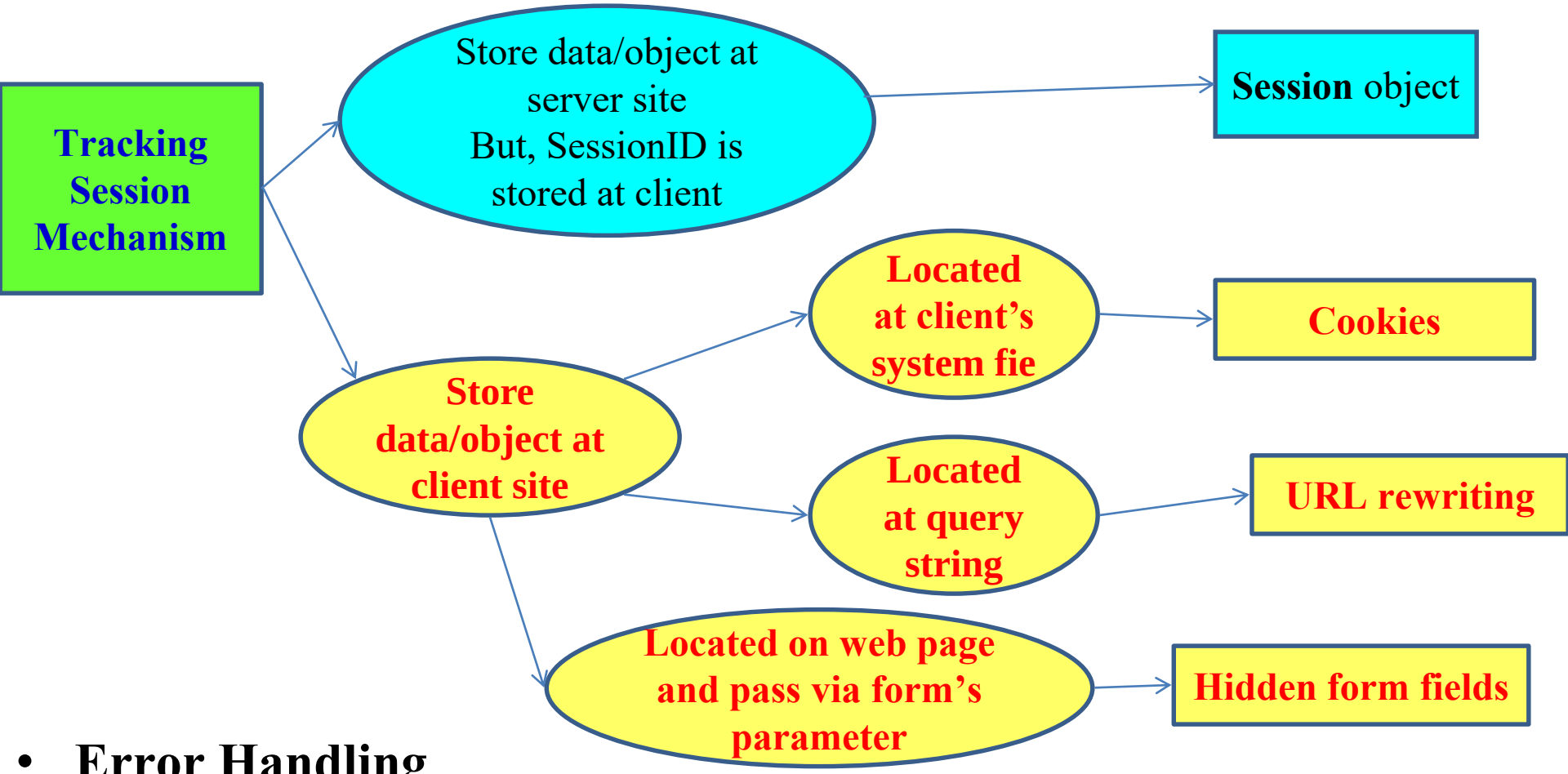
Fpt University

Security

Security Mechanisms
Deployment Descriptor Security Declarations
Authentication Types

Review

- **Session Tracking Mechanism**
  - Client must be stored the value that transfer to server in each its request

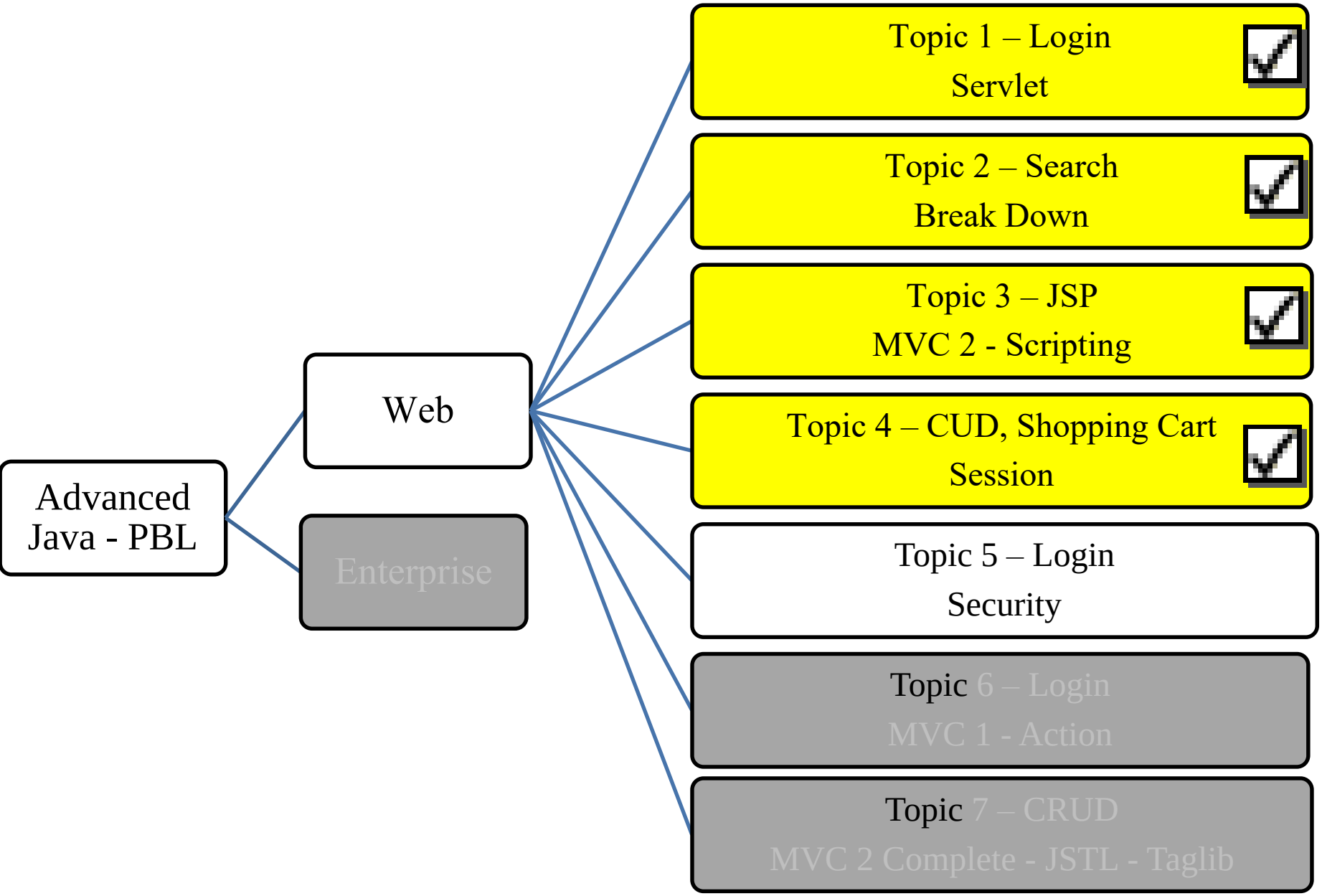


- **Error Handling**
  - Reporting Error: create the friendly UI to user when the system's errors occur.
  - Logging Error: store the errors (users or/and app) to the file to improve the application and get users' behaviors

Objectives

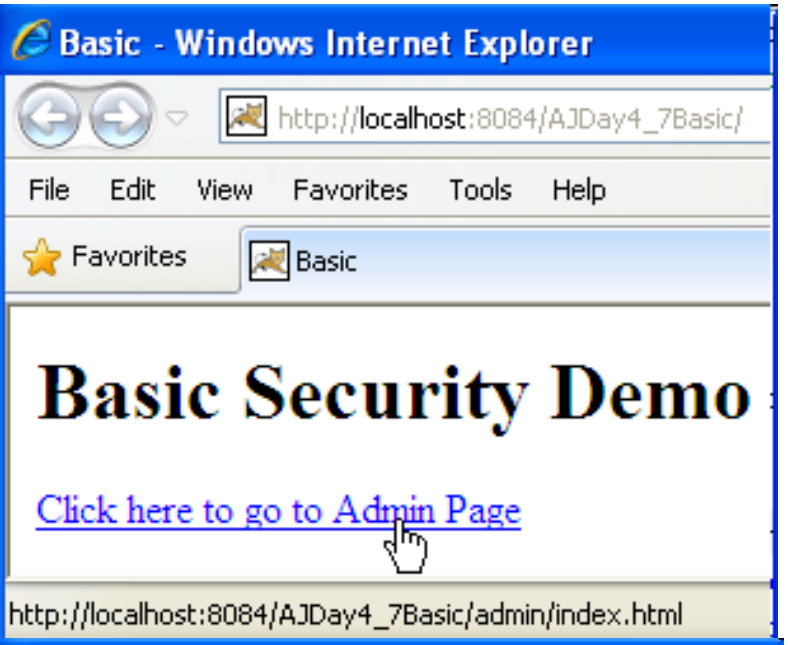
- **How to construct the security on the web site**
  - Authentication and Authorization with Basic, Digest, and Form
  - Confidentiality with HTTPS Client

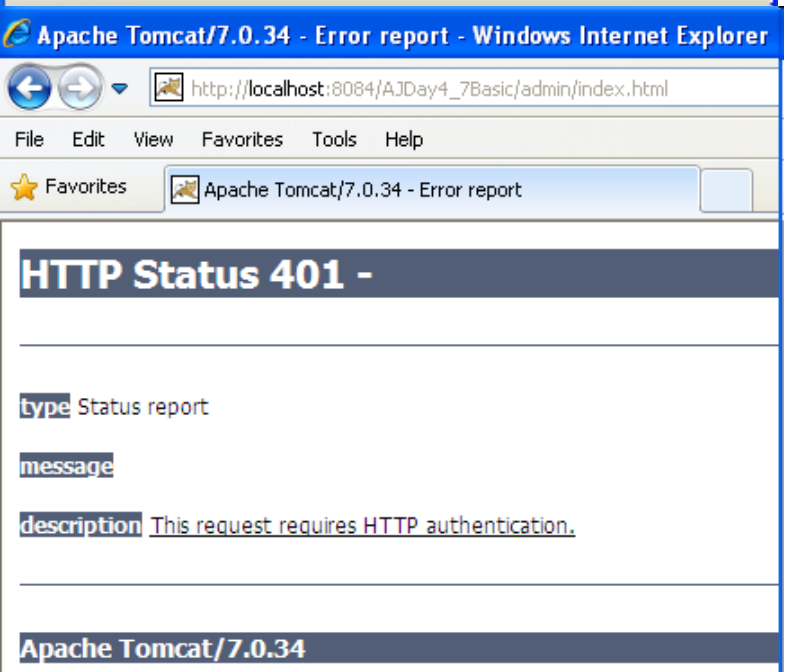
Objectives

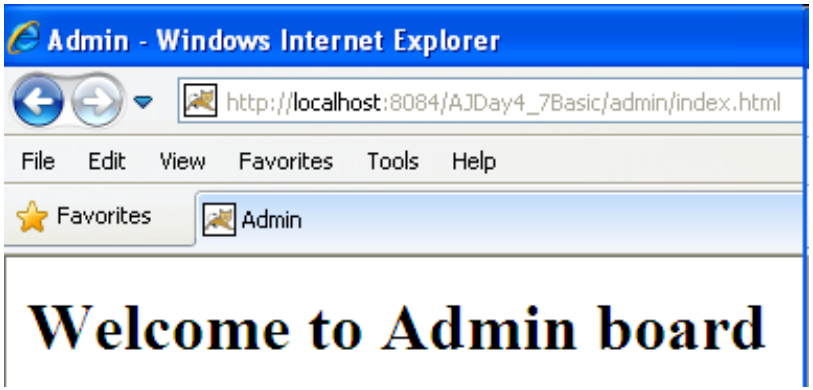

Advanced Java - PBL
Web
Enterprise
Topic 1 – Login Servlet
Topic 2 – Search Break Down
Topic 3 – JSP MVC 2 - Scripting
Topic 4 – CUD, Shopping Cart Session
Topic 5 – Login Security
Topic 6 – Login MVC 1 - Action
Topic 7 – CRUD MVC 2 Complete - JSTL - Taglib

Authentication Types

BASIC – Expectation







Authentication Types

BASIC – Expectation

Apache Tomcat/7.0.34 - Error report - Windows Internet Explorer

http://localhost:8084/AJDay4_7Basic/admin/index.html

File Edit View Favorites Tools Help

Favorites | Apache Tomcat/7.0.34 - Error report

HTTP Status 403 - Access to the requested resource has been denied

---

type Status report

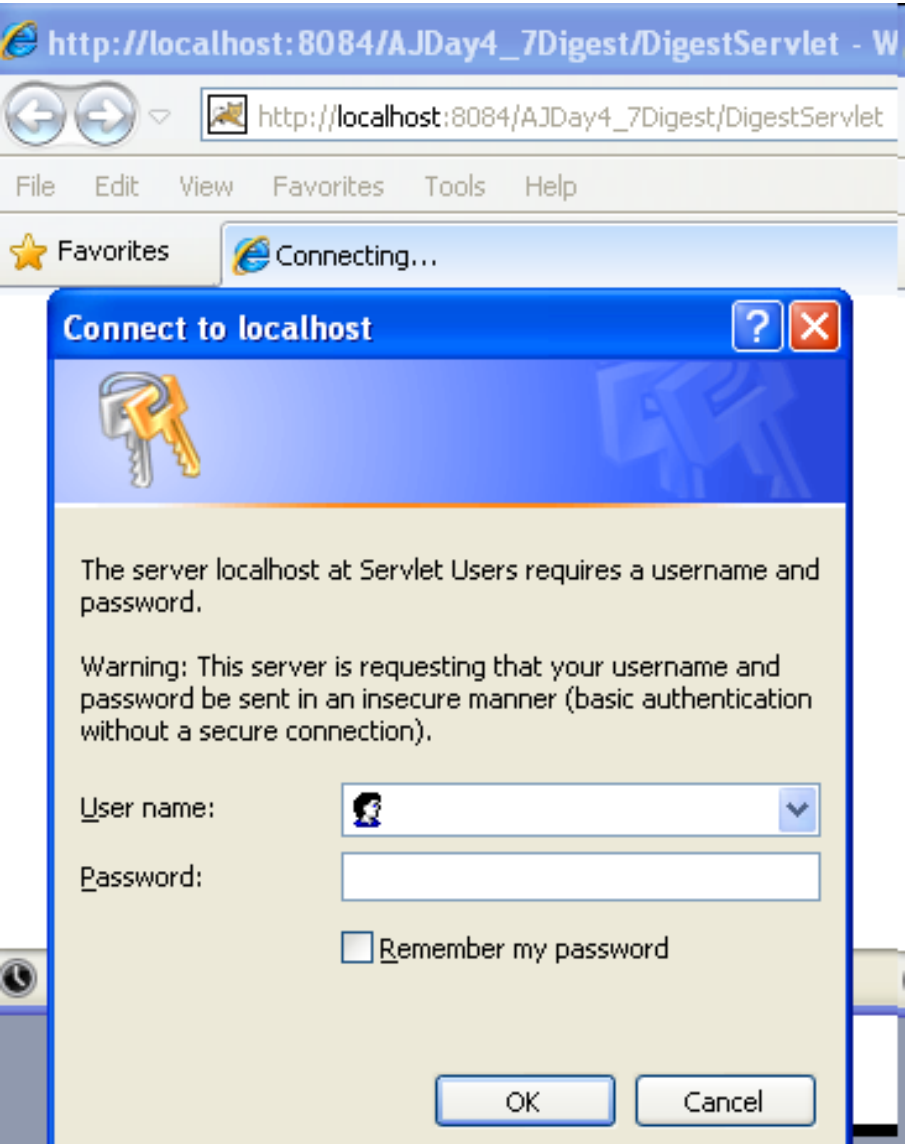
message Access to the requested resource has been denied

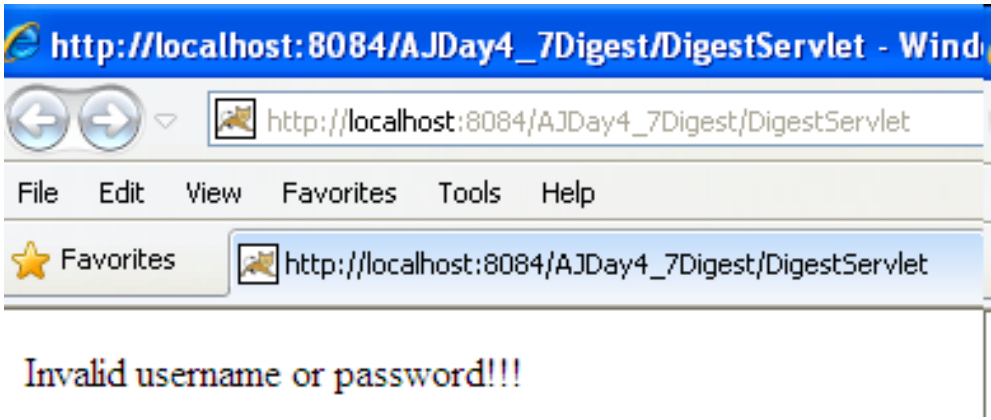
description Access to the specified resource has been forbidden.

