

# **Wykład 4**

## **Metody uwierzytelniania - Bezpieczeństwo (3)**

**wg „The Java EE 5 Tutorial ”**

**Autor: Zofia Kruczkiewicz**

# Struktura wykładu

1. **Protokół SSL do zabezpieczenia aplikacji na poziomie protokołu transportowego http - https**
2. **Weryfikacja protokołu HTTPS obsługiwanego przez Serwera Aplikacji Java EE 5.0**
3. **Ustawienie protokołu https do przesyłania danych w aplikacji**
4. **Uruchomienie aplikacji - zastosowanie protokołu HTTPS**
5. **Programowe mechanizmy bezpieczeństwa aplikacji (na poziomie kontenera)**

# 1. Protokół SSL do zabezpieczenia aplikacji na poziomie protokołu transportowego http - https (1)

- **Instalacja certyfikatu serwera** dla niekomercyjnych zastosowań oznaczonego bez udziału dobrze znanego autorytetu certyfikacji (certificate authority - CA) takiego jak VeriSign lub Thawte - instalacja pliku keystore.jks w bazie klucza Serwera Aplikacji Java EE 5.0 (keystore file) w katalogu `<JAVAEE_HOME>/domains/domain1/config/`.
- **Instalacja certyfikatu klienta** dla niekomercyjnych zastosowań oznaczonego bez udziału dobrze znanego autorytetu certyfikacji (certificate authority - CA) takiego jak VeriSign lub Thawte - instalacja pliku cacerts.jks w bazie poufności Serwera Aplikacji Java EE 5.0 (truststore file) –w katalogu `<JAVAEE_HOME>/domains/domain1/config/`
- **Protokół SSL używa kryptografii klucza publicznego opartego na parze kluczy – zawierającej klucz publiczny i prywatny.** Jeżeli dane są szyfrowane jednym kluczem, deszyfrowane są pozostałym kluczem z pary. Ta właściwość jest fundamentem kontroli dostępu do danych i poufności transakcji. Np.. Używając SSL, serwer oblicza wartość podpisu, szyfruje ją kluczem prywatnym. Zasyfrowana wartość jest nazywana podpisem cyfrowym (*digital signature*). Klient deszyfruje podpis kluczem publicznym używając klucza publicznego z serwera i porównuje znany podpis z wartością podpisu po deszyfracji. Jeśli obie wartości są równe, klient wierzy w prawdziwość podpisu, ponieważ tylko prywatny klucz mógł być użyty do utworzenia podpisu.
- **Cyfrowe certyfikaty są używane w protokole HTTPS do uwierzytelniania klientów internetowych – dokumenty tożsamości serwera www i klienta,** stąd protokół ten nie może wystartować bez wcześniejszego zainstalowania cyfrowego certyfikatu.

# Protokół SSL do zabezpieczenia aplikacji na poziomie protokołu transportowego http - protokół https (2)

*Technologia SSL (Secure Socket Layer)* zabezpiecza protokół HTTP. SSL pozwala przeglądarkom i serwerom www komunikować się za pomocą bezpiecznego połączenia.

## SSL zapewnia:

- ***Uwierzytelnianie (Authentication)***: Podczas inicjującego połączenia z serwerem www za pomocą bezpiecznego połączenia, serwer www prezentuje przeglądarce listę uwierzytelniającą w formie certyfikatu serwera. Certyfikat serwera umożliwia sprawdzenie „kto i jakie ma prawa”. W pewnych przypadkach serwer może żądać certyfikatu od klienta, który potwierdzi „kto i jakie ma prawa” – jest to mechanizm uwierzytelniania klienta.
- ***Poufność (Confidentiality) - szyfrowanie***: Kiedy dane są przesyłane między klientem i serwerem w sieci, strona trzecia może zobaczyć i przechwycić te dane. Dane te jednak są zaszyfrowane i nie mogą być deszyfrowane przez stronę trzecią - pozostają poufne.
- ***Integralność (Integrity) – podpisy cyfrowe***: Kiedy dane są przesyłane między klientem i serwerem w sieci, strona trzecia widzi dane i może je przechwycić, jednak nie może ich zmodyfikować.

# Informacje uzupełniające dotyczące protokołu SSL (3)

## Algorytmy w SSL

- **Inicjacja połączenia SSL - algorytm asymetryczny z kluczem publicznym i prywatnym serwera WWW (np. algorytm RSA).** Klient przeglądarki wysyła serwerowi obsługiwaną wersję protokołu SSL, sposoby szyfrowania i kompresji danych oraz identyfikator sesji. Komunikat ten zawiera również liczbę losową używaną potem przy generowaniu kluczy. Serwer zwraca klientowi wybrane parametry połączenia: wersję protokołu SSL, rodzaj szyfrowania i kompresji, oraz podobną liczbę losową. Następnie serwer może wysłać swój **certyfikat** klientowi do sprawdzenia swojej tożsamości przez klienta oraz może zażądać przesłania certyfikatu przez klienta. Klient wtedy wysyła swój **certyfikat** oraz podpisuje za pomocą klucza prywatnego związanego z tym certyfikatem skrót ustalonych wcześniej danych o połączeniu. Serwer wysyła swój **klucz publiczny**. Klient generuje losowo **klucz prywatny symetryczny zw. kluczem sesji** (korzystając z liczby wylosowanej przez siebie i serwera) do szyfrowania przesyłanej później informacji szyfrem symetrycznym, szyfruje go z użyciem klucza publicznego serwera i przesyła go do serwera. Serwer za pomocą swojego klucza prywatnego asymetrycznego odczytuje ten klucz prywatny symetryczny.
- **Przesyłanie informacji podczas połączenia SSL - algorytm symetryczny z kluczem prywatnym (np. RC-4).** Przeglądarka oraz serwer posiadają ten sam **klucz prywatny** – rozpoczyna się transmisja danych między serwerem i przeglądarką jest szyfrowana za pomocą klucza prywatnego symetrycznego. W każdej sesji generowany jest nowy klucz prywatny przez przeglądarkę.
- **Funkcja skrótu (np. MD-5) używana do generowania podpisów cyfrowych dla przesyłanej informacji.** Podpisy zapewniają integralność przesyłanej informacji - każda próba zamiany zawartości informacji zostaje wykryta.

