the NIO implementation. The default
         SSLImplementation will depend on the presence of the APR/native
         library and the useOpenSSL attribute of the
         AprLifecycleListener.
         Either JSSE or OpenSSL style configuration may be used regardless of
         the SSLImplementation selected. JSSE style configuration is used below.
    -->
  <Connector
      protocol="org.apache.coyote.http11.Http11NioProtocol"
      port="8443"
      maxThreads="150"
      SSLEnabled="true"
      useBodyEncodingForURI="true">
    <SSLHostConfig>
      <Certificate
        certificateKeystoreFile="$conf/lab5"
        certificateKeystorePassword="123456"
        type="RSA"
        />
      </SSLHostConfig>
  </Connector>

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <!--
    <Connector protocol="AJP/1.3"
               address="::1"
               port="8009"
               redirectPort="8443" />
    -->

    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.  The Engine implementation for Tomcat stand alone
         analyzes the HTTP headers included with the request, and passes them
         on to the appropriate Host (virtual host).
         Documentation at /docs/config/engine.html -->

    <!-- You should set jvmRoute to support load-balancing via AJP ie :
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    -->
    <Engine defaultHost="localhost" name="Catalina">

      <!--For clustering, please take a look at documentation at:
          /docs/cluster-howto.html  (simple how to)
          /docs/config/cluster.html (reference documentation) -->
      <!--
      <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

```
      -->

      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
      </Realm>

      <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">

        <!-- SingleSignOn valve, share authentication between web applications
             Documentation at: /docs/config/valve.html -->
        <!--
        <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
        -->

        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using pattern="common" -->
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="%h %l
%u %t &quot;%r&quot; %s %b" prefix="localhost_access_log" suffix=".txt"/>

      <Context docBase="/home/kivyfreakt/temp/tomcat10/wtpwebapps/social" path="/social"
reloadable="true" source="org.eclipse.jst.jee.server:social"/></Host>
    </Engine>
  </Service>
</Server>
```

Приложение 3. (web.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:web="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" id="WebApp_ID" version="2.4">
 <display-name>social</display-name>

 <welcome-file-list>
  <welcome-file>Main.jsp</welcome-file>
 </welcome-file-list>

 <servlet>
  <servlet-name>Main</servlet-name>
  <jsp-file>/Main.jsp</jsp-file>
 </servlet>
 <servlet-mapping>
  <servlet-name>Main</servlet-name>
  <url-pattern>/main</url-pattern>
 </servlet-mapping>
 <servlet>
  <servlet-name>Post</servlet-name>
  <jsp-file>/Post.jsp</jsp-file>
 </servlet>
 <servlet-mapping>
  <servlet-name>Post</servlet-name>
  <url-pattern>/post</url-pattern>
 </servlet-mapping>

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

  <security-role>
    <role-name>user</role-name>
  </security-role>

  <security-constraint>
    <web-resource-collection>
```

```
      <web-resource-name>/</web-resource-name>
      <url-pattern>/</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>

  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>Write Post List</realm-name>
  </login-config>

</web-app>
```