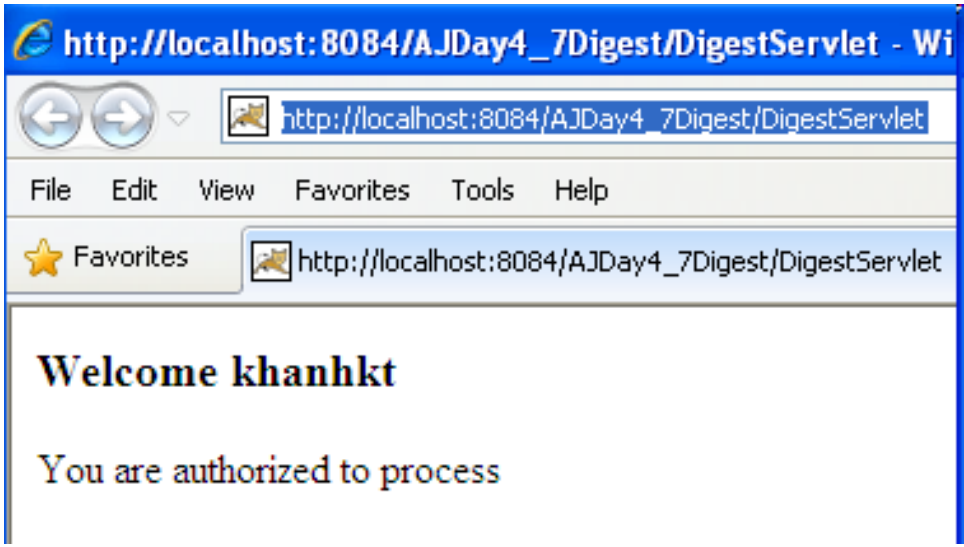
Apache Tomcat/7.0.34

Authentication Types

DIGEST – Expectation

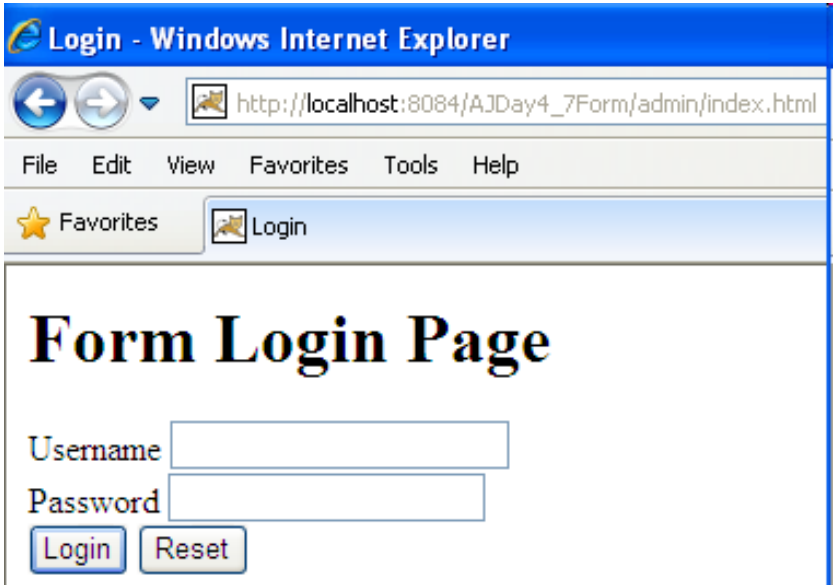






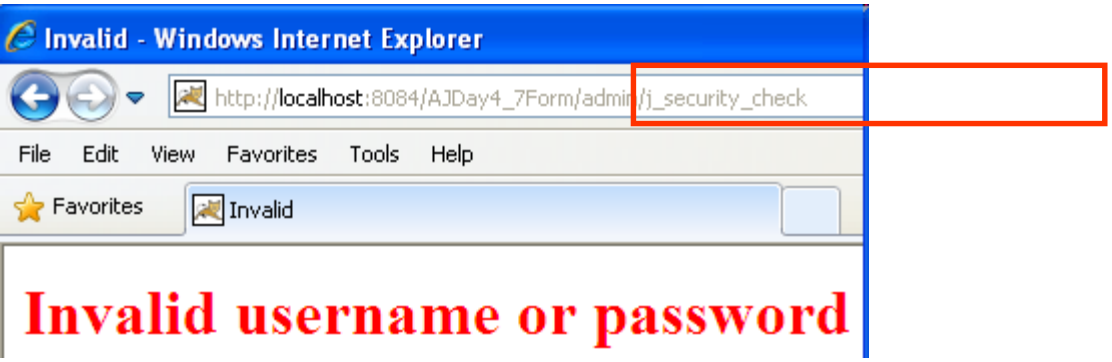
Authentication Types

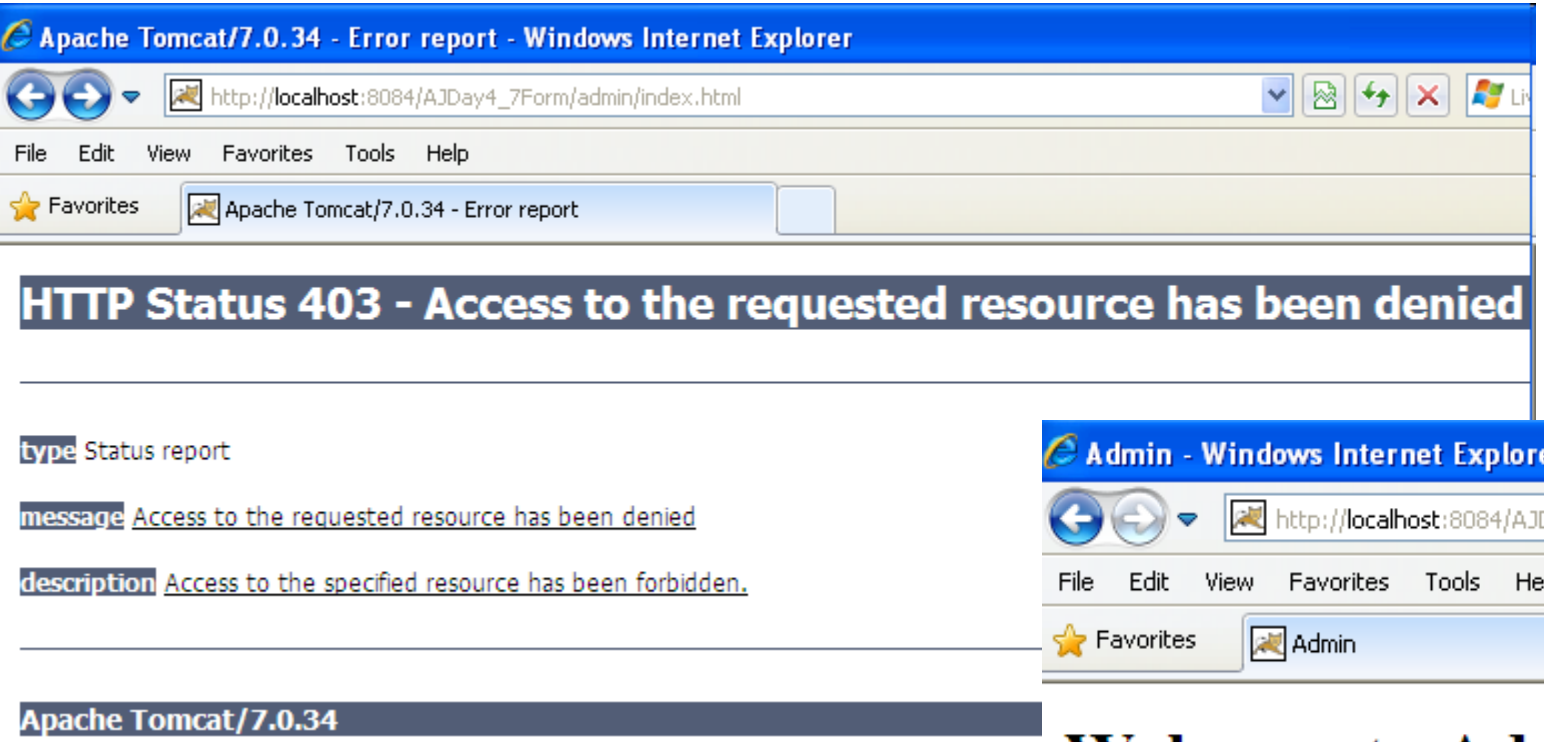
FORM – Expectation

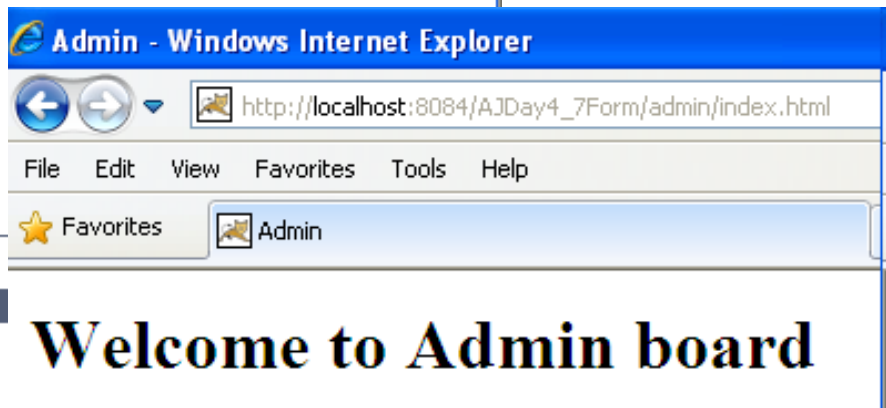


Authentication Types

FORM – Expectation

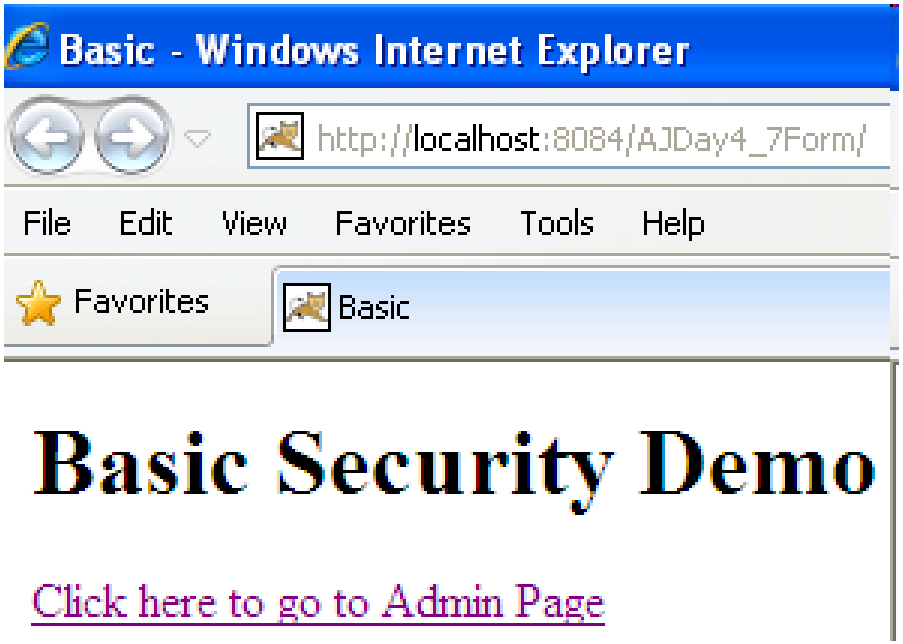


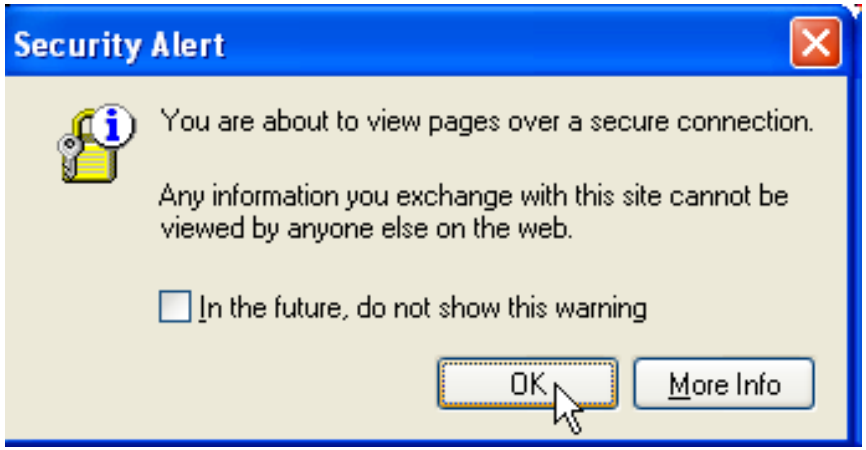




Authentication Types

HTTPS CLIENT – Expectation





Authentication Types

HTTPS CLIENT – Expectation

Certificate Error: Navigation Blocked - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/index.html

File Edit View Favorites Tools Help

Favorites | Certificate Error: Navigation Blocked

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.
The security certificate presented by this website was issued for a different website's address.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

More information

http://localhost:8084/AJDay4_7Form/admin/index.html

Security Alert

You are about to view pages over a secure connection.

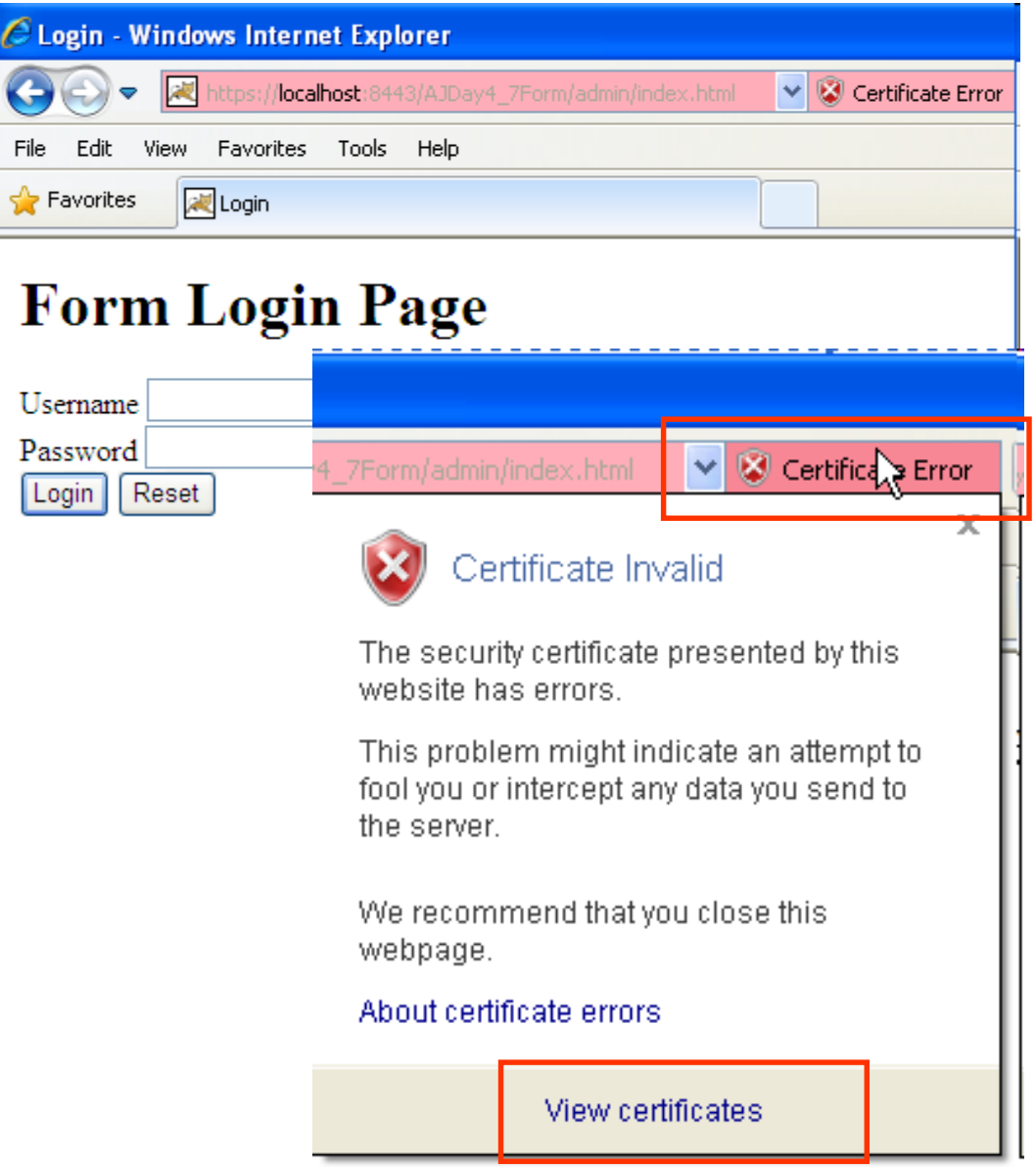
Any information you exchange with this site cannot be viewed by anyone else on the web.

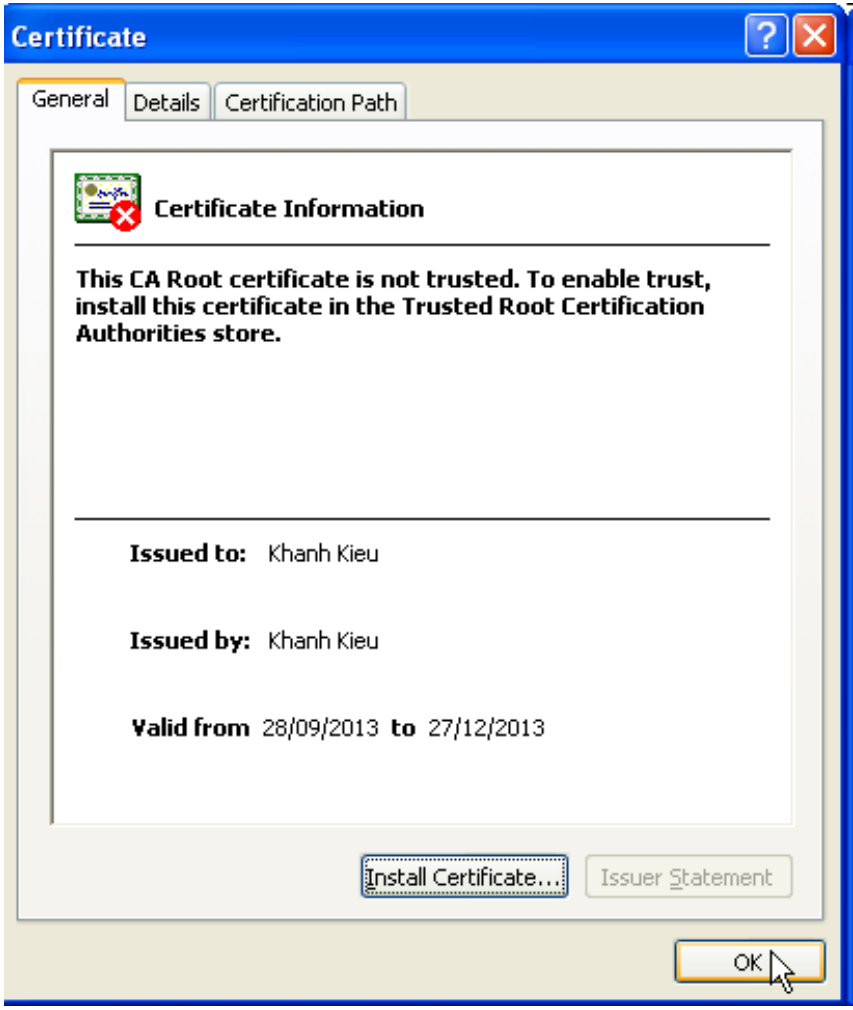
In the future, do not show this warning

OK | More Info

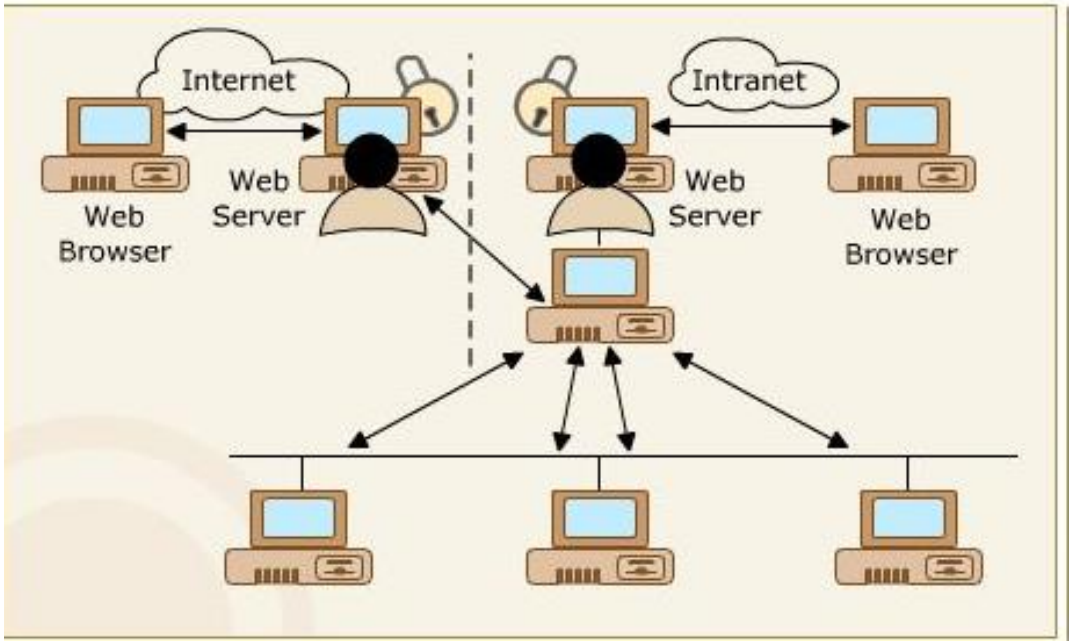
Authentication Types

HTTPS CLIENT – Expectation





Need of Securing Web Application



- Is accessed over a network such as Internet / Intranet
- Access to confidential information by unauthorized users
- Unauthorized use of resources
- Heavy traffic
- Malicious Code

Security Mechanisms

- There are **4** security **mechanisms**
  - **Authentication**
    - **Ensures** that the **identity** of the **client** is true
    - Can be **achieved** through **basic means** (user IDs and password) vs. **complex means** (digital certificates)
  - **Authorization** (access to controlled resources)
    - **After** the **completion** of **authentication**, **permissions** have to **granted** to the user to **perform** all **desired operations** or **resources**. (Grant permissions to the correct user)
  - **Data integrity**
    - Is the **process** of **ensuring** that **any messages passed** through a network have **not been tampered** with in **transit**
  - **Confidentiality** (data privacy)
    - This keep **information hidden from people other** than the involved client and server
    - The **encryption process** is used