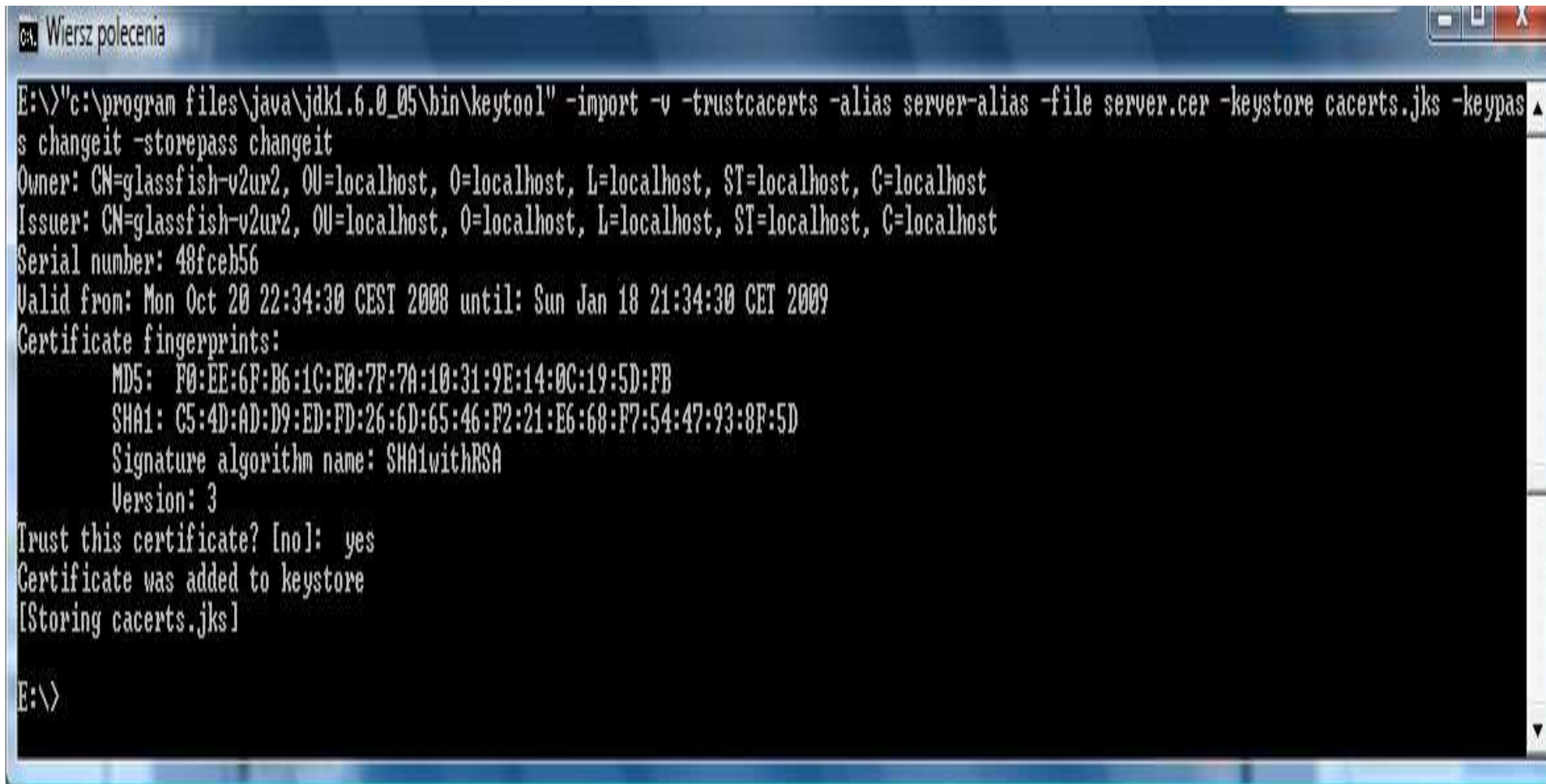
## **Certyfikat SSL (4)**

W pierwszej fazie nawiązywania połączenia SSL serwer i przeglądarka wymieniają certyfikaty. Certyfikat jest odpowiednikiem dokumentu tożsamości dla serwera WWW oraz dla klienta.

### **Certyfikat zawiera następujące składniki:**

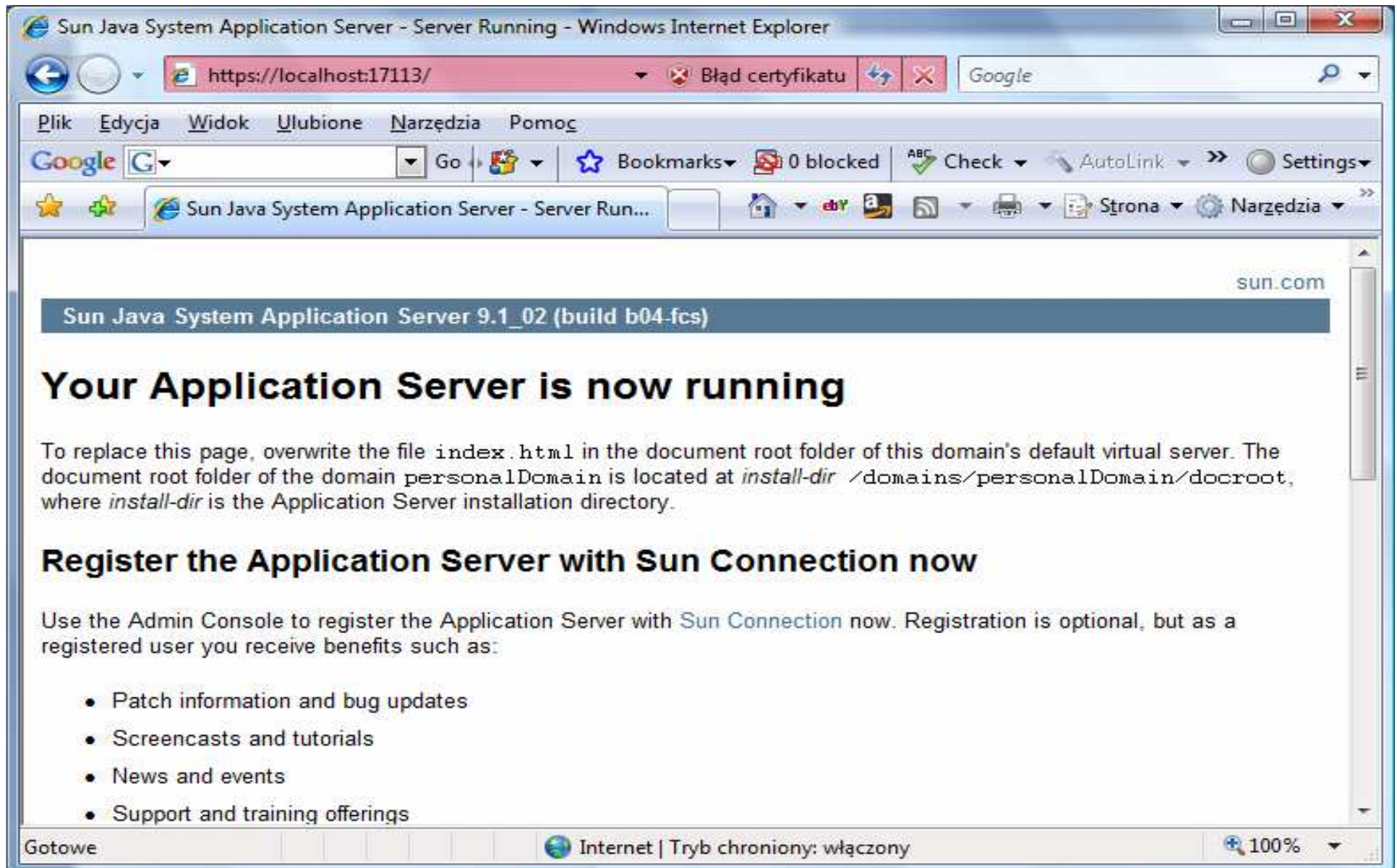
- nazwę właściciela certyfikatu,
- nazwę wydawcy certyfikatu,
- publiczny klucz właściciela dla algorytmu asymetrycznego,
- cyfrowy podpis wystawcy certyfikatu (np. Verisign),
- okres ważności,
- numer seryjny (tzw. fingerprint).

## Zawartość generowanego certyfikatu przez programistę (5)



```
Wiersz polecenia
E:\>"c:\program files\java\jdk1.6.0_05\bin\keytool" -import -v -trustcacerts -alias server-alias -file server.cer -keystore cacerts.jks -keypass
s changeit -storepass changeit
Owner: CN=glassfish-v2ur2, OU=localhost, O=localhost, L=localhost, ST=localhost, C=localhost
Issuer: CN=glassfish-v2ur2, OU=localhost, O=localhost, L=localhost, ST=localhost, C=localhost
Serial number: 48fceb56
Valid from: Mon Oct 20 22:34:30 CEST 2008 until: Sun Jan 18 21:34:30 CET 2009
Certificate fingerprints:
    MD5: F0:EE:6F:B6:1C:E0:7F:7A:10:31:9E:14:0C:19:5D:FB
    SHA1: C5:4D:AD:D9:ED:FD:26:6D:65:46:F2:21:E6:68:F7:54:47:93:8F:5D
Signature algorithm name: SHA1withRSA
Version: 3
Trust this certificate? [no]: yes
Certificate was added to keystore
[Storing cacerts.jks]
E:\>
```

## 2. Weryfikacja protokołu HTTPS obsługiwanego przez Serwera Aplikacji Java EE 5.0 za pomocą połączenia SSL (1)





## Komunikat po instalacji platformy Java EE 5.0 (2)

- Using port 13780 for Admin.
- **Using port 17012 for HTTP Instance.**
- Using port 16608 for JMS.
- Using port 12632 for IIOP.
- **Using port 17113 for HTTP\_SSL.**
- Using port 12752 for IIOP\_SSL.
- Using port 12852 for IIOP\_MUTUALAUTH.
- Using port 17618 for JMX\_ADMIN.
- Domain being created with profile:developer, as specified by variable AS\_ADMIN\_PROFILE in configuration file.
- The file in given locale [pl\_PL] at: [C:\Program Files\glassfish-v2ur2\lib\install\templates\locales\pl\_PL\index.html] could not be found. Using default (en\_US) index.html instead.
- Security Store uses: JKS Domain personalDomain created.

### 3. Ustawienie protokołu https do przesyłania danych w aplikacji (1)

The screenshot shows the NetBeans IDE interface with the 'Security' tab selected in the 'web.xml' configuration file. The left sidebar displays the project structure for 'WebWypożyczalnia3', including 'Web Pages' and 'Configuration Files'. The main editor area shows the 'Security Roles' and 'Security Constraints' sections.

**Security Roles**

Role Name	Description
klient1	
administrator1	

**Security Constraints**

**AdministratorConstraint**

Display Name: AdministratorConstraint

Web Resource Collection:

Name	URL Pattern	HTTP Method	Descri...
Administrator	/faces/*	GET, POST, HEAD, PUT, OPTIONS, TRACE, DELETE	

Enable Authentication Constraint

Description:

Role Name(s): administrator1

Enable User Data Constraint

Description:

Transport Guarantee: CONFIDENTIAL

Dropdown menu options: NONE, INTEGRAL, CONFIDENTIAL

## Ustawienie protokołu https do przesyłania danych w aplikacji (2)

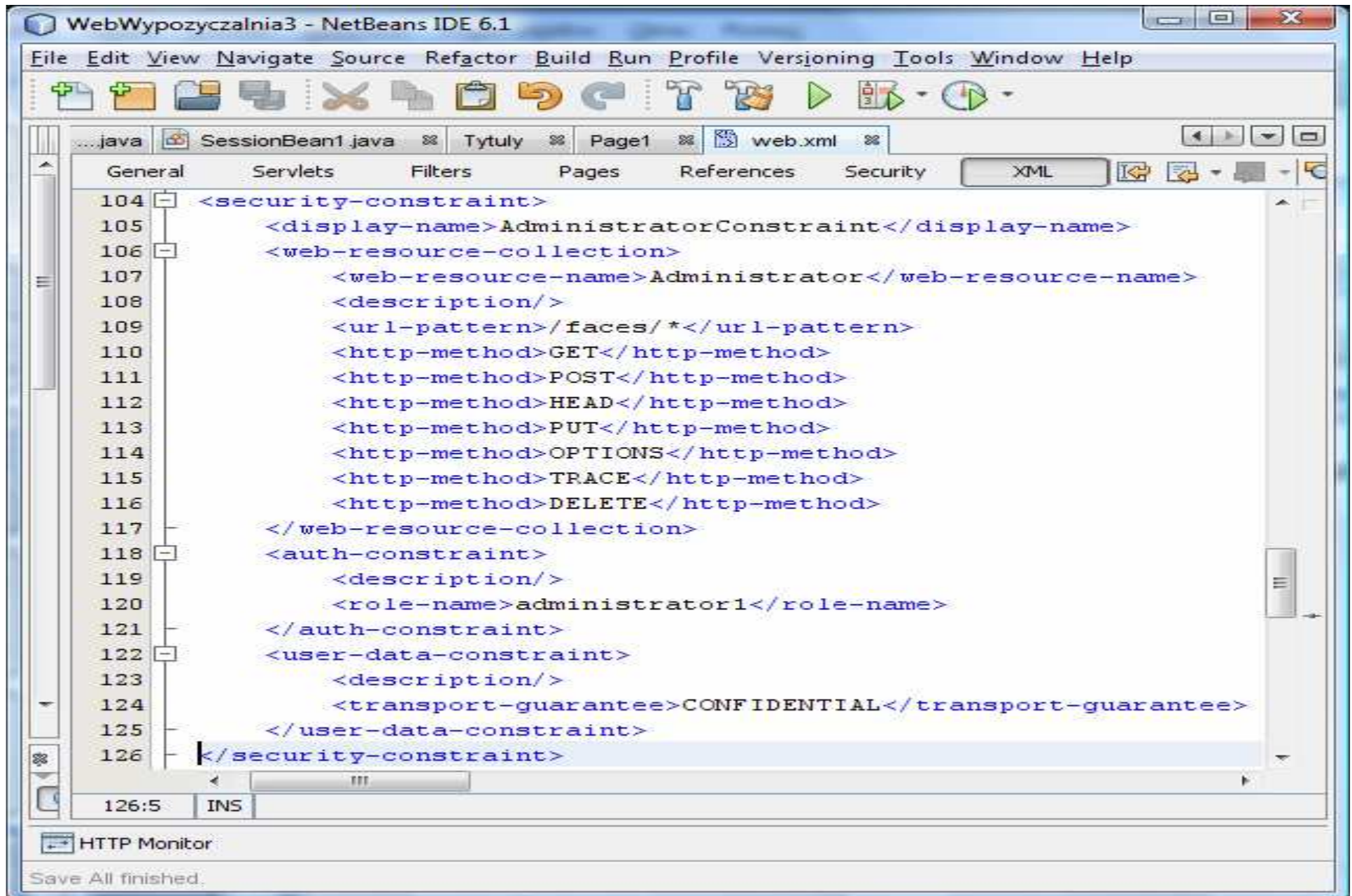
The screenshot shows the NetBeans IDE 6.1 interface for configuring a security constraint in a web application. The main window displays the 'Security' tab for the 'KlientConstraint' in the 'web.xml' file. The configuration includes:

- Display Name:** KlientConstraint
- Web Resource Collection:** A table listing the resources and methods covered by the constraint.
- Enable Authentication Constraint:** Checked.
- Role Name(s):** klient1
- Enable User Data Constraint:** Checked.
- Transport Guarantee:** CONFIDENTIAL

Name	URL Pattern	HTTP Method	Description
Klient	/faces/Page1.jsp, /faces/Tytuly.jsp, /faces/Ksiazki.jsp	GET, POST, HEAD, PUT, OPTIONS, TRACE, DELETE	

The interface also shows a list of files in the project 'WebWypożyczalnia3' on the left, including 'web.xml' and various JSP files. The bottom status bar shows 'HTTP Monitor'.