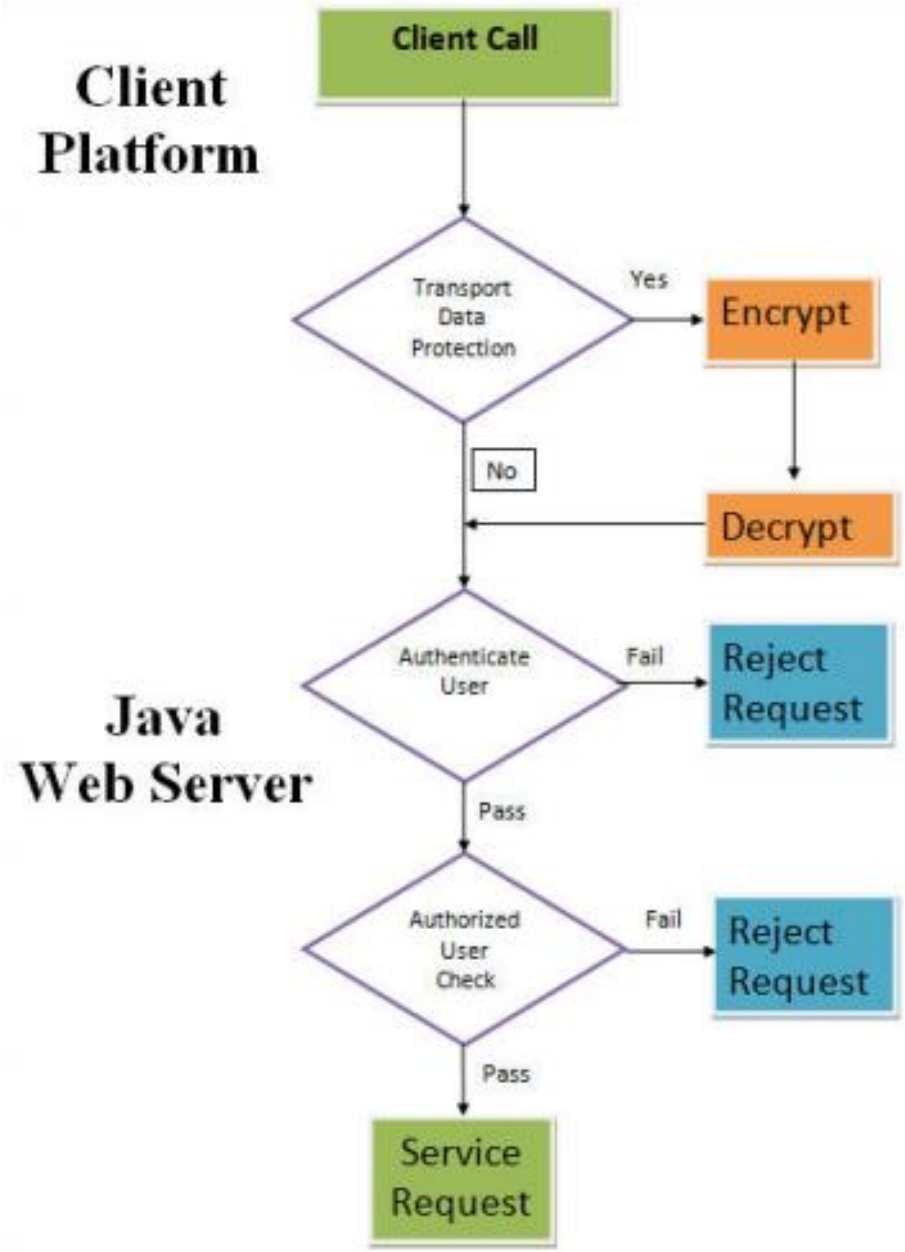
Security Mechanisms

General User

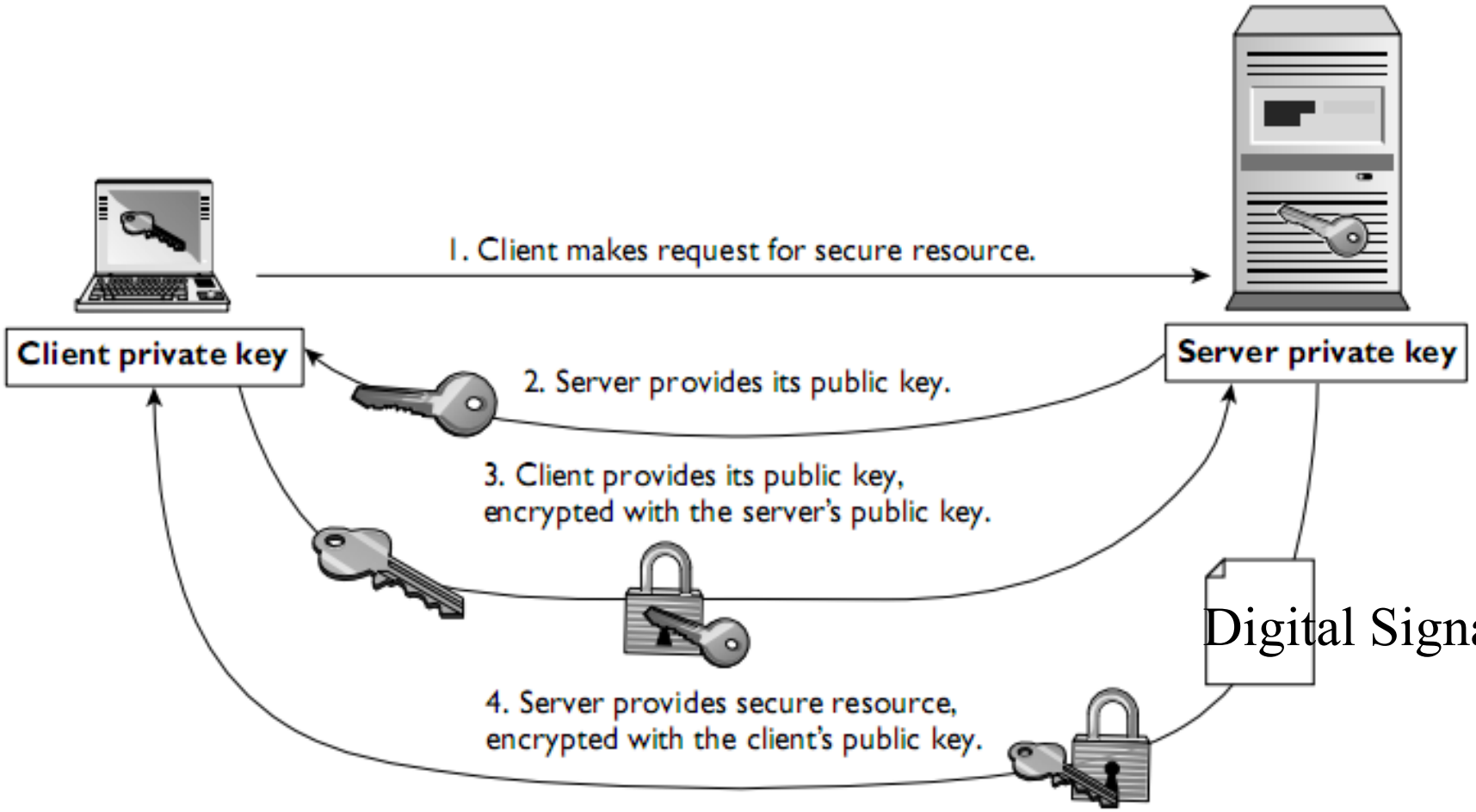
Logging In

Authentication

Well-defined security Role

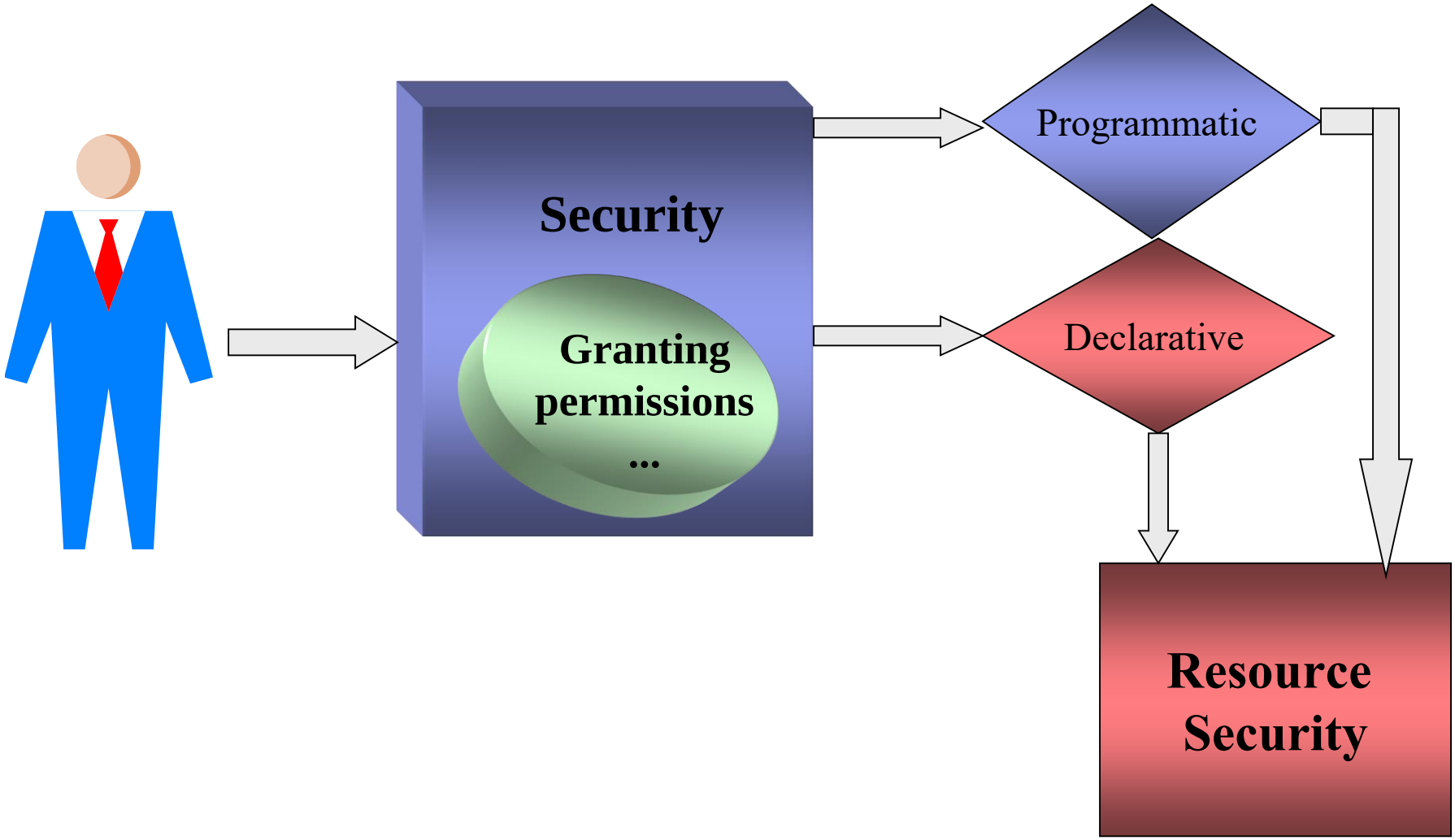


Security Mechanisms



Digital Signature

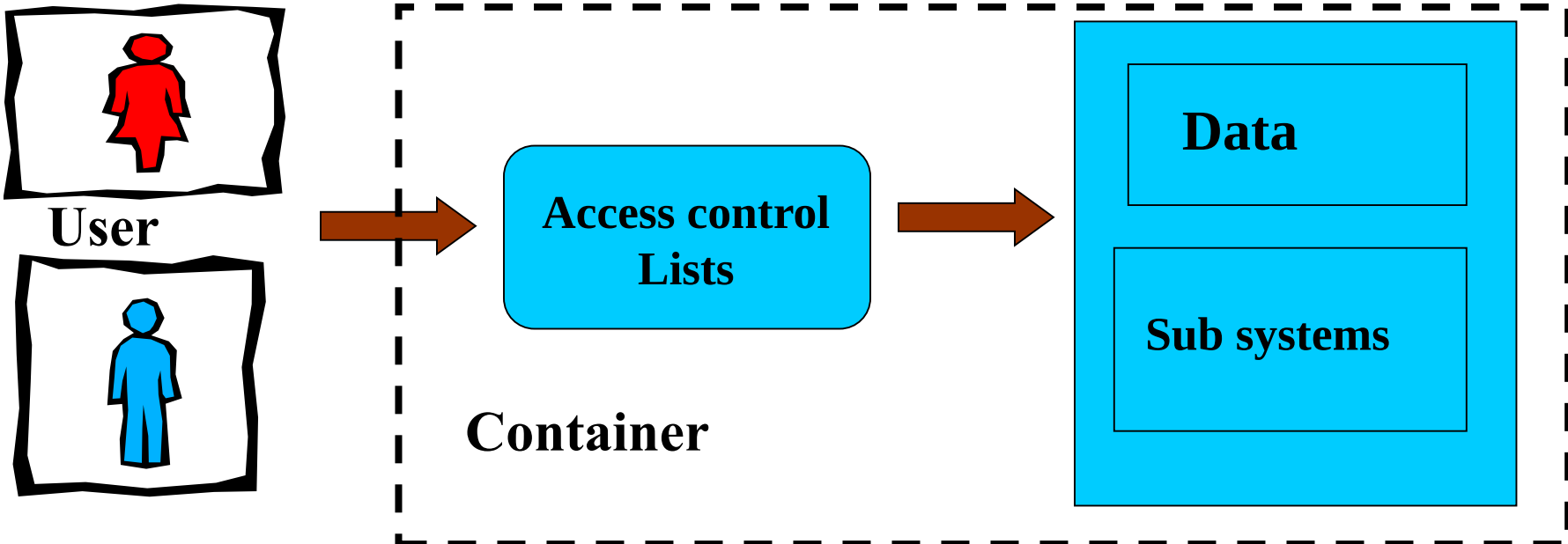
Security Mechanisms



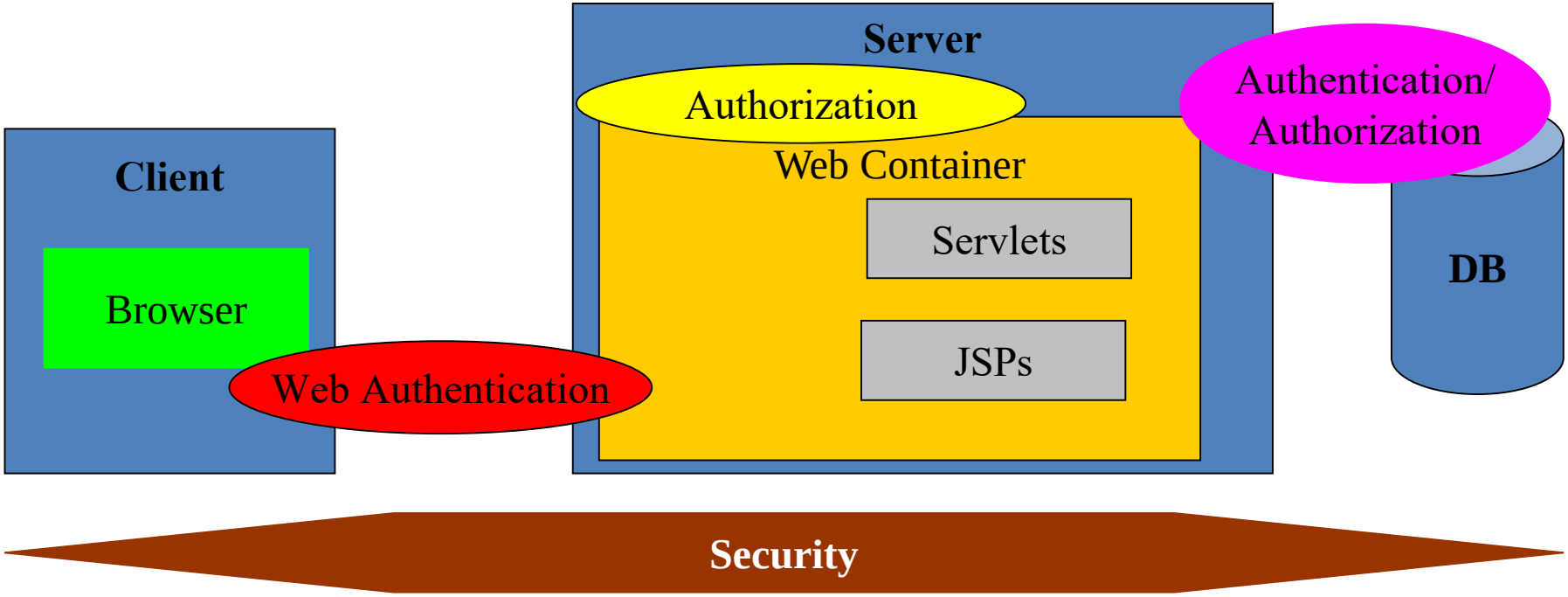
Security Mechanisms

J2EE/JavaEE is applied JAAS

- **Access Control List (ACL)**
  - An ACL file is made up of entries, which contain a set of permissions for a particular resource and a set of users who can access those resources
    - Permissions represent a **right to access a particular resource** or to **perform some action** on an application **to lists of users and groups** that are created by an administrator
  - **Ex**: In Web Tomcat Server, it is a **tomcat-users.xml** file
  - **When** user **request** any **resources** from the server, the **container uses ACL** to **check** the **users' role** for **accessing** the requested resource **with** his/her **permission**



Security Mechanisms
J2EE/JavaEE is applied JAAS



DD Security Declaration

The <security-constraint> element

```
<security-constraint>
 <web-resource-collection>
  <description>description</description>
  <web-resource-name>resourceName</web-resource-name>
  <url-pattern>/resource</url-pattern>
  <http-method>METHOD</http-method>
 </web-resource-collection>
 <auth-constraint>
   <description>description</description>
   <role-name>roleName</role-name>
 </auth-constraint>
 <user-data-constraint>
    <description>description</description>
    <transport-guarantee>NONE|INTEGRAL|CONFIDENTIAL</transport-
   guarantee>
 </user-data-constraint>
</security-constraint>
```

DD Security Declaration

The <security-constraint> element

- **<security-constraint>** is used to **associate resources** and **HTTP methods with logical roles** for
  - **Authorization**
  - Guaranteeing on resource **security in transit**
- **<web-resource-collection>** defines the **resources** to be **secured**
- **<auth-constraint>** **lists** the **named roles authorized** to the resources defined in the web resource collection. **If it is not exits, all clients can access the web resource without any permission**
- **<user-data-constraint>** defines **guarantees** on the network **used to transmit resources to clients**
  - **NONE**: no guarantee (is equivalent to omitting)
  - **INTEGRAL**: the web server must be able to **detect any tampering with** HTTP requests and responses for protected resources
  - **CONFIDENTAL**: the web server must be **ensure the content** of HTTP requests and responses so that protected resources **remain secret to all but authorized parties**

DD Security Declaration

The <security-constraint> element – Example

```
<security-constraint>
    <display-name>Constraint1</display-name>
    <web-resource-collection>
        <web-resource-name>AdminPage</web-resource-name>
        <description/>
        <url-pattern>/admin/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>manager</role-name>
    </auth-constraint>
</security-constraint>
```

DD Security Declaration

The <security-constraint> element – Example

```
<security-constraint>
    <display-name>FormAuth</display-name>
    <web-resource-collection>
        <web-resource-name>FormResource</web-resource-name>
        <description/>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>manager</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

DD Security Declaration

ACL on Tomcat Server – Example

tomcat-users.xml

Source History

```xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="admin"/>
    <role rolename="manager-script"/>
    <role rolename="user"/>
    <role rolename="tomcat"/>
    <user username="ide" password="tDBmxPNO" roles="manager-script,admin"/>
    <user username="khanhkt" password="trongkhanh" roles="manager-script,admin"/>
    <user username="tomcat" password="tomcat" roles="tomcat"/>
    <user username="guest" password="123456" roles="user"/>
</tomcat-users>
```

DD Security Declaration

The <login-config> element

```
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name/>
</login-config>
```

```
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name/>
  <form-login-config>
    <form-login-page>loginPage<form-login-page/>
    <form-error-page>errorPage<form-error-page>
  </form-login-config>
</login-config>
```

```
<login-config>
  <auth-method>DIGEST|CLIENT-CERT</auth-method>
</login-config>
```

DD Security Declaration

The <login-config> element – Example

```
<security-constraint>
    <display-name>Constraint1</display-name>
    <web-resource-collection>
        <web-resource-name>BasicAuthentication</web-resource-name>
        <description/>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>manager</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name/>
</login-config>
```

DD Security Declaration

The <login-config> element – Example