## Zawartość deskryptora aplikacji web.xml (3)



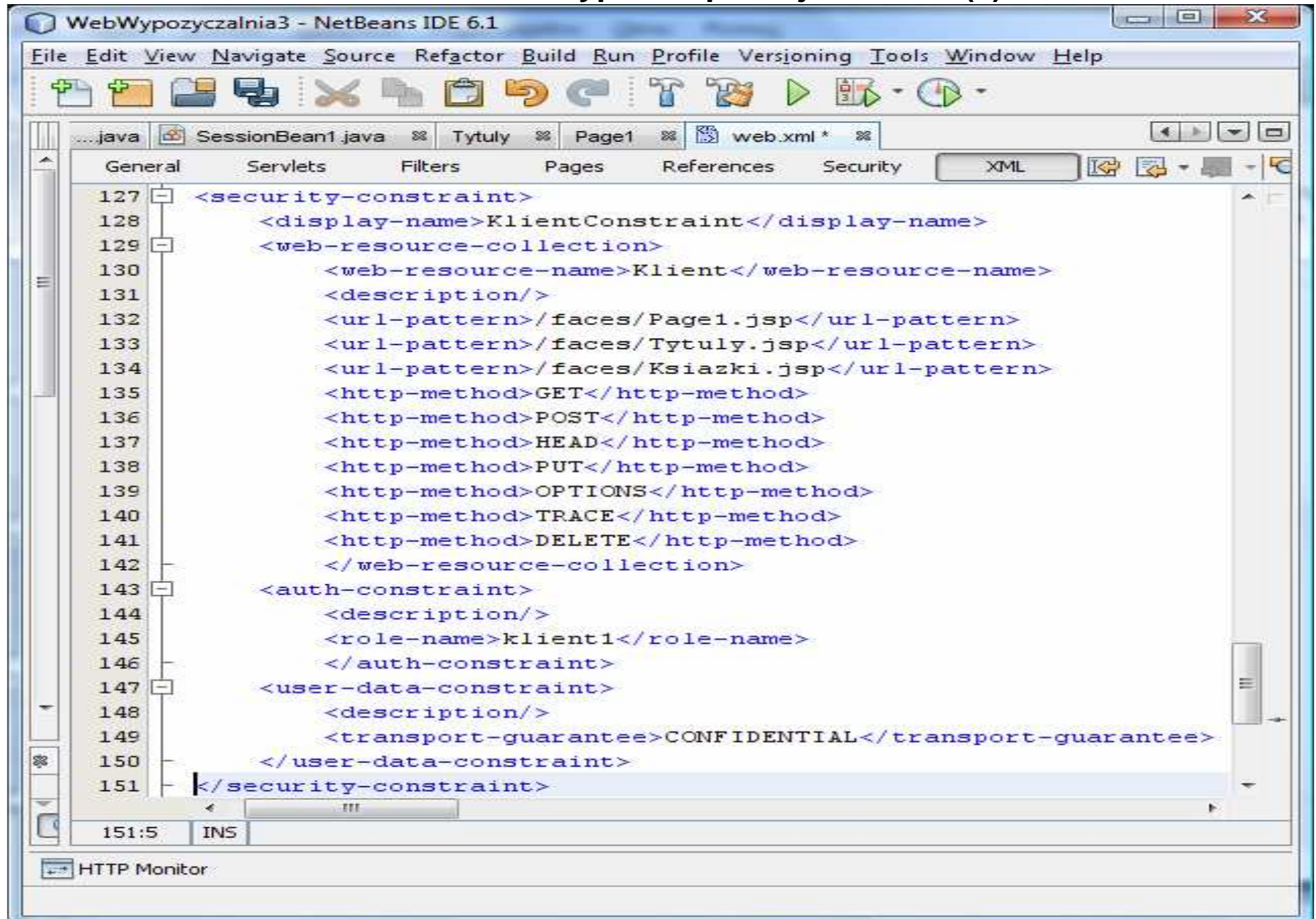
```
104 <security-constraint>
105     <display-name>AdministratorConstraint</display-name>
106     <web-resource-collection>
107         <web-resource-name>Administrator</web-resource-name>
108         <description/>
109         <url-pattern>/faces/*</url-pattern>
110         <http-method>GET</http-method>
111         <http-method>POST</http-method>
112         <http-method>HEAD</http-method>
113         <http-method>PUT</http-method>
114         <http-method>OPTIONS</http-method>
115         <http-method>TRACE</http-method>
116         <http-method>DELETE</http-method>
117     </web-resource-collection>
118     <auth-constraint>
119         <description/>
120         <role-name>administrator1</role-name>
121     </auth-constraint>
122     <user-data-constraint>
123         <description/>
124         <transport-guarantee>CONFIDENTIAL</transport-guarantee>
125     </user-data-constraint>
126 </security-constraint>
```

126:5 | INS

HTTP Monitor

Save All finished.