```
<security-constraint>
    <display-name>FormAuth</display-name>
    <web-resource-collection>
        <web-resource-name>FormResource</web-resource-name>
        <description/>
        <url-pattern>/admin/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>manager</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name/>
    <form-login-config>
        <form-login-page>/admin/admin.jsp</form-login-page>
        <form-error-page>/fail.jsp</form-error-page>
    </form-login-config>
</login-config>
```

DD Security Declaration

The <security-role> element

<security-role>
  <description>**description**</description>
  <role-name>**roleName**</role-name>
</security-role>

```
<security-role>
    <description/>
    <role-name>manager</role-name>
</security-role>
```

Additional

Configuring Tomcat

- Addition the username, password to the tomcat-users.xml file is located at
  - **C:\Documents and Settings\LoggedUser\Application Data\NetBeans\ 7.4\apache-tomcat-7.0.41.0_base\conf\ tomcat-users.xml**
  - **C:\Users\LoggedUser\AppData\Roaming\NetBeans\7.4\ apache-tomcat-7.0.41.0_base\conf\tomcat-users.xml**
- **Ex:**

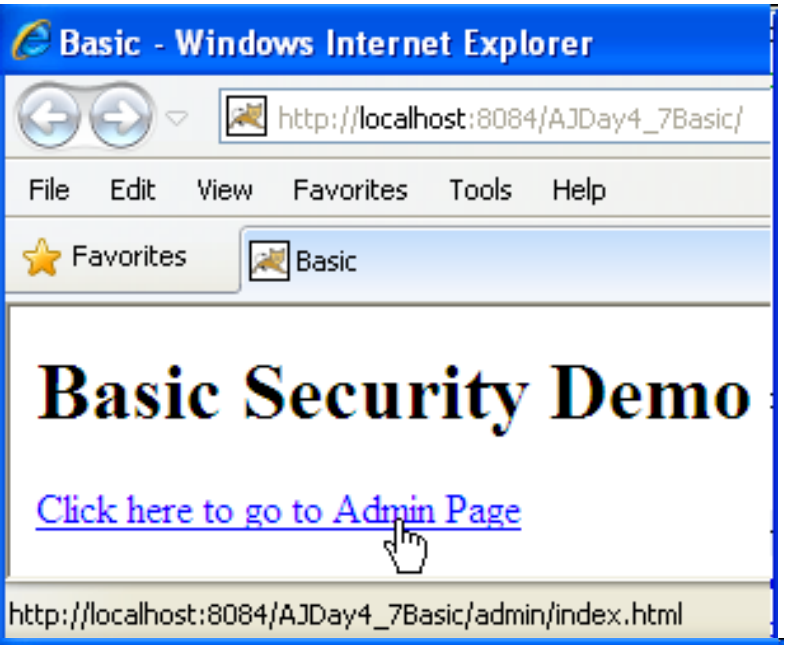
tomcat-users.xml

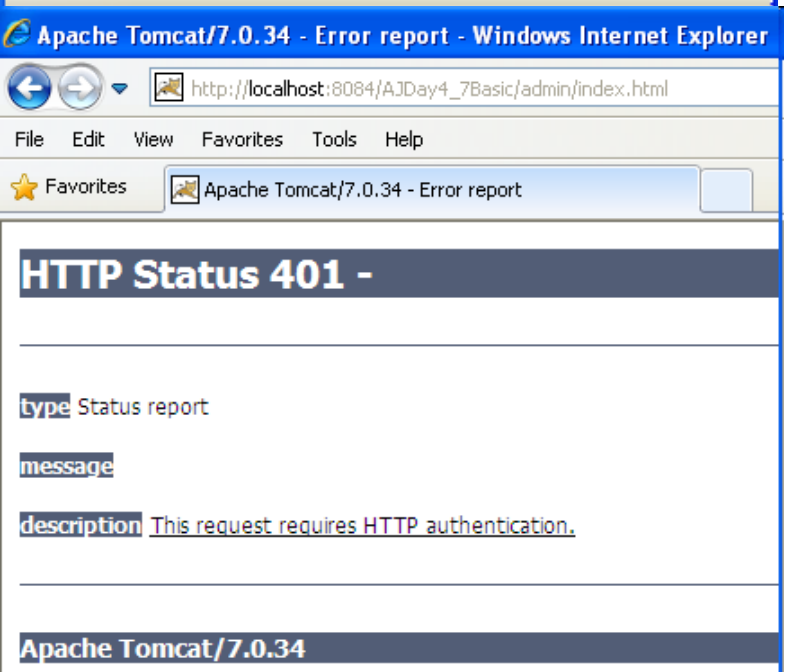
Source History

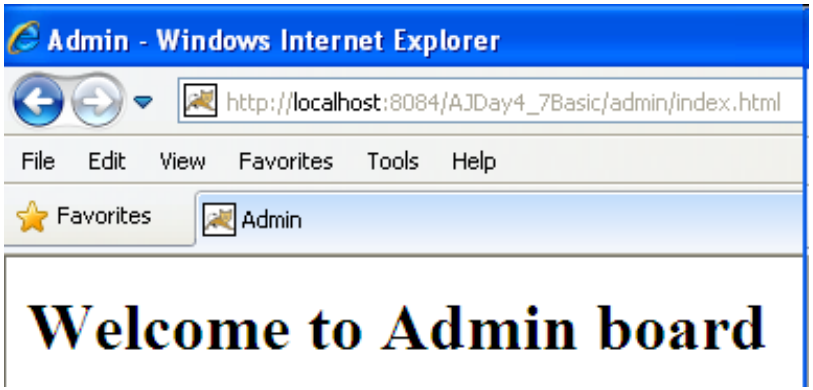
```
18 <tomcat-users>
19     <role rolename="admin"/>
20     <role rolename="manager-script"/>
21     <role rolename="tomcat"/>
22     <role rolename="role1"/>
23     <user username="tomcat" password="tomcat" roles="tomcat"/>
24     <user username="both" password="tomcat" roles="tomcat,role1"/>
25     <user username="role1" password="tomcat" roles="role1"/>
26     <user username="ide" password="2mdXj4dj" roles="manager-script,admin"/>
27     <user username="khanhkt" password="trongkhanh" roles="manager-script,admin"/>
28 </tomcat-users>
```

Authentication Types

BASIC

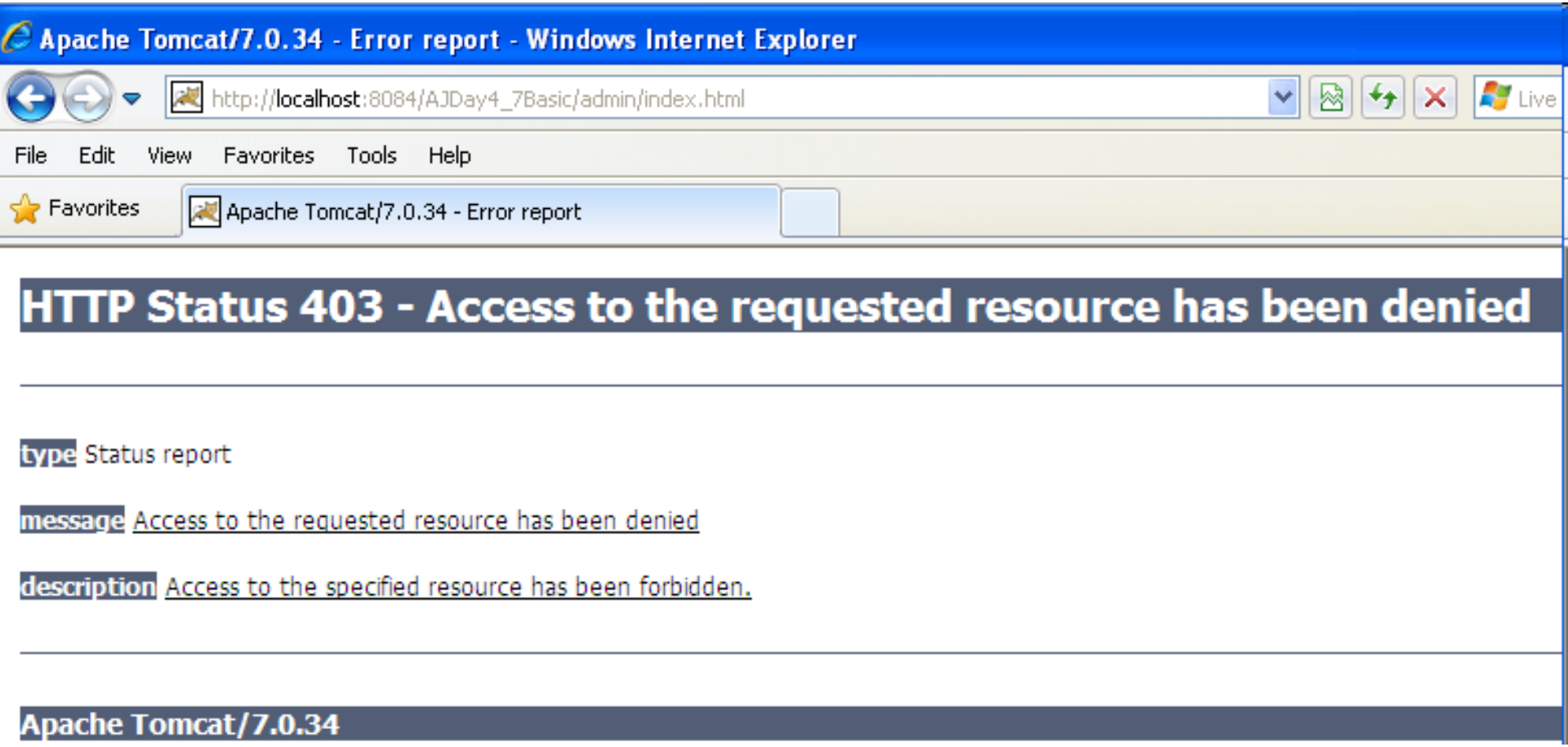






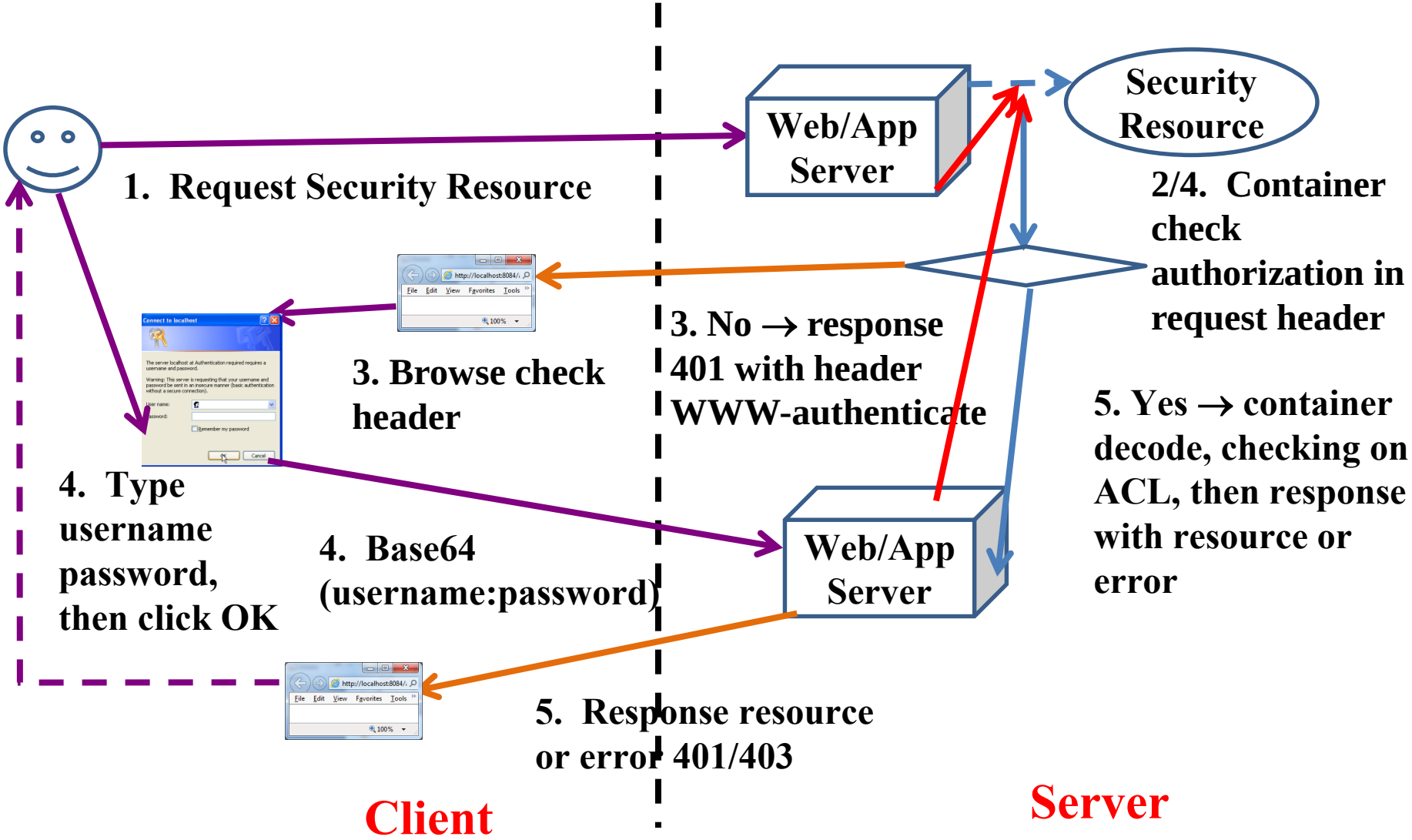
Authentication Types

BASIC



Authentication Types

BASIC



Authentication Types

BASIC

- The user sends request to access or request the web resource
- The **container uses ACL** to **authenticate** the **user permission when** the **secure web resource is accessed**
- The container **response** that **contains** the **authentication request attached in header response**
- Depending the response header, the **browse** is **triggered** to **show a standard dialog**. This dialog **allows entry of a username and password**, and displays a **realm name**
- When the **OK button** is pressed, the form "**username:password**' is processed on Base64 encoding and transmitted over the Internet/ network to the container
- The container **checks** the **authenticated information on ACL**
  - If the user does **not exists** in ACL, the container **send error code 401**
  - If the **user exists** in ACL but he/she does **not permit to access resource**, the container send **error code 403** (**access permission denied**)
  - **Otherwise**, the user can **access the requested web resource**

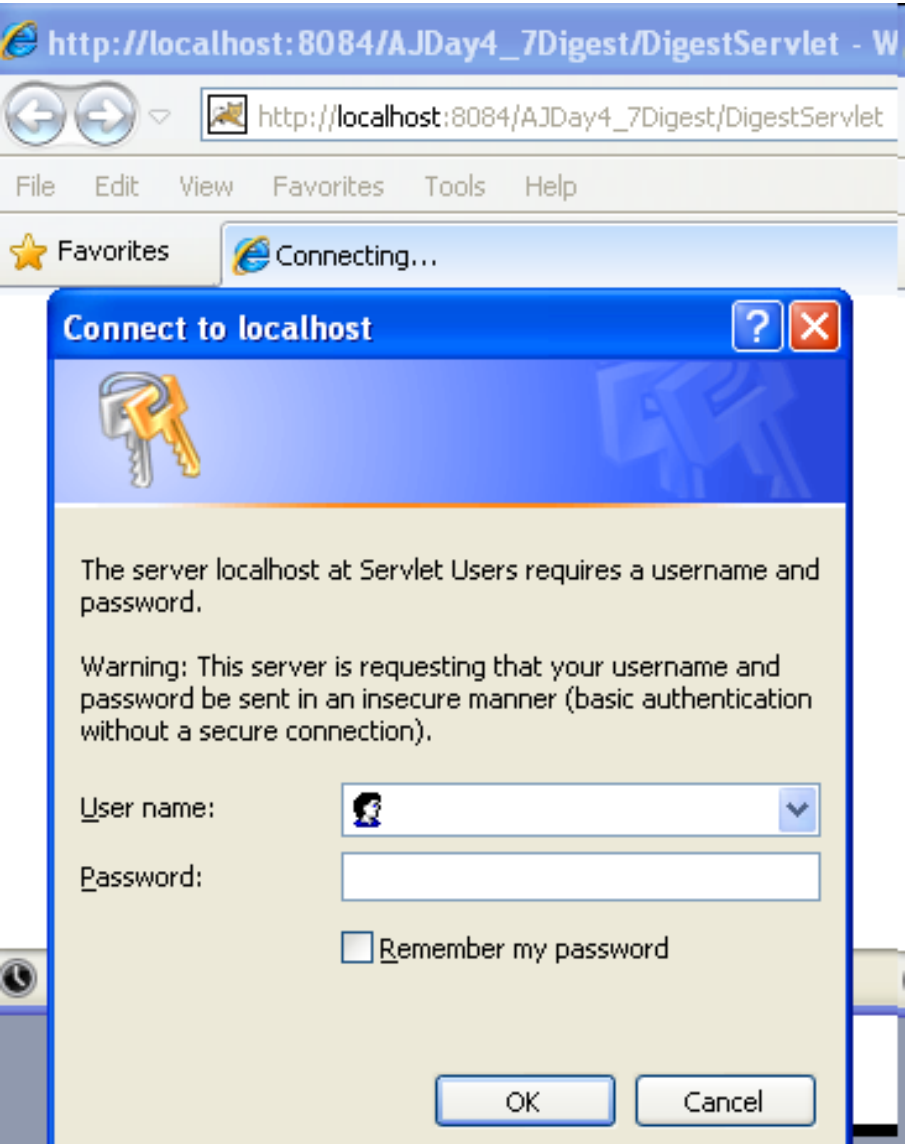
Authentication Types

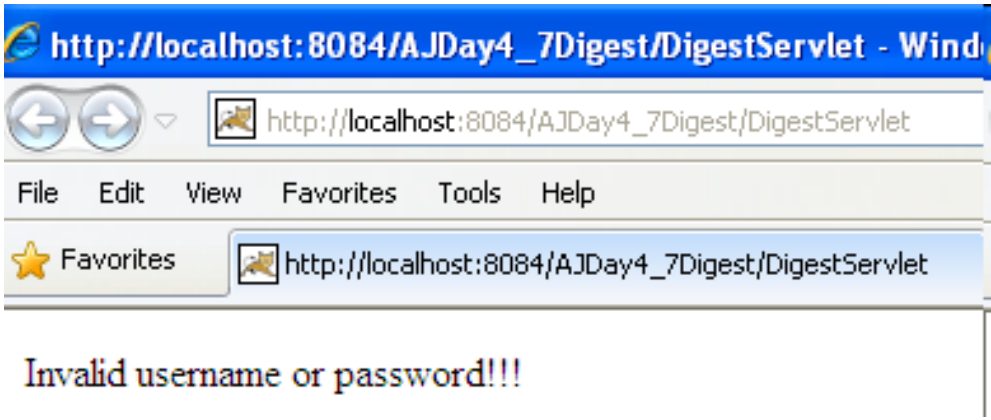
BASIC

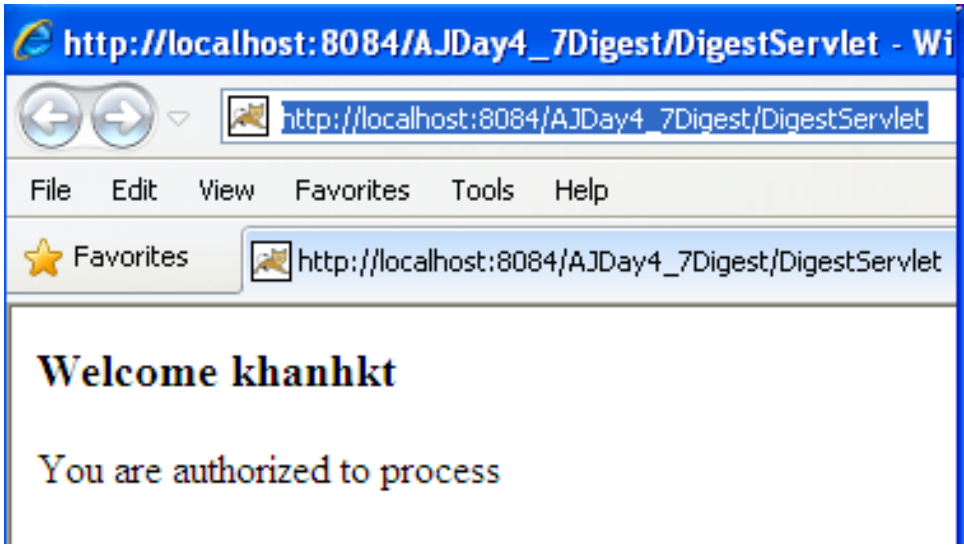
- **The realm**
  - Is a **database of users** and **groups identifying valid users** for the Web applications that are controlled by authentication policy
  - Are typically **repositories** containing users, groups, permissions, and secured resources.
  - **Contains** the usernames and passwords and the authentication and authorization **policy details** that control these users.
  - Can be **stored** in an **XML file, a text file**, and sometimes **even** in a **DBMS**
- **Advantages**
  - Browser **friendly**
- **Disadvantages**
  - It **does not provide any protection** for the information communicated between the client and the server. (It **works** on the **assumption** that the **client-server communication is reliable**)
  - There is **no logout mechanism** specified.

Authentication Types

DIGEST

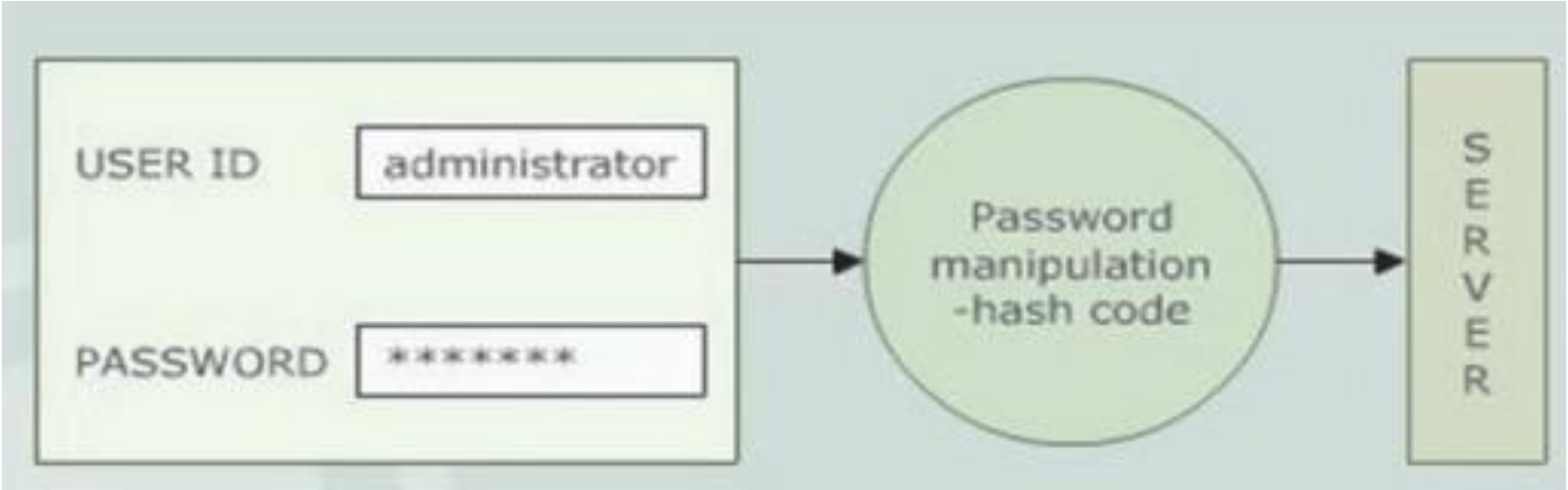






Authentication Types

DIGEST



- **Improves** a **little** on **BASIC** by **using** a **secure algorithm** to **encrypt** the **password** and other security details
- Builds over the **basic authentication but** the **password** is **first encrypted** and **then sent**
- Use **hash functions** to **secure web applications**
- A **one-way system** in process because it is **difficult to know the original password** according to the output
- **Advantages**: **impossible hacking**

Authentication Types

DIGEST

DigestServlet.java | web.xml

```
20   * @author Trong Khanh
21   */
22  public class DigestServlet extends HttpServlet {
23
24      Hashtable users = new Hashtable();
25
26      public void init() throws ServletException {
27          super.init();
28          users.put("khanhkt:123456", "allowed");
29          users.put("happykt:khanhkt", "");
30          users.put("funnykt:54321", "allowed");
31      }
```

Authentication Types

DIGEST

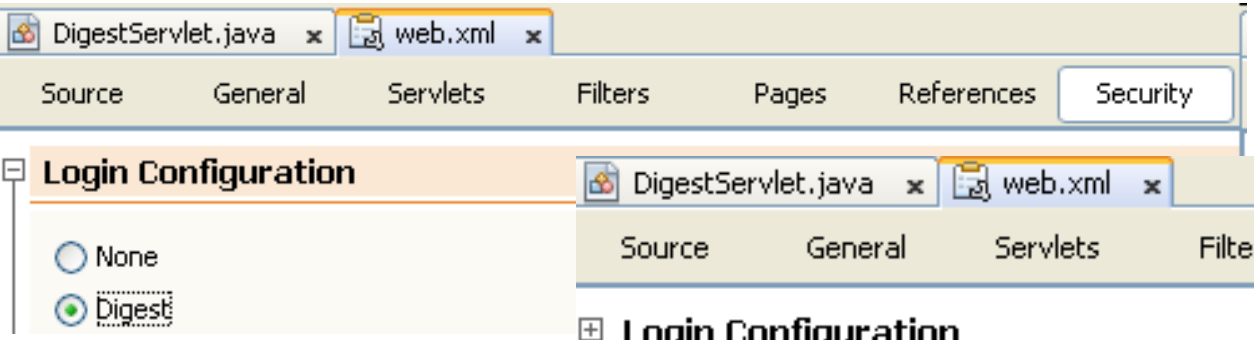
DigestServlet.java | web.xml

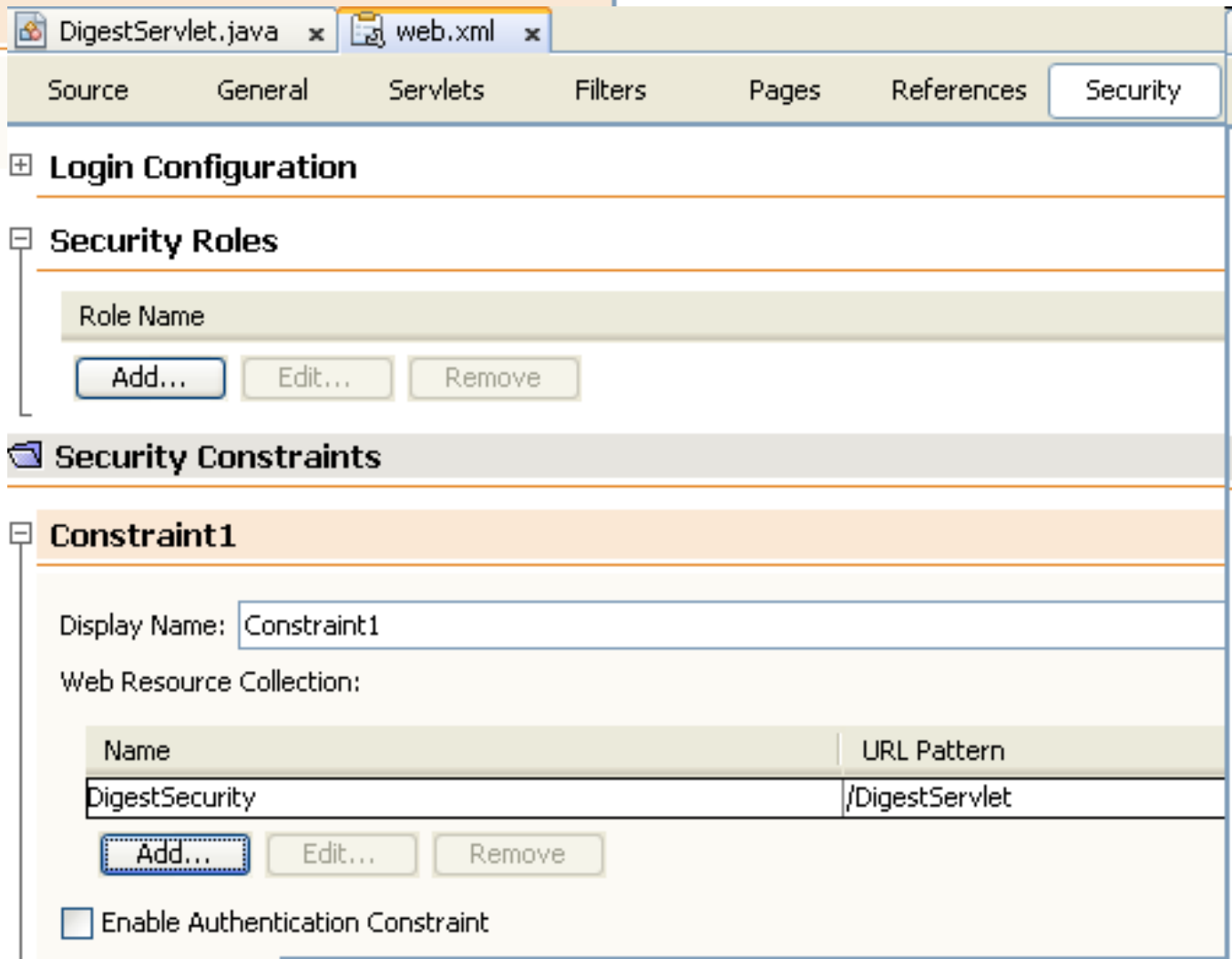
Source | History

```
42      protected void processRequest(HttpServletRequest request, HttpServletResponse r
43          response.setContentType("text/html;charset=UTF-8");
44          PrintWriter out = response.getWriter();
45          try {
46              String username = null;
47              boolean valid = false;
48              String authHeader = request.getHeader("Authorization");
49              if (authHeader != null) {
50                  StringTokenizer st = new StringTokenizer(authHeader);
51                  String basic = st.nextToken();
52                  if (basic.equalsIgnoreCase("Basic")) {
53                      String credential = st.nextToken();
54                      BASE64Decoder decoder = new BASE64Decoder();
55                      String userPass = new String(decoder.decodeBuffer(credential));
56                      int p = userPass.indexOf(":");
57                      username = userPass.substring(0, p);
58                      valid = users.containsKey(userPass)
59                              && users.get(userPass).equals("allowed"); }
60              }
61              if (!valid) {
62                  String s = "Basic realm=\"Servlet Users\"";
63                  response.setHeader("WWW-Authenticate", s);
64                  response.setStatus(401);
65              } else {
66                  out.println("<h3> Welcome " + username + "</h3>");
67                  out.println("You are authorized to process");
68                  return;
69              }
70              out.print("Invalid username or password!!!");
```

Authentication Types

DIGEST





Authentication Types

DIGEST

DigestServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Security | History