## Zawartość deskryptora aplikacji web.xml (4)

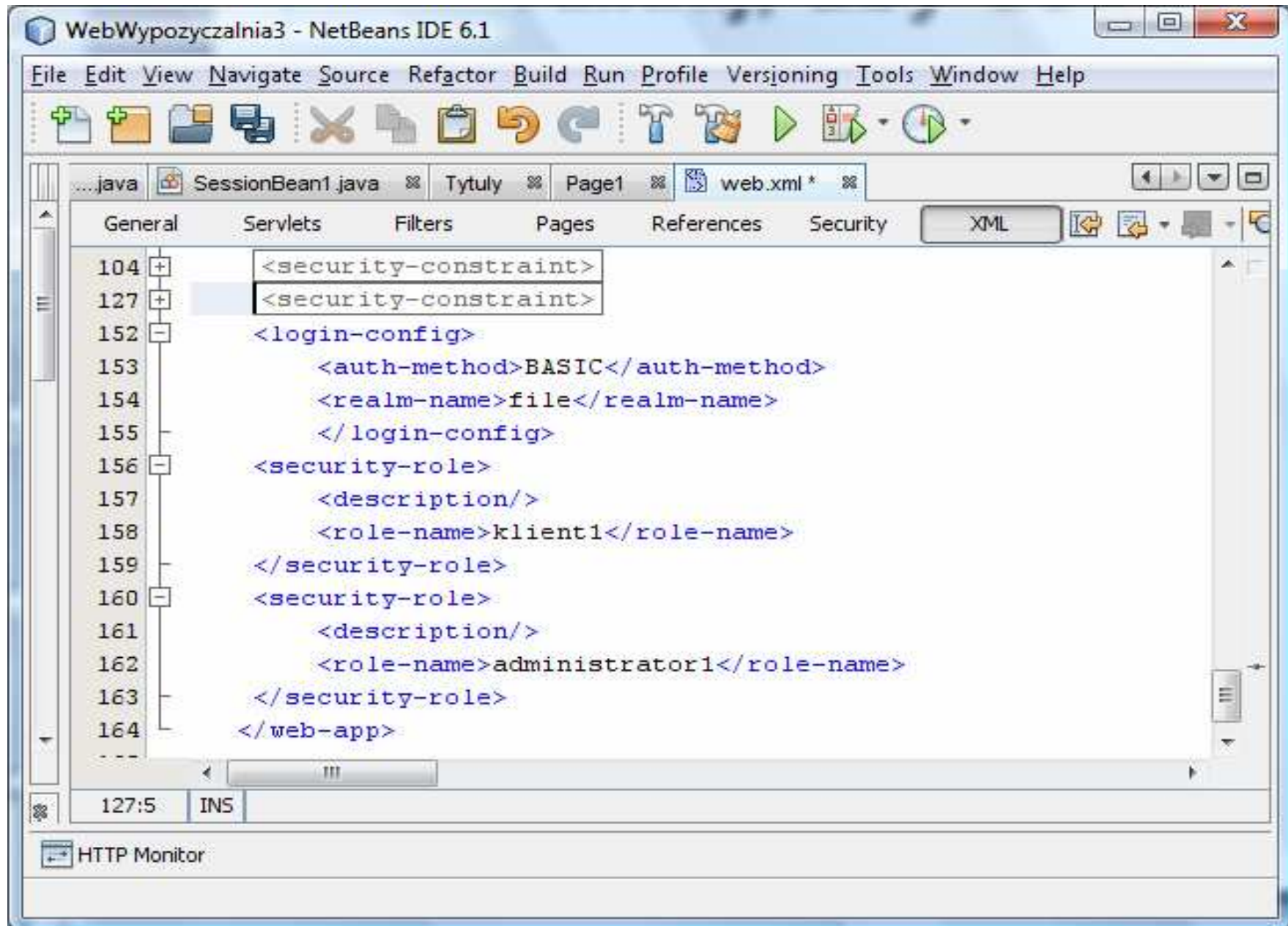


The screenshot shows the NetBeans IDE 6.1 interface with the XML editor open to the file 'web.xml'. The XML content is as follows:

```
127 <security-constraint>
128     <display-name>KlientConstraint</display-name>
129     <web-resource-collection>
130         <web-resource-name>Klient</web-resource-name>
131         <description/>
132         <url-pattern>/faces/Page1.jsp</url-pattern>
133         <url-pattern>/faces/Tytuly.jsp</url-pattern>
134         <url-pattern>/faces/Ksiazki.jsp</url-pattern>
135         <http-method>GET</http-method>
136         <http-method>POST</http-method>
137         <http-method>HEAD</http-method>
138         <http-method>PUT</http-method>
139         <http-method>OPTIONS</http-method>
140         <http-method>TRACE</http-method>
141         <http-method>DELETE</http-method>
142     </web-resource-collection>
143     <auth-constraint>
144         <description/>
145         <role-name>klient1</role-name>
146     </auth-constraint>
147     <user-data-constraint>
148         <description/>
149         <transport-guarantee>CONFIDENTIAL</transport-guarantee>
150     </user-data-constraint>
151 </security-constraint>
```

The status bar at the bottom of the editor shows the cursor is at line 151, column 5, in the 'INS' (Insert) mode. Below the editor, the 'HTTP Monitor' window is visible.