```
7   <servlet-mapping>
8       <servlet-name>DigestServlet</servlet-name>
9       <url-pattern>/DigestServlet</url-pattern>
10  </servlet-mapping>
11  <session-config>
12      <session-timeout>
13          30
14      </session-timeout>
15  </session-config>
16  <welcome-file-list>
17      <welcome-file>index.jsp</welcome-file>
18  </welcome-file-list>
19  <security-constraint>
20      <display-name>Constraint1</display-name>
21      <web-resource-collection>
22          <web-resource-name>DigestSecurity</web-resource-name>
23          <description/>
24          <url-pattern>/DigestServlet</url-pattern>
25      </web-resource-collection>
26  </security-constraint>
27  <login-config>
28      <auth-method>DIGEST</auth-method>
29  </login-config>
30 </web-app>
```

Authentication Types

FORM

- Associates a custom web page with the login process, as an **alternative to a browser dialog**
- There are only a **few rules**
  - The HTML form **must use the POST method**
  - The form must have "**j_security_check**" as its **action**
  - The form must include an input-capable **field for user called** "**j_username**"
  - The form must also include an input capable **field for password** called "**j_password**"
  - The form-based authentication is **required an error page**
- The **mechanism**
  - When the web page is required, the server **caches the URL** that tries to reach and **redirects to the form login pages**
  - The username and password is provided; assuming that the server is happy with these credentials, the required URL is passed
  - If the login fails, the server redirects to the error page

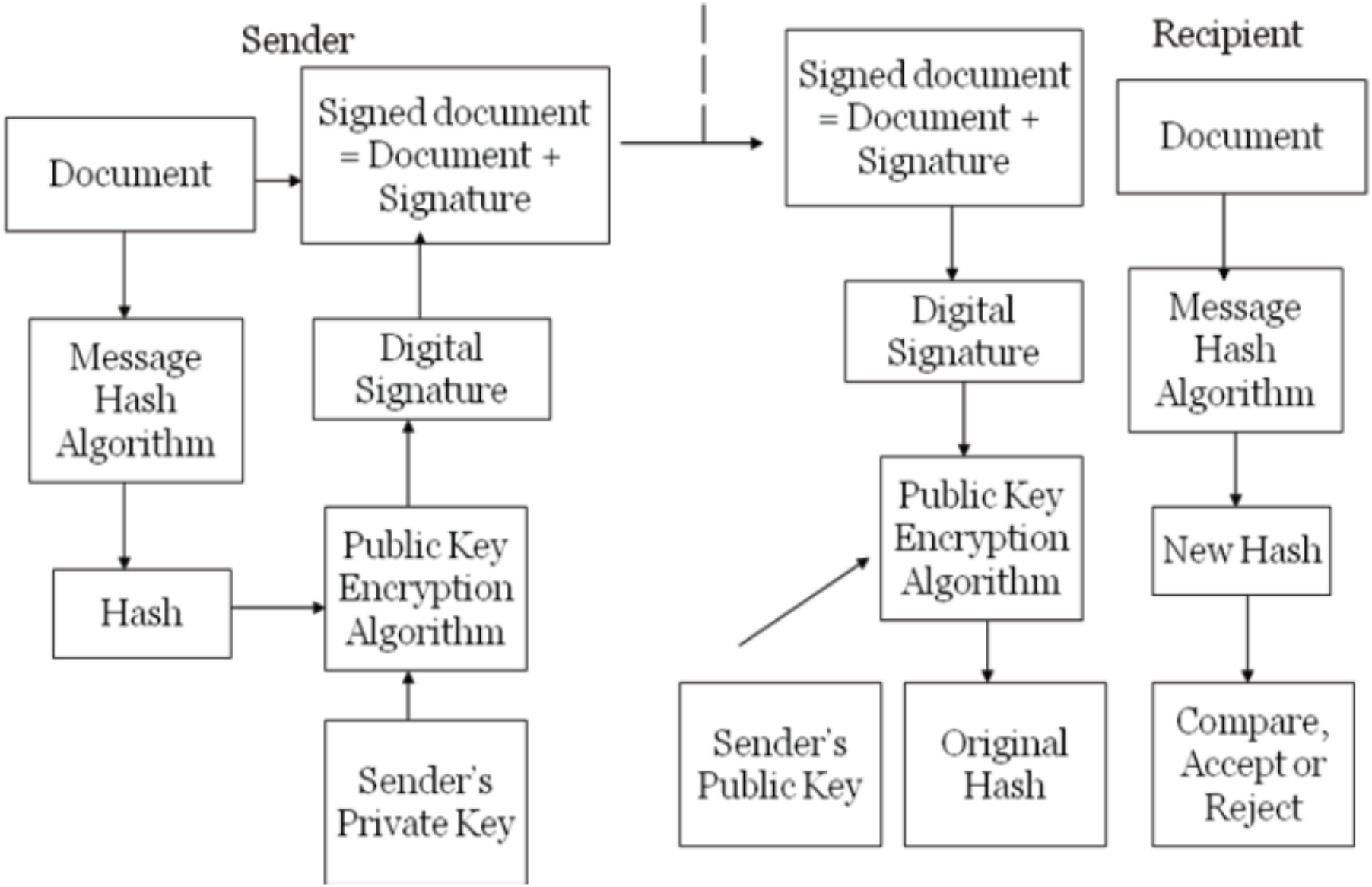
Authentication Types

CLIENT-CERT

- Uses **digital certificates** to achieve authentication
- Relies on **asymmetric keys** (public and private keys). Anything **encrypted with the public key can** be **decrypted with the private key and vice versa**
- The certificate is **generated** by **using specialized software** such as "**keytool**"
- **Mechanism**
  - The client, **first generates** a **private and public key**
  - The client **sends** the **public key** – and **other information to a third-party certificate authority (CA)**
  - The CA **binds** this **information and** the **client public key into** a **certificate**
  - The CA **adds** a **digital encrypted** with its private key, which makes a digest of information already in the certificate
  - The **certificate** is **returned to** the **client**, who installs it in his/her browser. When the **server requested authentication from client** browser, the **browser supplies** the **certificate to server** approving

Authentication Types

CLIENT-CERT – Mechanism



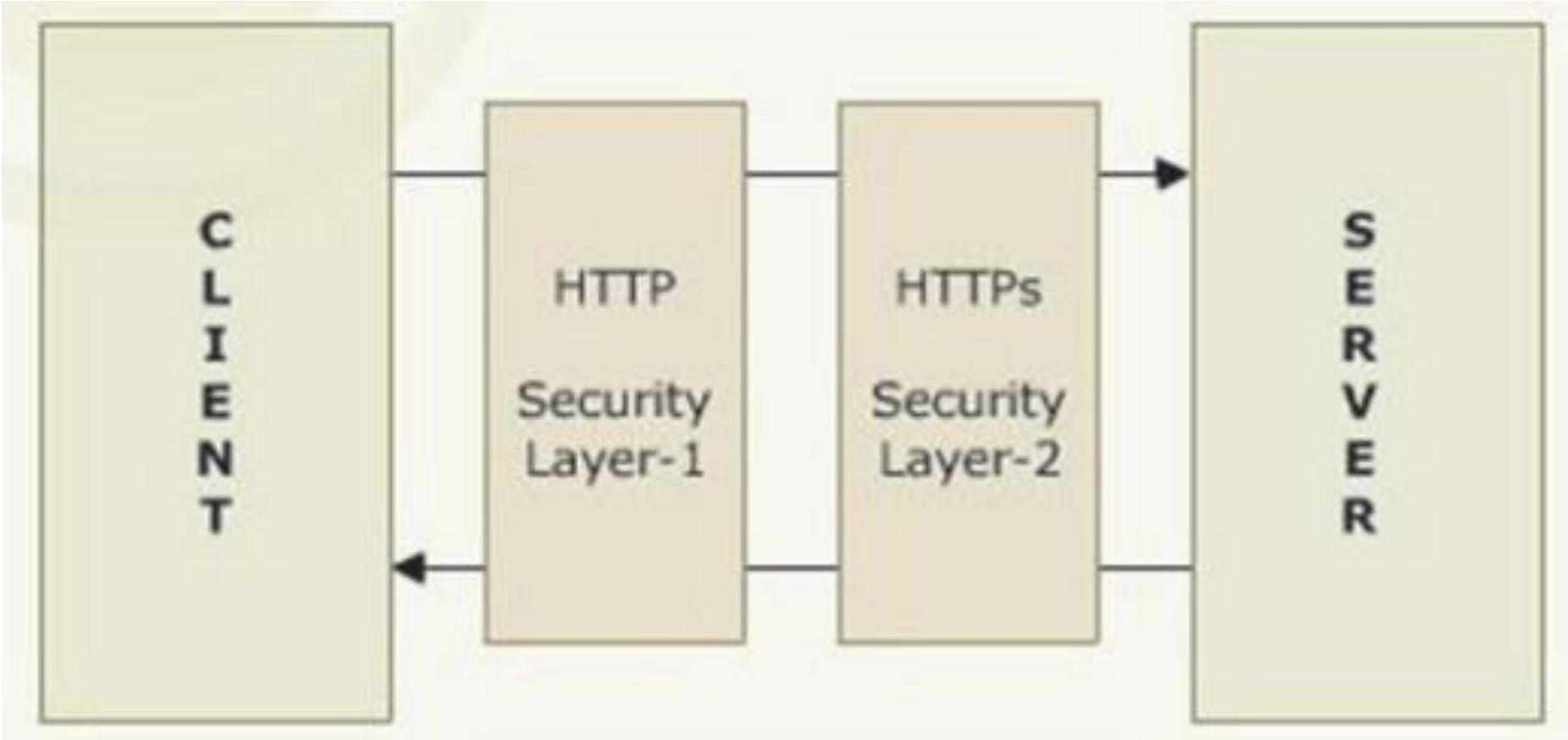
Authentication Types

HTTPS CLIENT

- Is a **secured client authentication** technique, which is based on **Public Key Certificates**.
- Authentication of users by **establishing a Secure Sockets Layer** (SSL) **connection** between sender and recipient
  - Sender – SSL Client
  - Recipient – SSL server
- **Extra authentication layer** in between HTTP and TCP
- This layer **confirms the client authentication**
- **Two kinds** of Certificated are used
  - **Server Certificates**: **Contain information** about **server** that **allows** a **client** to **identify** the **server before sharing sensitive information**
  - **Client Certificates**: **Contains personal information** about the **user** and **introduces** the **SSL client to the server**

Authentication Types

HTTPS CLIENT



Authentication Types

HTTPS CLIENT

- Steps in processing
  - **Step 1:** Generates Key (The key file (*.keystore) should be copied to **C:\Documents and Setting\username or C:\Users\username**
  - **Step 2:** Configuring the server configuration files at Web Server
  - **Step 3:** Configuring the application deployment descriptor file (web.xml)
  - **Step 4:** Restart the Web server and execute the application

Fpt University

Summary

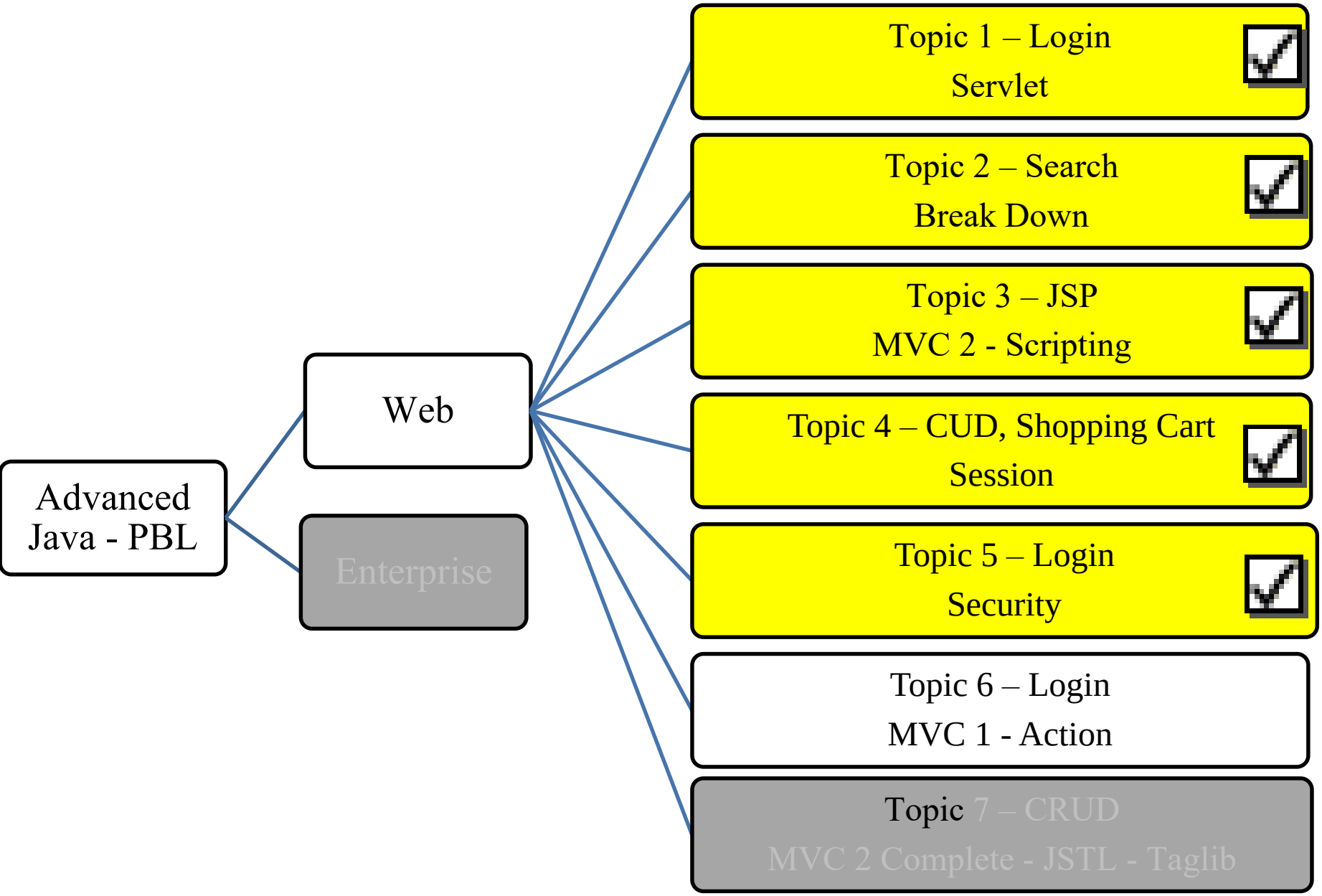
- **How to construct the security on the web site**
  - Authentication and Authorization with Basic, Digest, and Form
  - Confidentiality with HTTPS Client

Q&A

Next Lecture

- **JSP Standard Actions**
  - JavaBeans
  - Standard Actions
- **Dispatcher Mechanism**
  - Including, Forwarding, and Parameters
  - Vs. Dispatcher in Servlets
- **EL – Expression Languages**
  - What is EL?
  - How to use EL in JSP?

Next Lecture



Appendix – Authentication Types

BASIC

web.xml

Source | General | Servlets | Filters | Pages | References | Security

Login Configuration

- None
- Digest
- Client Certificate
- Basic
- Form

Form Login Page:

Form Error Page:

Realm Name:

Security Roles

Add Security Role

Role Name: manager-script

Description:

OK | Cancel

web.xml

Source | General | Servlets | Filters | Pages | References | Security

Login Configuration

Security Roles

Role Name

Add... | Edit... | Remove

Add

Security straints

Authentication Types

BASIC

web.xml

Source | General | Servlets | Filters | Pages | References | Security | History | Security Roles

Login Configuration

Security Roles

Role Name	Descrip
manager-script	
admin	

Add... | Edit... | Remove

Security Constraints

web.xml

General | Servlets | Filters | Pages | References | Security | XML | Security Constraints

Login Configuration

Security Roles

Security Constraints — Add Security Constraint

Authentication Types

BASIC

web.xml

General | Servlets | Filters | Pages | References | Security | XML | Constraint1

Login Configuration

Security Roles

Security Constraints — Add Security Constraint

Constraint1 — Remove

Display Name: Constraint1

Web Resource Collection:

Name	URL Pattern	HTTP Method

Add... | Edit... | Remove

Add Web Resource

Resource Name: BasicSecurity — Type resource name

Description:

URL Pattern(s): /admin/* — Type the URL /.../*

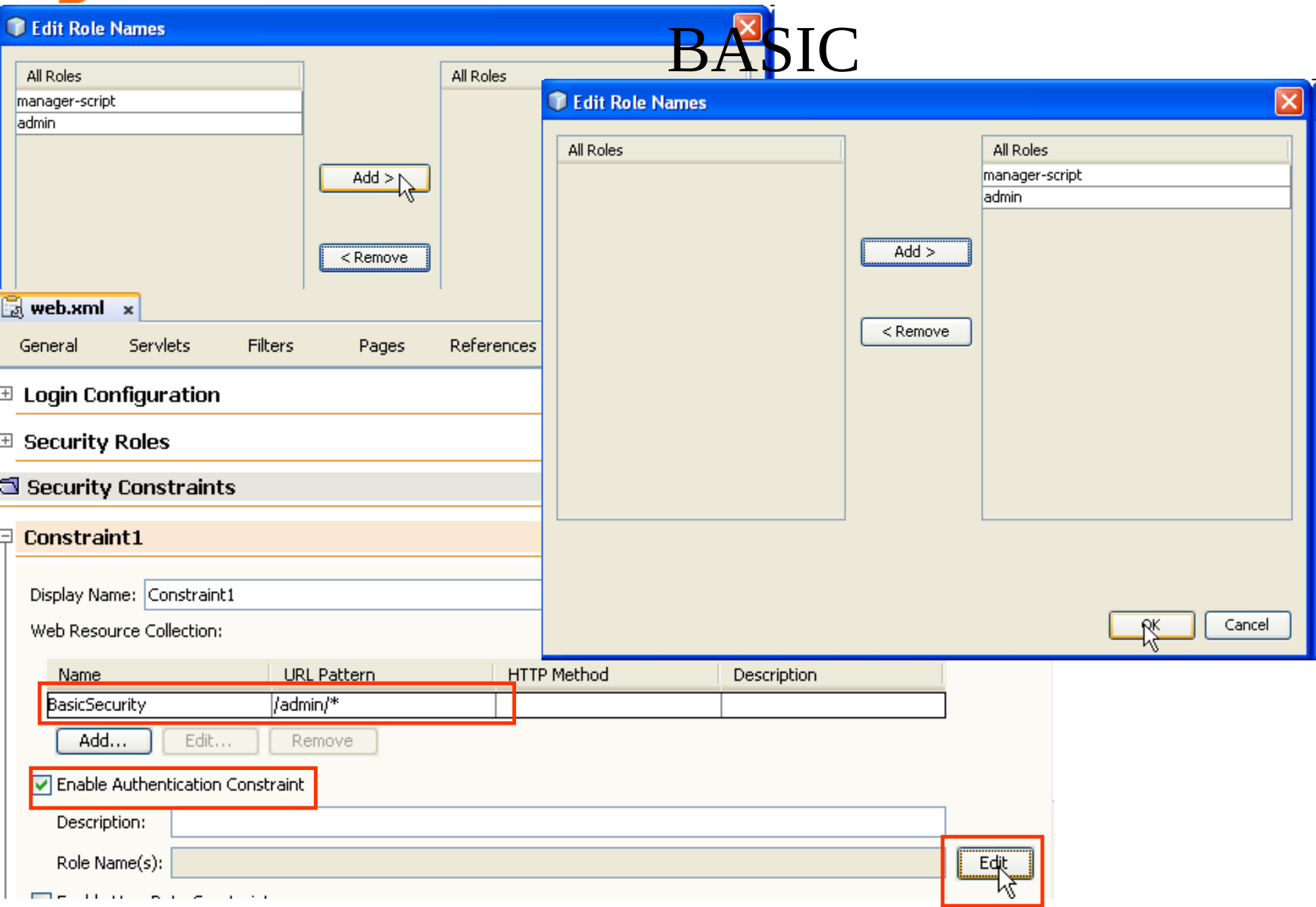
Use comma (,) to separate multiple patterns.

HTTP Method(s):
- All HTTP Methods
- Selected HTTP Methods
  - GET, POST, HEAD, PUT, OPTIONS, TRACE, DELETE

OK | Cancel

Authentication Types

BASIC



Authentication Types

BASIC

web.xml

Source | General | Servlets | Filters | Pages | References | Security | History | Constra

Login Configuration

Security Roles

Security Constraints — Add Security Constraint

Constraint1 — Remove

Display Name: Constraint1

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
BasicSecurity	/admin/*		

Add... | Edit... | Remove

☑ Enable Authentication Constraint

Description:

Role Name(s): manager-script, admin — Edit

☐ Enable User Data Constraint

Authentication Types

BASIC

web.xml

Source | General | Servlets | Filters | Pages | References | Security | History

```
10      </welcome-file-list>
11      <security-constraint>
12          <display-name>Constraint1</display-name>
13          <web-resource-collection>
14              <web-resource-name>BasicSecurity</web-resource-name>
15              <description/>
16              <url-pattern>/admin/*</url-pattern>
17          </web-resource-collection>
18          <auth-constraint>
19              <description/>
20              <role-name>manager-script</role-name>
21              <role-name>admin</role-name>
22          </auth-constraint>
23      </security-constraint>
24      <login-config>
25          <auth-method>BASIC</auth-method>
26      </login-config>
27      <security-role>
28          <description/>
29          <role-name>manager-script</role-name>
30      </security-role>
31      <security-role>
32          <description/>
33          <role-name>admin</role-name>
34      </security-role>
35  </web-app>
```

Authentication Types

BASIC

Basic - Windows Internet Explorer

http://localhost:8084/AJDay4_7Basic/

File Edit View Favorites Tools Help

Favorites Basic

Basic Security Demo

Click here to go to Admin Page

http://localhost:8084/AJDay4_7Basic/admin/index.html

Connect to localhost

The server localhost at Authentication require
username and password.

Warning: This server is requesting that your username and password be sent in an insecure manner (basic authentication without a secure connection).

User name:

Password:

Remember my password

OK Cancel

Invalid - Windows Internet Explorer

http://localhost:8084/AJDay4_7Basic/admin/index.html

File Edit View Favorites Tools Help

Favorites Invalid

Invalid username or password

Projects

- AJDay4_7Basic
  - Web Pages
    - META-INF
    - WEB-INF
    - admin
      - index.html
    - index.html
    - index.jsp
    - invalid.html
    - notPermission.html

Permission - Windows Internet Explorer

http://localhost:8084/AJDay4_7Basic/admin/index.html

File Edit View Favorites Tools Help

Favorites Permission

Cannot be accessed!!!

Admin - Windows Internet Explorer

http://localhost:8084/AJDay4_7Basic/admin/index.html

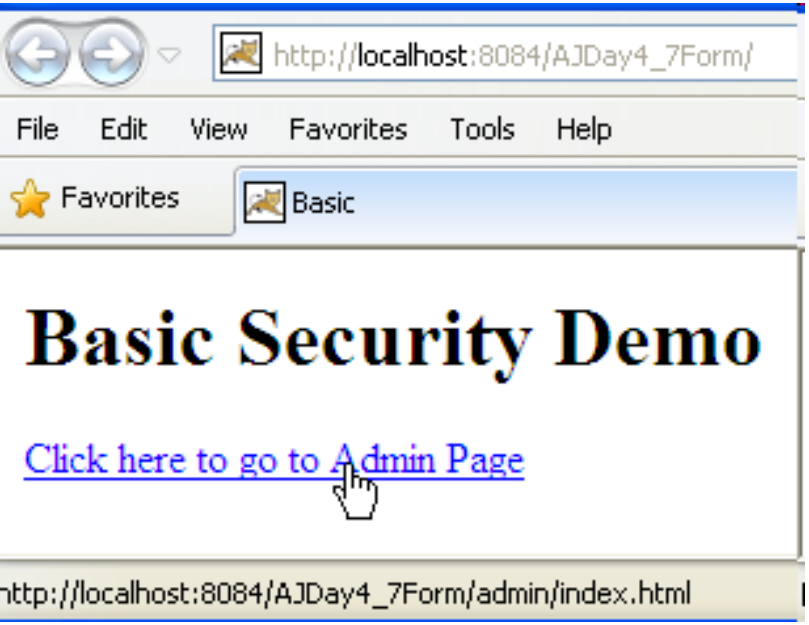
File Edit View Favorites Tools Help

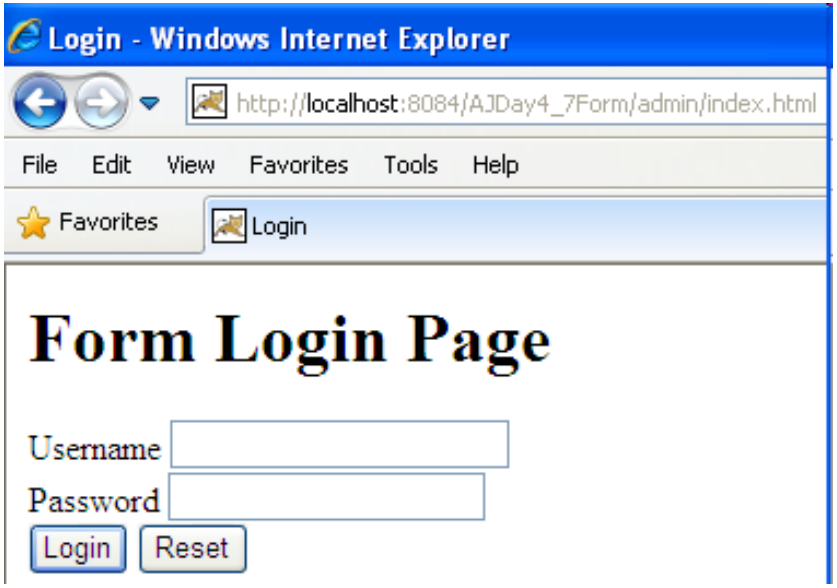
Favorites Admin

Welcome to Admin board

Appendix – Authentication Types

FORM – Example






```
<body>
    <h1>Form-based Authentication</h1>
    <form action="j_security_check" method="POST">
        Username <input type="text" name="j_username" value="" /><br/>
        Password <input type="password" name="j_password" value="" /><br/>
        <input type="submit" value="Login" />
    </form>
</body>
```

Authentication Types

FORM – Example

Invalid - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/j_security_check

File Edit View Favorites Tools Help

Favorites Invalid

Invalid username or password

Apache Tomcat/7.0.34 - Error report - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/index.html

File Edit View Favorites Tools Help

Favorites Apache Tomcat/7.0.34 - Error report

HTTP Status 403 - Access to the requested resource has been denied

type Status report

message Access to the requested resource has been denied

description Access to the specified resource has been forbidden.

Apache Tomcat/7.0.34

Admin - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/index.html

File Edit View Favorites Tools Help

Favorites Admin

Welcome to Admin board

Authentication Types

FORM – Example

web.xml

Source | General | Servlets | Filters | Pages | References | Security | History | Constraint1

Login Configuration

- None
- Digest
- Client Certificate
- Basic
- Form

Form Login Page: /admin/login.html [Browse...]

Form Error Page: /admin/invalid.html [Browse...]

Realm Name:

Security Roles

Role Name	Description
manager-script	
admin	

Add... | Edit... | Remove

Security Constraints [Add Security Constraint]

Authentication Types

FORM – Example

web.xml

Source | General | Servlets | Filters | Pages | References | Security | History | Constra

Login Configuration

Security Roles

Security Constraints — Add Security Constraint

Constraint1 — Remove

Display Name: Constraint1

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
FormSecurity	/admin/*		

Add... | Edit... | Remove

☑ Enable Authentication Constraint

Description:

Role Name(s): manager-script, admin — Edit

☐ Enable User Data Constraint

Authentication Types

FORM – Example

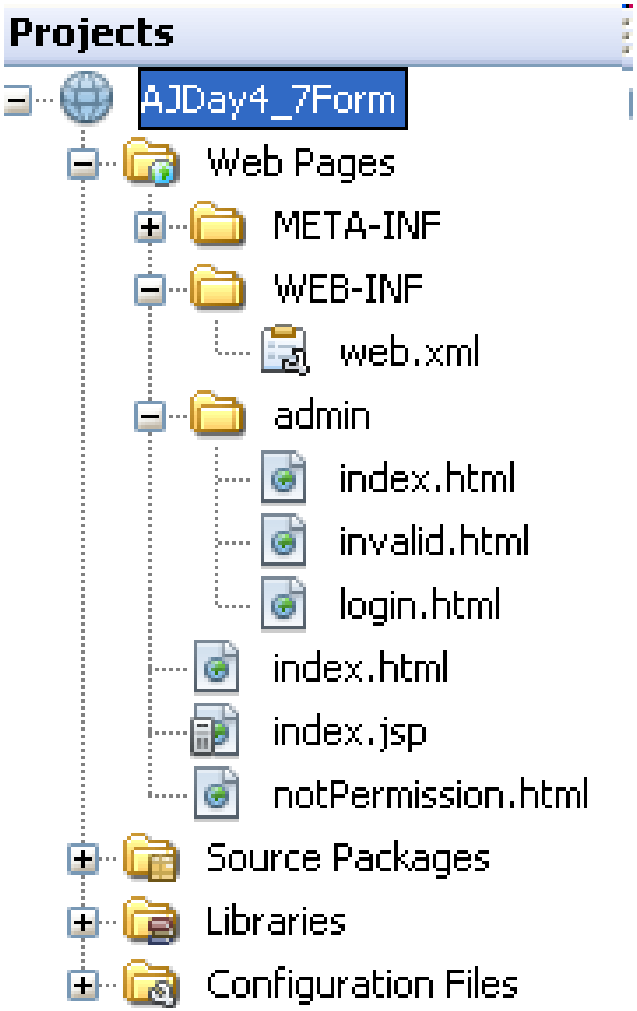
web.xml

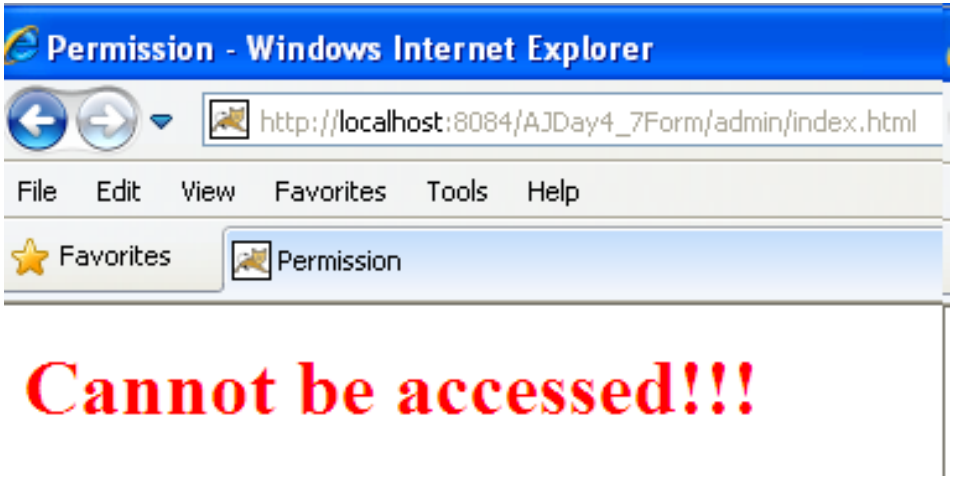
Source | General | Servlets | Filters | Pages | References | Security | History

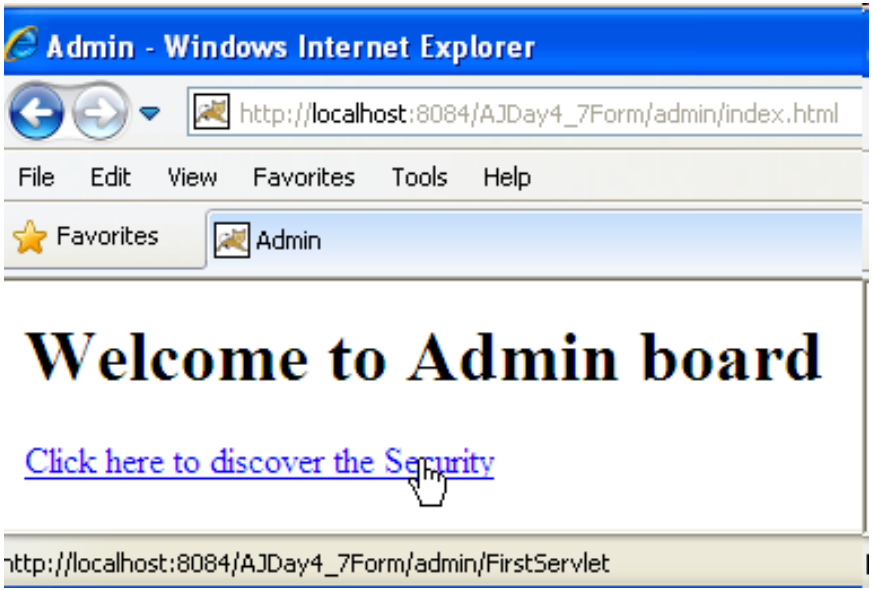
```
11      <security-constraint>
12          <display-name>Constraint1</display-name>
13          <web-resource-collection>
14              <web-resource-name>FormSecurity</web-resource-name>
15              <description/>
16              <url-pattern>/admin/*</url-pattern>
17          </web-resource-collection>
18          <auth-constraint>
19              <description/>
20              <role-name>manager-script</role-name>
21              <role-name>admin</role-name>
22          </auth-constraint>
23      </security-constraint>
24      <login-config>
25          <auth-method>FORM</auth-method>
26          <form-login-config>
27              <form-login-page>/admin/login.html</form-login-page>
28              <form-error-page>/admin/invalid.html</form-error-page>
29          </form-login-config>
30      </login-config>
31      <security-role>
32          <description/>
33          <role-name>manager-script</role-name>
34      </security-role>
35      <security-role>
36          <description/>
37          <role-name>admin</role-name>
38      </security-role>
39  </web-app>
```

Authentication Types

FORM – Example







Authentication Types

FORM – Example

Apache Tomcat/7.0.34 - Error report - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/FirstServlet

File Edit View Favorites Tools Help

Favorites | Apache Tomcat/7.0.34 - Error report

HTTP Status 404 - /AJDay4_7Form/admin/FirstServlet

type Status report

message /AJDay4_7Form/admin/FirstServlet

description The requested resource is not available.

Apache Tomcat/7.0.34

Servlet - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/FirstServlet

File Edit View Favorites Tools Help

Favorites | Servlet

The Security Secret

Welcome, khanhkt

Admin: true

Authentication Types

FORM – Example

index.html

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Admin</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <h1>Welcome to Admin board</h1>

        <a href="FirstServlet">Click here to discover the Security</a>
    </body>
</html>
```

web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/n
    <servlet>
        <servlet-name>FirstServlet</servlet-name>
        <servlet-class>sample.servlet.FirstServlet</ser
    </servlet>
    <servlet-mapping>
        <servlet-name>FirstServlet</servlet-name>
        <url-pattern>/admin/FirstServlet</url-pattern>
    </servlet-mapping>
```

FirstServlet.java

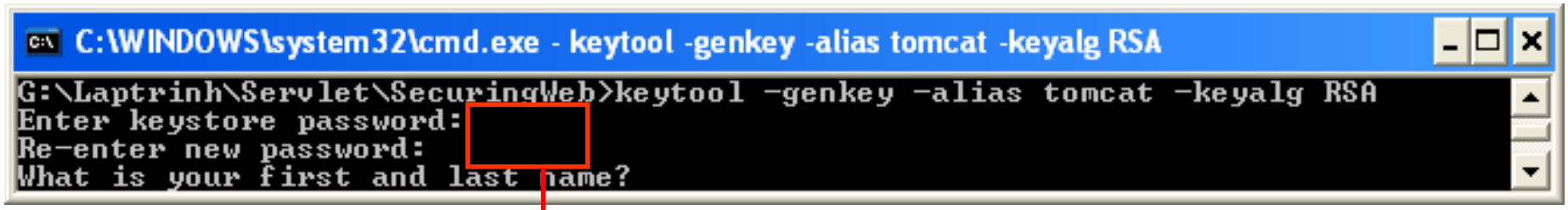
```java
 * @author Trong Khanh

class FirstServlet extends HttpServlet {
...*/
tected void processRequest(HttpServletRequest request, HttpServ
    throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
try {
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>The Security Secret</h1>");
    out.print("Welcome, " + request.getRemoteUser());
    out.print("<br/>Admin: " + request.isUserInRole("admin"));
    out.println("</body>");
    out.println("</html>");
} finally {
    out.close();
}
}
```

Appendix – Authentication Types

HTTPS CLIENT

- **Step 1:** Generates Key
  - Using the key generator tool creates a keystore to store the keys generated as keytool –genkey –alias **privatekeyName** –keyalg **EncodingAlgorithms**



Password and re-enter password can not be seen when typing (high security)

Authentication Types

HTTPS CLIENT

- **Step 1:** Generates Key (cont)

C:\WINDOWS\system32\cmd.exe - keytool -genkey -alias tomcat -keyalg RSA

```
C:\Documents and Settings\Trong Khanh>keytool -genkey -alias tomcat -keyalg RSA
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Khanh Kieu
What is the name of your organizational unit?
  [Unknown]:  Home
What is the name of your organization?
  [Unknown]:  MyHome
What is the name of your City or Locality?
  [Unknown]:  HCM
What is the name of your State or Province?
  [Unknown]:  South
What is the two-letter country code for this unit?
  [Unknown]:  vi
Is CN=Khanh Kieu, OU=Home, O=MyHome, L=HCM, ST=South, C=vi correct?
  [no]:
```

Authentication Types

HTTPS CLIENT

- **Step 1:** Generates Key (cont)

```
C:\WINDOWS\system32\cmd.exe - keytool -genkey -alias tomcat -keyalg RSA
Is CN=Khanh Kieu, OU=Home, O=MyHome, L=HCM, ST=South, C=vi correct?
  [no]:  yes

Enter key password for <tomcat>
        <RETURN if same as keystore password>:  _
```

Should be pressed Enter

- The .keystore file is created at the directory
  - **C:\Documents and Setting\username\.keystore**
  - **C:\Users\username\.keystore**

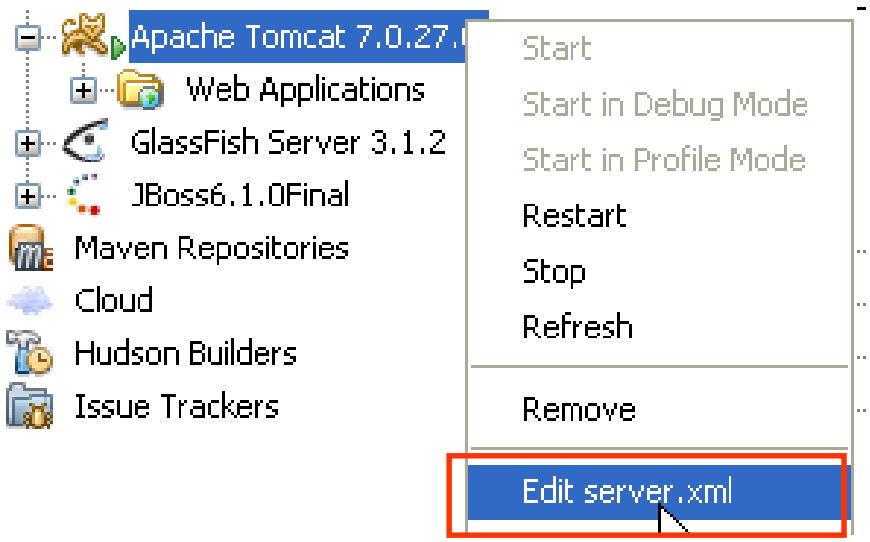
c:\Documents and Settings\Trong Khanh*.*

Name	Ext	Size
.keystore		2.220

Authentication Types

HTTPS CLIENT – Step 2

- Go to the
  - **C:\Documents and Settings\LoggedUser\Application Data\NetBeans\ 7.4\apache-tomcat-7.0.41.0_base\conf\ server.xml**
    - **C:\Users\LoggedUser\AppData\Roaming\NetBeans\7.4\ apache-tomcat-7.0.41.0_base\conf\server.xml**
  - **Or right click on the Server in tab Runtime, click edit server.xml**



Authentication Types

HTTPS CLIENT – Step 2

- Uncomment the <!-- … --> as

server.xml

Source History

```
74      <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
75                 maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
76                 clientAuth="false" sslProtocol="TLS" keystorePass="trongkhanh" />
```

Keystore password

Search key SSL HTTP/1.1 on the server.xml file

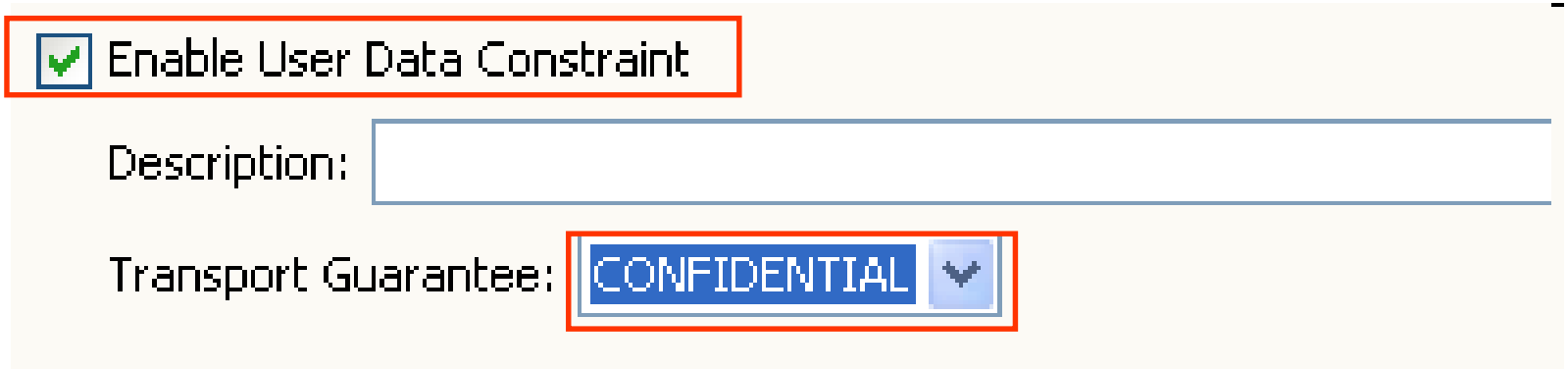
Output

Apache Tomcat 7.0.27.0 | Apache Tomcat 7.0.27.0 Log | AJDay4_7Basic (run-deploy)

```
thg 10 30, 2013 8:42:58 SA org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-apr-8084"]
thg 10 30, 2013 8:42:58 SA org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8443"]
thg 10 30, 2013 8:42:58 SA org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-apr-8009"]
thg 10 30, 2013 8:42:58 SA org.apache.catalina.startup.Catalina start
INFO: Server startup in 419 ms
```

Authentication Types

HTTPS CLIENT – Step 3

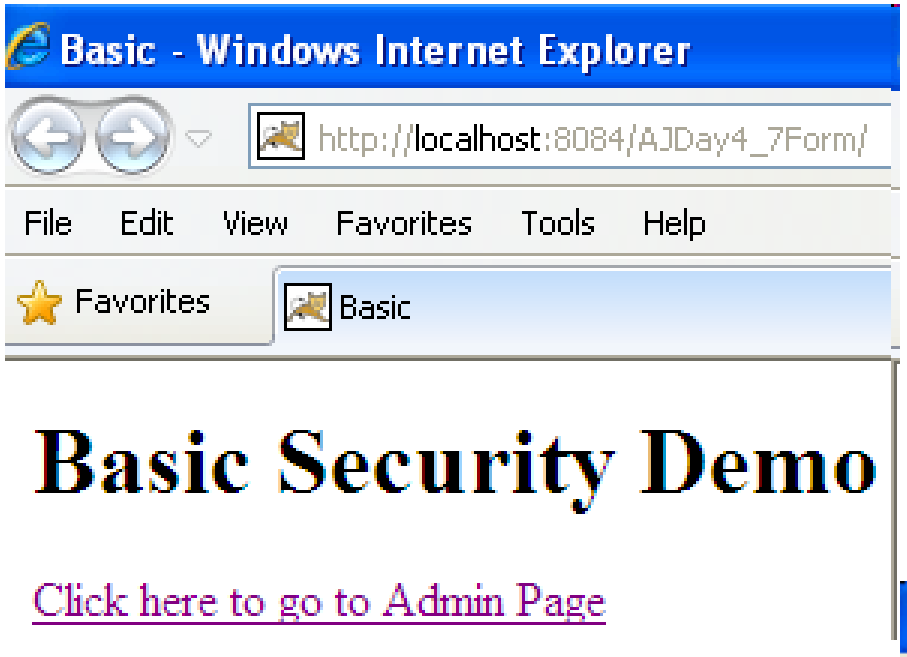


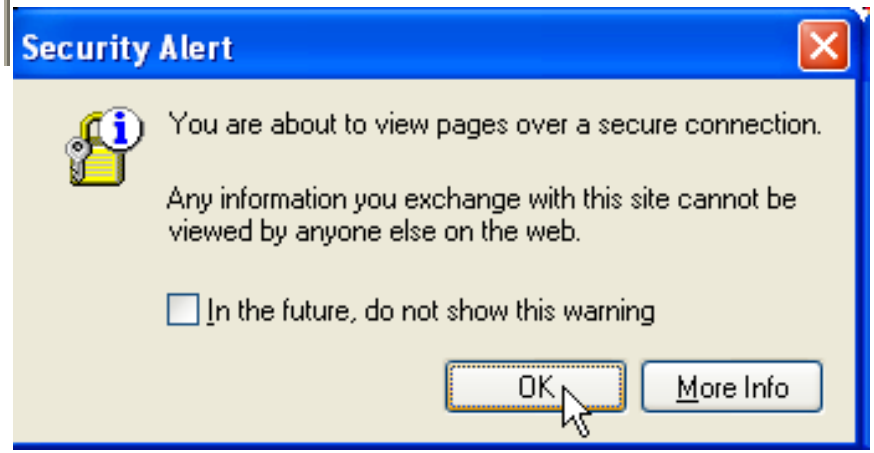