## Zawartość deskryptora aplikacji web.xml (5)

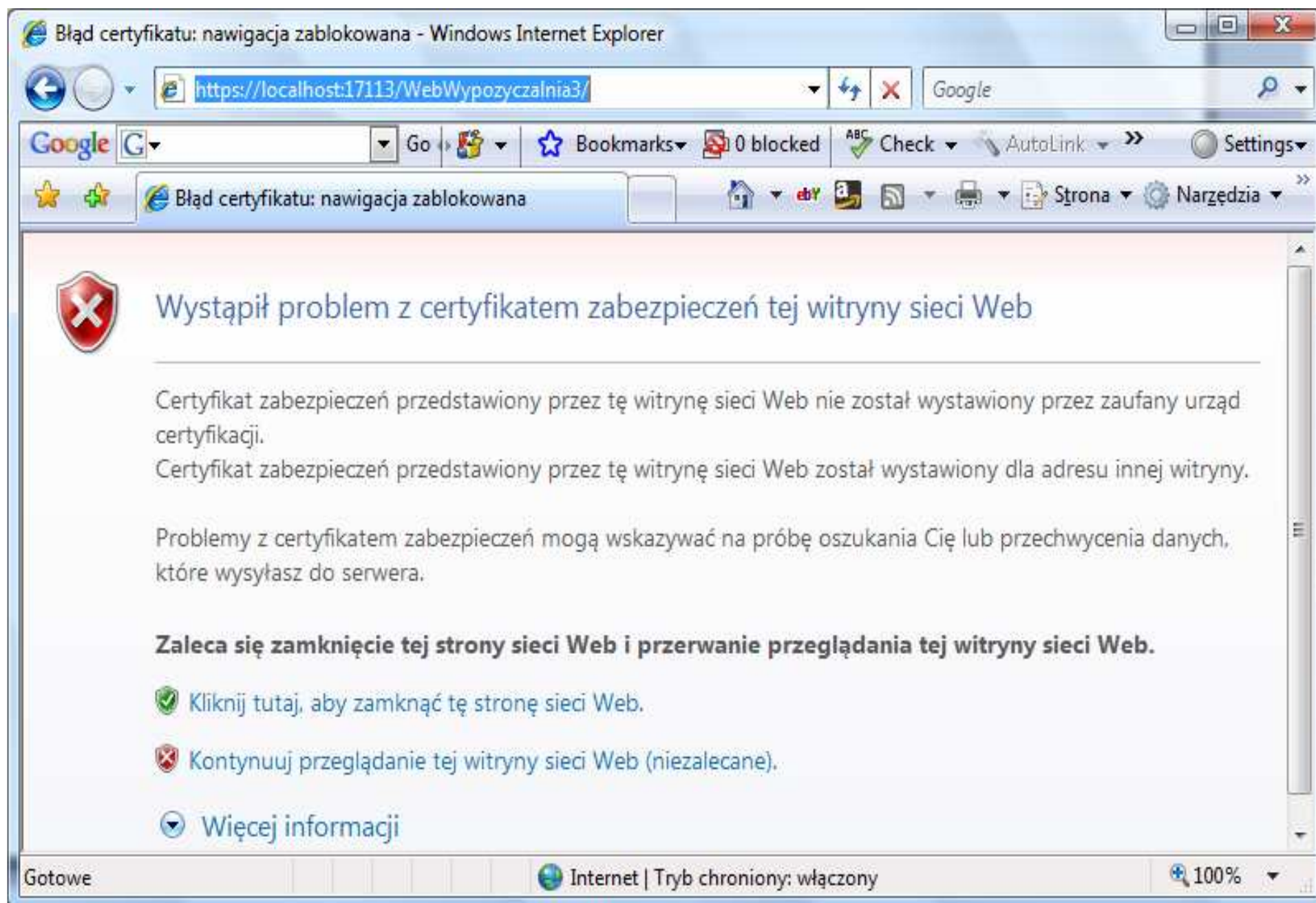


The screenshot shows the NetBeans IDE 6.1 interface with the XML editor open to the file web.xml. The editor displays the following XML content:

```
104 <security-constraint>
127 <security-constraint>
152 <login-config>
153     <auth-method>BASIC</auth-method>
154     <realm-name>file</realm-name>
155 </login-config>
156 <security-role>
157     <description/>
158     <role-name>klient1</role-name>
159 </security-role>
160 <security-role>
161     <description/>
162     <role-name>administrator1</role-name>
163 </security-role>
164 </web-app>
```

The status bar at the bottom indicates the cursor is at line 127, column 5, in the INS (Insert) mode. The HTTP Monitor window is visible at the bottom of the IDE.

## 4. Uruchomienie aplikacji - zastosowanie protokołu HTTPS (1)




Błąd certyfikatu: nawigacja zablokowana - Windows Internet Explorer

<https://localhost:17113/WebWypożyczalnia3/>

Google G Go Bookmarks 0 blocked Check AutoLink Settings




Błąd certyfikatu: nawigacja zablokowana Strona Narzędzia

 Wystąpił problem z certyfikatem zabezpieczeń tej witryny sieci Web

Certyfikat zabezpieczeń przedstawiony przez tę witrynę sieci Web nie został wystawiony przez zaufany urząd certyfikacji.  
Certyfikat zabezpieczeń przedstawiony przez tę witrynę sieci Web został wystawiony dla adresu innej witryny.

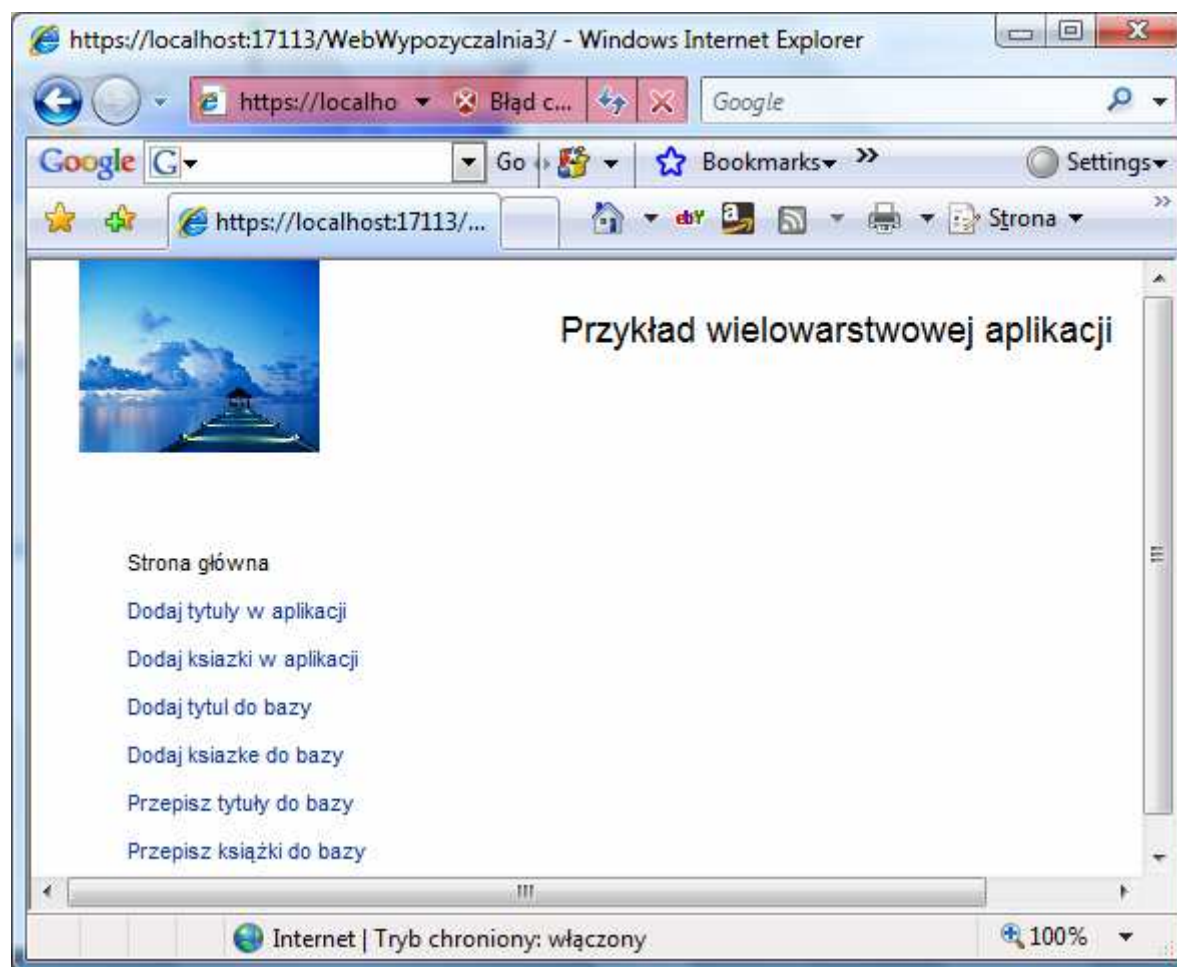
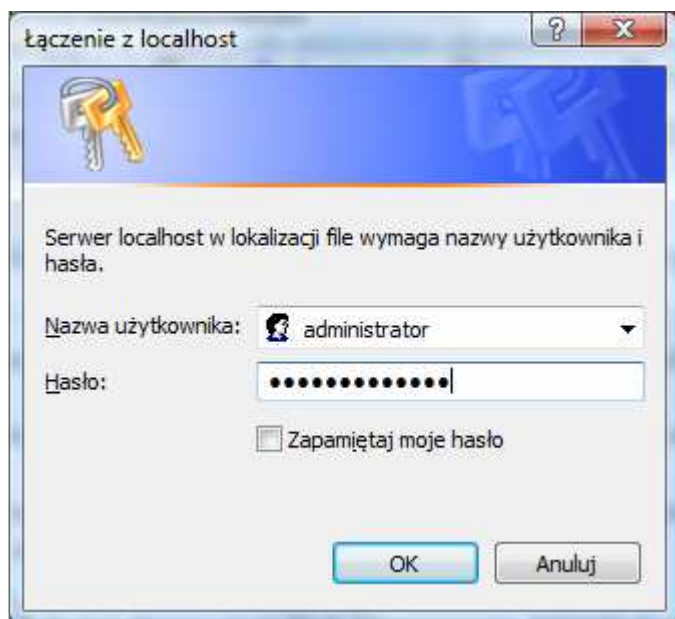
Problemy z certyfikatem zabezpieczeń mogą wskazywać na próbę oszukania Cię lub przechwycenia danych, które wysyłasz do serwera.