```
<security-constraint>
    <display-name>Constraint1</display-name>
    <web-resource-collection>
    <auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<login-config>
```

Authentication Types

HTTPS CLIENT – Step 4

- Restart the Web server and execute the application





Authentication Types

HTTPS CLIENT – Step 4

Certificate Error: Navigation Blocked - Windows Internet Explorer

http://localhost:8084/AJDay4_7Form/admin/index.html

File Edit View Favorites Tools Help

Favorites | Certificate Error: Navigation Blocked

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.
The security certificate presented by this website was issued for a different website's address.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

More information

http://localhost:8084/AJDay4_7Form/admin/index.html

Security Alert

You are about to view pages over a secure connection.

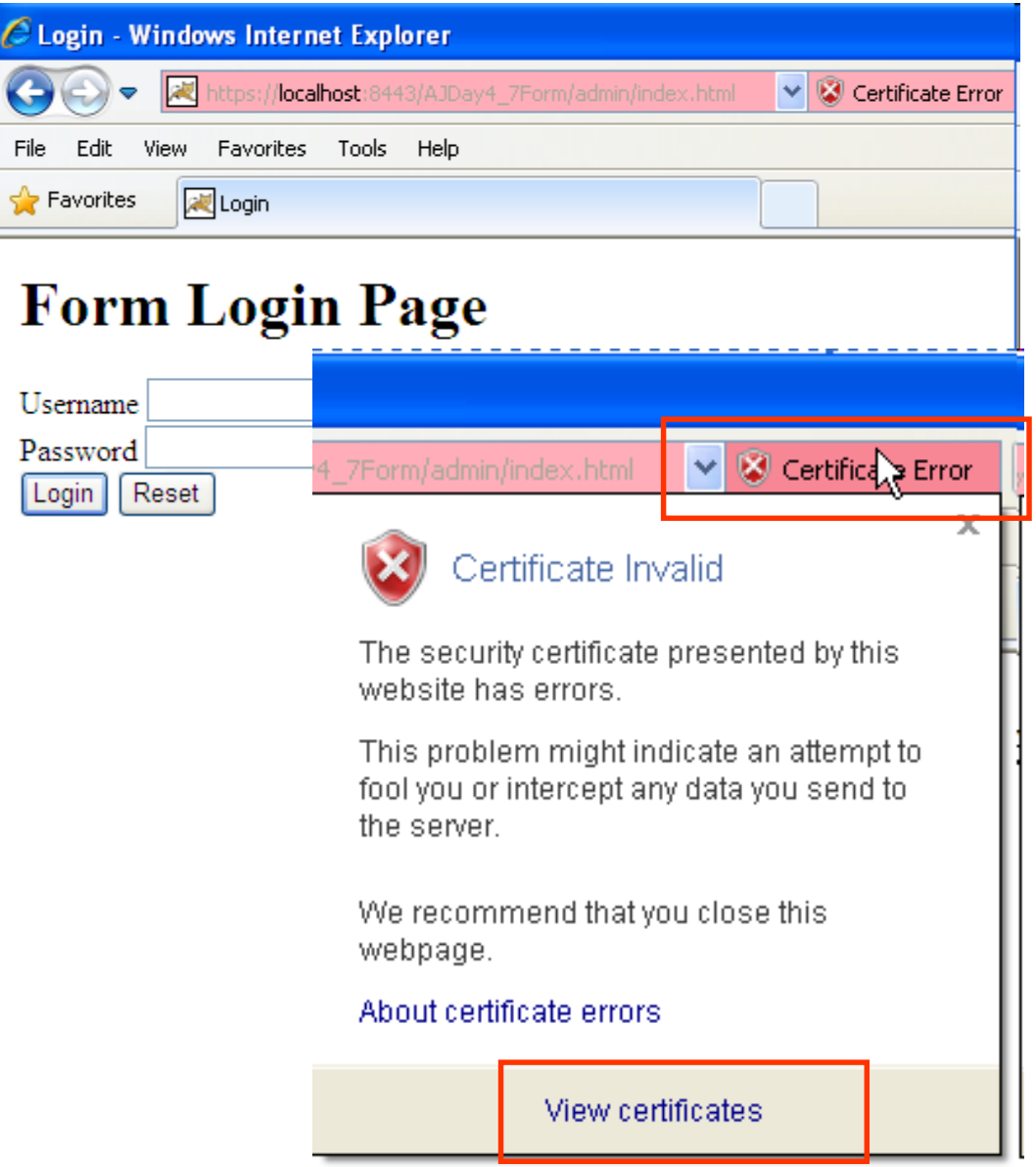
Any information you exchange with this site cannot be viewed by anyone else on the web.

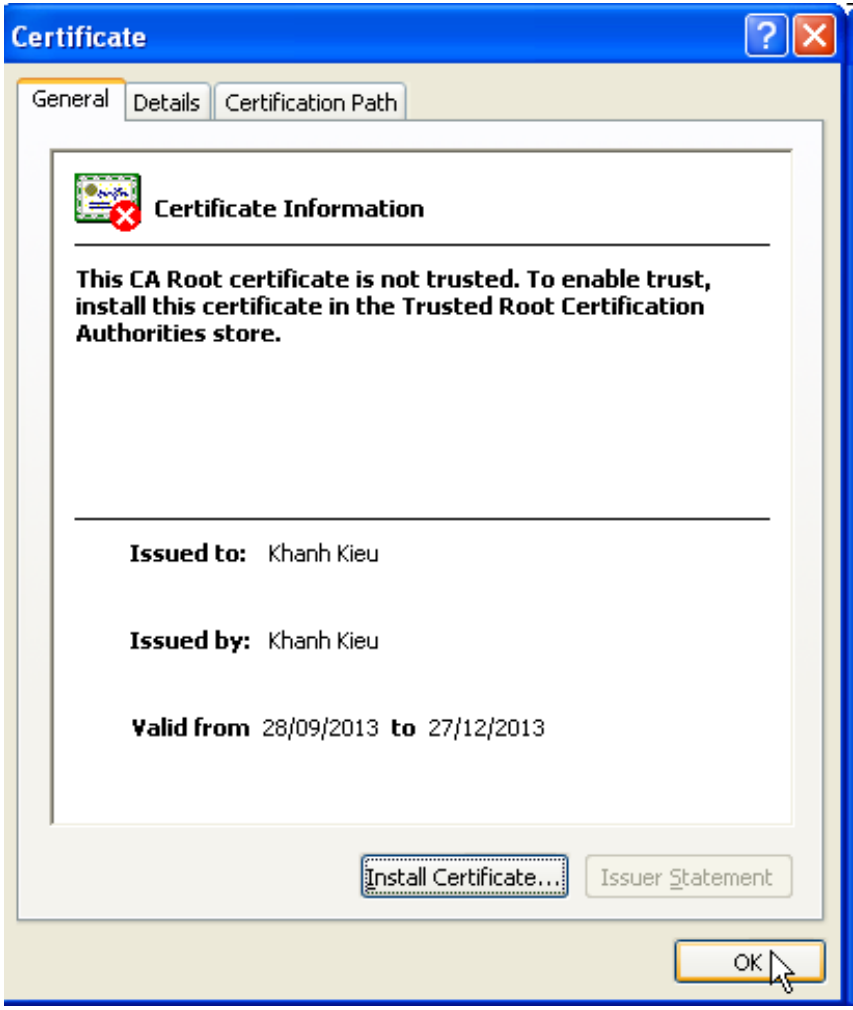
In the future, do not show this warning

OK | More Info

Authentication Types

HTTPS CLIENT – Step 4

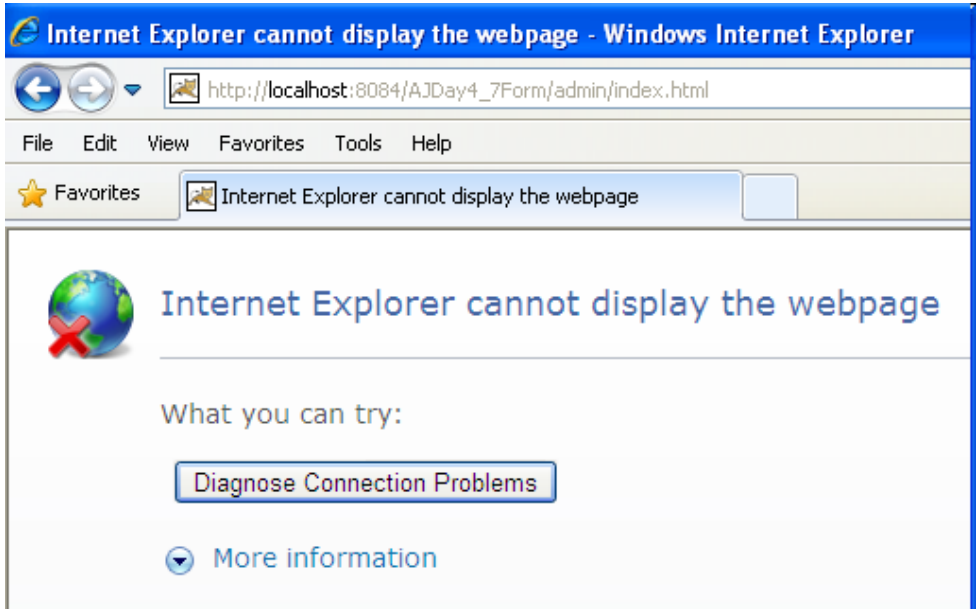


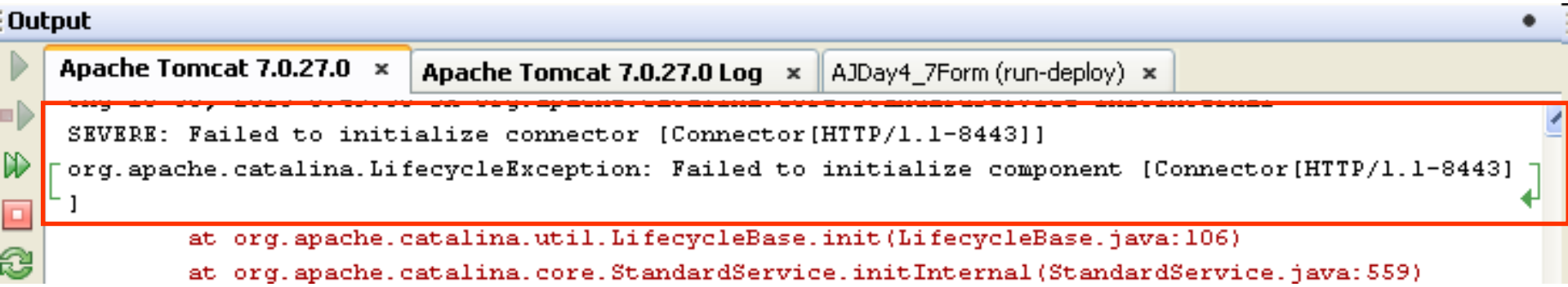


Authentication Types

HTTPS CLIENT – Step 4

- Modify one of following contents to check application
  - *Comment above configuration contents on server.xml file*
  - *Restart server*
  - *Running the application again*
  - Delete the .keystore file
  - Restart Server





Appendix

JDBC Realms

- Is an **implementation of a tomcat Realm** that use a set of **configurable tables inside a RDMS** to store user's data, this tables are accessed by means of standard JDBC drivers
- **Step 1**: Creating DB on RDBMS
- **Step 2**: Config JDBCRealm on Tomcat

Appendix

Creating the table on RDBMS

Table - dbo.users | Summary

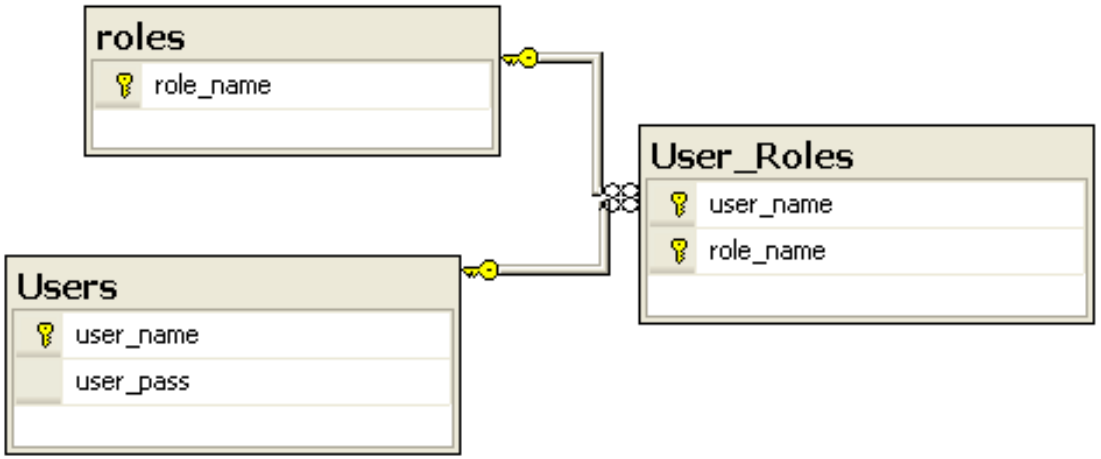
Column Name	Data Type	Allow Nulls
user_name	varchar(15)	☐
user_pass	varchar(15)	☐

Table - dbo.roles | Summary

Column Name	Data Type	Allow Nulls
role_name	varchar(15)	☐
		☐

Table - dbo.user_roles | Summary

Column Name	Data Type	Allow Nulls
user_name	varchar(15)	☐
role_name	varchar(15)	☐



KHANHKIEU\SQL...K8 - dbo.users

user_name	user_pass
ide	2mdXj4dj
khanh	trongkhanh
tomcat	tomcat
users	guest
NULL	NULL

KHANHKIEU\SQL...K8 - dbo.roles

role_name
admin
guest
manager-script
tomcat
NULL

KHANHKIEU\SQL...bo.user_roles

user_name	role_name
ide	admin
ide	manager-script
khanh	admin
khanh	manager-script
tomcat	guest
users	tomcat
NULL	NULL

Appendix

Configuring JDBCRealm on Tomcat

- Go the context.xml to modify the following content

context.xml

Source History

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/FormSecuritySE0964">
    <Realm className="org.apache.catalina.realm.JDBCRealm"
        driverName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
        connectionURL="jdbc:sqlserver://localhost:1433;instanceName=SQL2008;databaseName=Sinhvien2K8"
        connectionName="sa" connectionPassword="trongkhanh"
        userTable="users" userNameCol="user_name" userCredCol="user_pass"
        userRoleTable="user_roles" roleNameCol="role_name" />
</Context>
```

•Notes: Before the server is started, the DB driver (sqljdbc4.jar) must be located at c:\Program Files\Apache Software Foundation\Apache Tomcat 8.x.x\lib

Appendix

Declarative Security

- Provides security to resource with the help of the **server configuration**
- **Works** as a **different layer from the web component** which it works.
- **Advantages**
  - **Gives scope to the programmer** to **ignore** the **constraints** of the programming **environment**
  - **Updating the mechanism** does not require total change in Security model
  - It is easily maintainable
- **Limitation**
  - Access is **provided to all or denied**
  - Access is **provided by the Server only if the password matches**
  - All the **pages use same authentication mechanism**
  - It can **not use both form-based and basic authentication for different page**

Appendix

Declarative Security Implementation

- Setting up User Names, Passwords, Roles
- Setting Authentication mechanism to Authentication Type
  - Creating Login Page and Error Page with Form-based authentication
  - Defining all the deployment descriptor security declaration
- Specify URLs that should be password protected
- Specify URLs that Should be available only with SSL (if necessary)

Appendix

Programmatic Security

- Authenticates users and grant access to the users
- **Servlet** either **authenticates** the **user** or **verify** that the **user has authenticates earlier**
- There are **2 types:**
  - **Hard vs. Soft** (combining between code and declaration)
- **Advantages**
  - Ensue total portability
  - Allowed **password matching strategies**
- **Limitation**
  - Much **harder to code and maintain**
  - Every **resource must use the code**
- **Implementation**
  - Check whether there is an authorization request header (**checking header**)
  - Get the String, which contains the encoded user name / password
  - Reverse the base64 encoding of the user name / password String (**digest auth**)
  - Check the user name and password (**login form with web server or DB**)
  - If authentication fails, send the proper response to the client (**error handling**)

Appendix

Programmatic Security

- Some **method** supporting from HttpServletRequest

Methods	Descriptions
getAuthType	- **public string getAuthType()** - returns the authentication scheme name
getHeader	- Reference details in previous topic
getRemoteUser	- **public String getRemoteUser()** - It the user is authenticated it returns the login name of the user else it returns null.
isUserInRole	- **public boolean isUserInRole(String role)** - Returns a Boolean value, which indicates whether the authenticated user is included in the logical “role”.
getUserPrincipal	- **public Principal getUserPrincipal()** - Returns a java.security.Principal object

Appendix

Programmatic Security

- **Securing on Servlet**

```
<servlet>
    <servlet-name>servletName</servlet-name>
    <servlet-class>servletClass</servlet-class>
    <security-role-ref>
        <description/>
        <role-name>aliasName_map_rolelink</role-name>
        <role-link>realRoleName</role-link>
    </security-role-ref>
</servlet>
```

Appendix

Programmatic Security – Example

AuthServlet.java

```
28      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
29      throws ServletException, IOException {
30          response.setContentType("text/html;charset=UTF-8");
31          PrintWriter out = response.getWriter();
32          try {
33              out.println("<html>");
34              out.println("<head>");
35              out.println("<title>Programmatically Security</title>");
36              out.println("</head>");
37              out.println("<body>");
38              out.println("<h1>Programmatically Security Demo</h1>");
39
40              String user = request.getUserPrincipal().getName();
41              String userName = request.getRemoteUser();
42
43              out.println("<h1>The accessed user is: " + user + "</h1>");
44              out.println("<h2>The username is: " + userName + "</h2>");
45              out.println("The Auth Type: " + request.getAuthType()+ "</h3>");
46
47              if(request.isUserInRole("MGR")){
48                  out.println("<h2>Are you a manager???</h2>");
49              } else if(request.isUserInRole("user")){
50                  out.println("<h2>Are you a user???</h2>");
51              } else if (request.isUserInRole("guest")){
52                  out.println("<h2>Are you a guest???</h2>");
53              }
54              out.println("</body>");
55              out.println("</html>");
56          } finally {
```

Appendix

Programmatic Security – Example

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Security

Login Configuration

- None
- Digest
- Client Certificate
- Basic
- Form

Form Login Page:

Form Error Page:

Realm Name:

Security Roles

Role Name
manager-script

Add... | Edit

AuthServlet.java | web.xm

Source | General | Servlet

Login Configuration

Security Roles

Security Constraints

Constraint1

Display Name: Constraint1

Web Resource Collection:

Name
Programmatically

Add... | Edit...

Enable Authentication Constraint

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Security

```
16      <welcome-file-list>
17          <welcome-file>AuthServlet</welcome-file>
18      </welcome-file-list>
19      <security-constraint>
20          <display-name>Constraint1</display-name>
21          <web-resource-collection>
22              <web-resource-name>Programmatically</web-res
23              <description/>
24              <url-pattern>/index.jsp</url-pattern>
25          </web-resource-collection>
26          <auth-constraint>
27              <description/>
28              <role-name>*</role-name>
29          </auth-constraint>
30      </security-constraint>
31      <login-config>
32          <auth-method>BASIC</auth-method>
33      </login-config>
34      <security-role>
35          <description/>
36          <role-name>manager-script</role-name>
37      </security-role>
38  </web-app>
```

Appendix

Programmatic Security – Example

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Security | History

Servlets

AuthServlet -> /AuthServlet

Servlet Name: AuthServlet

Description:

Servlet Class: sample.servlet.AuthServlet [Browse...]

JSP File: [Browse...]

URL Pattern(s): /AuthServlet

Use comma (,) to separate multiple patterns.

Initialization Parameters:

Parameter Name	Parameter Value	Descriptio

Add... | Edit... | Remove

Security Role References:

Role Ref Name	Role Ref Link	Description

Add... | Edit... | Remove

Add

Run As:

Add Security Role Reference

Role Ref Name: MGR

Role Ref Link: manager-script

Description:

OK | Cancel

Appendix

Programmatic Security – Example

- Applied the Form Security to this Application, then run directly the AuthServlet and the web.xml does not secure the AuthServlet
  - The errors occur at the server consoles.

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Securi

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <web-app version="2.5" xmlns="http://java.sun.com/xm
3       <servlet>
4           <servlet-name>AuthServlet</servlet-name>
5           <servlet-class>sample.servlet.AuthServlet</s
6           <security-role-ref>
7               <description/>
8               <role-name>MGR</role-name>
9               <role-link>manager-script</role-link>
10          </security-role-ref>
11      </servlet>
12      <servlet-mapping>
13          <servlet-name>AuthServlet</servlet-name>
14          <url-pattern>/AuthServlet</url-pattern>
15      </servlet-mapping>
16      <session-config>
```

- Errors occur because the security is not applied to AuthServlet resource

```
thg 9 28, 2013 8:40:42 CH org.apache.catalina.core.StandardWrapperValve invoke
SEVERE: Servlet.service() for servlet [AuthServlet] in context with path [/AJDay4_7Programmatic] threw exception
java.lang.NullPointerException
        at sample.servlet.AuthServlet.processRequest(AuthServlet.java:44)
        at sample.servlet.AuthServlet.doGet(AuthServlet.java:79)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:621)
```

Appendix

Programmatic Security – Example

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Security | History

```
24  <security-constraint>
25      <display-name>Constraint1</display-name>
26      <web-resource-collection>
27          <web-resource-name>Programmatically</web-resource-name>
28          <description/>
29          <url-pattern>/AuthServlet</url-pattern>
30      </web-resource-collection>
31      <auth-constraint>
```

dows Internet Explorer

http://localhost:8084/AJDay4_7Programmatic/

File Edit View Favorites Tools Help

Favorites | Programmatically Security

Programmatically Security Demo

The accessed user is: khanhkt

The username is: khanhkt

The Auth Type: BASIC

Are you a manager???

Appendix

Programmatic Security – Example

```
if(request.isUserInRole("manager")){
```

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Security | History

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" x
3       <servlet>
4           <servlet-name>AuthServlet</servlet-name>
5           <servlet-class>sample.servlet.AuthServlet</servlet-class>
6           <security-role-ref>
7               <description/>
8               <role-name>MGR</role-name>
9               <role-link>*</role-link>
10          </security-role-ref>
11      </servlet>
```

Appendix

Programmatic Security – Example

```
if(request.isUserInRole("manager")){
```

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <web-app version="2.5" xmlns="http://java.sun
3       <servlet>
4           <servlet-name>AuthServlet</servlet-nar
5           <servlet-class>sample.servlet.AuthServ
6           <security-role-ref>
7               <description/>
8               <role-name>MGR</role-name>
9               <role-link>*</role-link>
10          </security-role-ref>
11      </servlet>
```


```
INFO: Reloading Context with name [/AJDay4_7Programmatic] is completed
thg 9 28, 2013 8:47:13 CH org.apache.catalina.startup.ContextConfig validateSecurityRoles
INFO: WARNING: Security role name * used in a <role-link> without being defined in a <security-role>
```

Appendix

Programmatic Security – Example

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Se

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <web-app version="2.5" xmlns="http://java.sun.com/
3       <servlet>
4           <servlet-name>AuthServlet</servlet-name>
5           <servlet-class>sample.servlet.AuthServlet<
6           <security-role-ref>
7               <description/>
8               <role-name>MGR</role-name>
9               <role-link>manager-script</role-link>
10          </security-role-ref>
```

Only user with manager-script role can access the protected web resources. Others users' role will get 403 error

Apache Tomcat/7.0.34 - Error report - Windows Internet Explorer

http://localhost:8084/AJDay4_7Programmatic/

File Edit View Favorites Tools Help

Favorites | Apache Tomcat/7.0.34 - Error report

HTTP Status 403 - Access to the requested resource has been denied

type Status report

message Access to the requested resource has been denied

description Access to the specified resource has been forbidden.

Apache Tomcat/7.0.34

Appendix

Programmatic Security – Example

AuthServlet.java | web.xml

Source | General | Servlets | Filters | Pages | References | Securi

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <web-app version="2.5" xmlns="http://java.sun.com/xm
3       <servlet>
4           <servlet-name>AuthServlet</servlet-name>
5           <servlet-class>sample.servlet.AuthServlet</s
6           <security-role-ref>
7               <description/>
8               <role-name>MGR</role-name>
9               <role-link>*</role-link>
10          </security-role-ref>
11      </servlet>
12      <servlet-mapping>
16      <session-config>
21      <welcome-file-list>
24      <security-constraint>
25          <display-name>Constraint1</display-name>
26          <web-resource-collection>
31          <auth-constraint>
32              <description/>
33              <role-name>manager-script</role-name>
34          </auth-constraint>
35      </security-constraint>
36      <login-config>
39      <security-role>
    </web-app>
```

- Same result with previous slides because only manage role can access
- Same result with checking MGR in servlet code