**Zaleca się zamknięcie tej strony sieci Web i przerwanie przeglądania tej witryny sieci Web.**

-  [Kliknij tutaj, aby zamknąć tę stronę sieci Web.](#)
-  [Kontynuuj przeglądanie tej witryny sieci Web \(niezalecane\).](#)
-  [Więcej informacji](#)

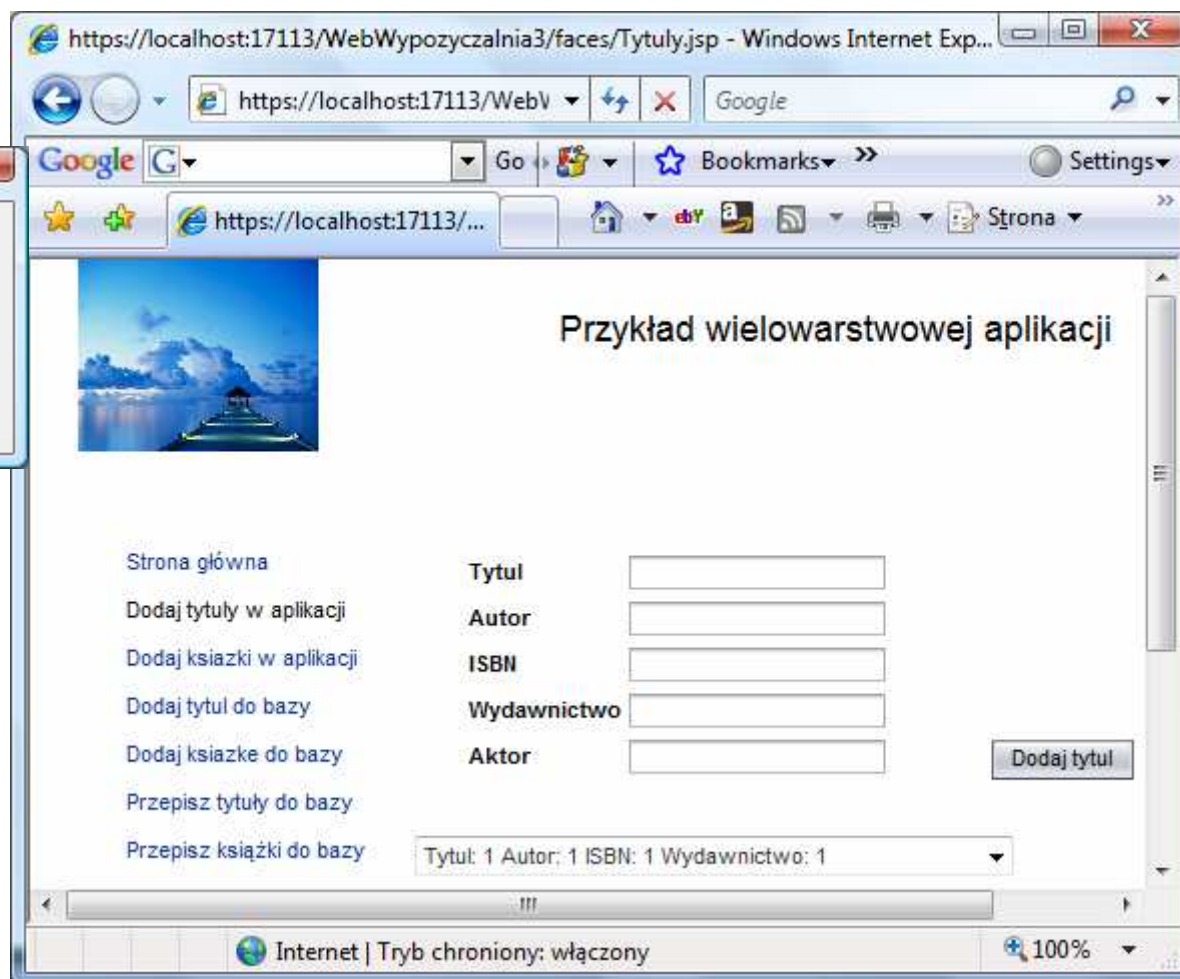
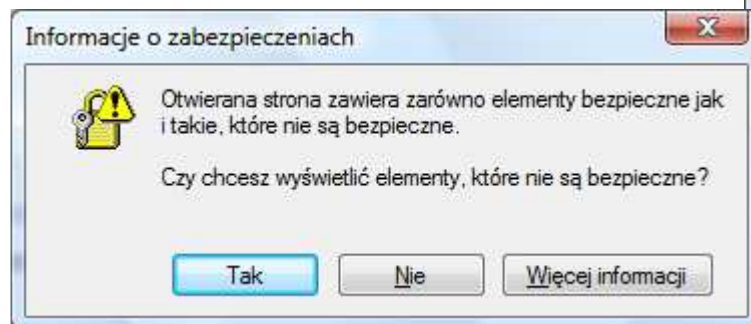
Gotowe Internet | Tryb chroniony: włączony 100%

## Uruchomienie aplikacji - zastosowanie protokołu HTTPS (2)





## Uruchomienie aplikacji - zastosowanie protokołu HTTPS (3)



https://localhost:17113/WebWypożyczalnia3/faces/Ksiazki.jsp - Windows Internet Explorer

https://localhost:17113/WebWypożyczal... Błąd certyfikatu Google

Google G Go Bookmarks 0 blocked Check

https://localhost:17113/WebWypożyczalnia3...



## Przykład wielowarstwowej aplikacji

- Strona główna
- Dodaj tytuły w aplikacji
- Dodaj książki w aplikacji
- Dodaj tytuł do bazy
- Dodaj książkę do bazy
- Przepisz tytuły do bazy
- Przepisz książki do bazy

Gotowe

### Informacje o zabezpieczeniach

Otwierana strona zawiera zarówno elementy bezpieczne jak i takie, które nie są bezpieczne.

Czy chcesz wyświetlić elementy, które nie są bezpieczne?

Tak Nie Więcej informacji

https://localhost:17113/WebWypożyczalnia3/faces/Ksiazki.jsp - Windows Internet Explorer

https://localhost:17113/WebWypożyczalnia3/faces/Ksiazki.jsp Google

Google G Go Bookmarks 0 blocked Check Settings

https://localhost:17113/WebWypożyczalnia3... Strona Narzędzia



## Przykład wielowarstwowej aplikacji

- Strona główna
- Dodaj tytuły w aplikacji
- Dodaj książki w aplikacji
- Dodaj tytuł do bazy
- Dodaj książkę do bazy
- Przepisz tytuły do bazy
- Przepisz książki do bazy

Numer

Termin

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1

Dodaj książkę

Gotowe

Internet | Tryb chroniony: włączony

100%

## 5. Programowe mechanizmy bezpieczeństwa aplikacji (na poziomie kontenera)

Interfejs **HttpServletRequest** dostarcza następujące metody umożliwiające dostęp do informacji o bezpieczeństwie dostępu nałożonym na użytkownika aplikacji:

1. ***getRemoteUser***: podaje nazwę zdalnego użytkownika, który był uwierzytelniony. Jeśli nie uwierzytelniano użytkownika, metoda zwraca **null**.
2. ***isUserInRole***: określa czy zdalny użytkownik występuje w roli podanej jako parametr typu **String** metody ***isUserInRole***. Jeśli nie uwierzytelniano użytkownika, metoda zwraca **false**.

Można również użyć jako parametru metody roli zdefiniowanej za pomocą adnotacji **@DeclareRoles** lub nazwy roli zdefiniowanej w zasięgu elementu **<security-role-ref>** w zagnieżdżonym elemencie **<role-name>** w deskrytorze aplikacji.

3. ***getUserPrincipal***: podaje nazwę typu **principal** bieżącego użytkownika i zwraca obiekt typu **java.security.Principal**. Jeśli nie uwierzytelniano użytkownika, metoda zwraca **null**.

## Ad. 2

**Add Security Role Reference**

Role Ref Name:

Role Ref Link:

Description:

WebWypozyczalnia3 - NetBeans IDE 6.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Files: Serv...

WebWypozyczalnia3

- Web Pages
- Configuration Files
  - MANIFEST.MF
  - faces-config.xml
  - sun-web.xml
  - web.xml
- Server Resources

General **Servlets** Filters Pages References Security XML

Security Role References:

Role Ref Name	Role Ref Link	Description
klient2	klient1	

NetBeans IDE 6.1

Source Refactor Build Run Profile Versioning Tools Window Help

...age RequestBean1.java SessionBean1.java Tytuly Page1 web.xml \*

General Servlets Filters Pages References Security **XML**

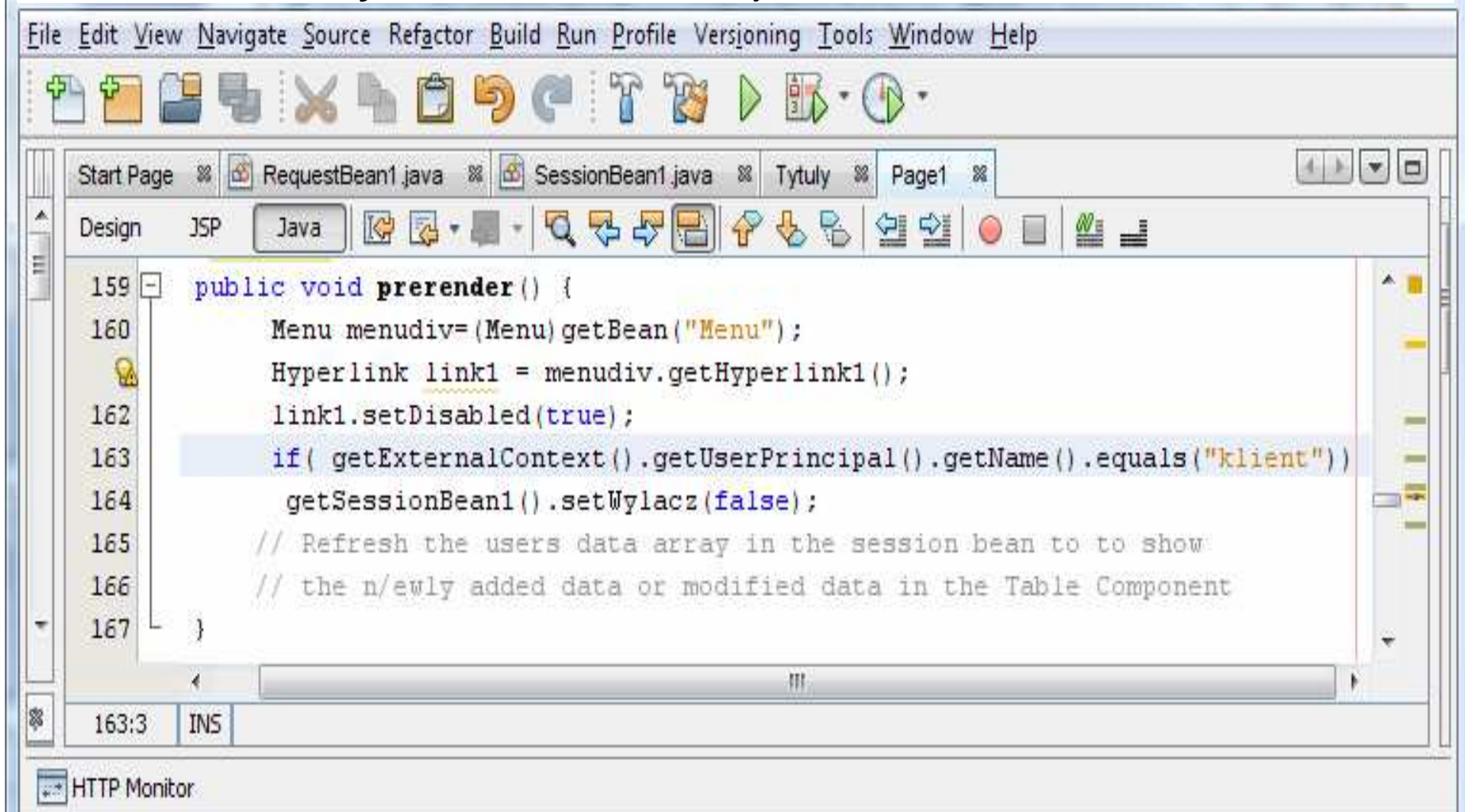
```
33 <servlet>
34   <servlet-name>Faces Servlet</servlet-name>
35   <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
36   <init-param>
37     <param-name>javax.faces.LIFECYCLE_ID</param-name>
38     <param-value>com.sun.faces.lifecycle.PARTIAL</param-value>
39   </init-param>
40   <load-on-startup>1</load-on-startup>
41   <security-role-ref>
42     <description/>
43     <role-name>klient2</role-name>
44     <role-link>klient1</role-link>
45   </security-role-ref>
46 </servlet>
```

46:15 INS

web.xml - Navigator

HTTP Monitor

**Ad.3** - W klasie **Page1.java** strony startowej w metodzie **prerender** pobrano metodą **getExternalContext** obiekt implementujący interfejs **HttpServletRequest** i następnie metodą **getUserPrincipal** pobrano obiekt typu **java.security.Principal**, reprezentujący nazwę uwierzytelnionego użytkownika. Pobrana nazwa jest porównana z nazwą „klient”. Jeśli uwierzytelniony użytkownik ma nazwę „klient”, wartość atrybutu **wylacz** w **SessionBean1** jest ustawiona na **false**.



The screenshot shows an IDE window with the following content:

- Menu bar: File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, Help
- Toolbar: Standard IDE icons for file operations and development.
- Tab bar: Start Page, RequestBean1.java, SessionBean1.java, Tytuly, Page1
- Design view: JSP, Java
- Code editor:

```
159 public void prerender() {
160     Menu menudiv= (Menu) getBean("Menu");
161     Hyperlink link1 = menudiv.getHyperlink1();
162     link1.setDisabled(true);
163     if( getExternalContext().getUserPrincipal().getName().equals("klient") )
164         getSessionBean1().setWylacz(false);
165     // Refresh the users data array in the session bean to to show
166     // the n/ewly added data or modified data in the Table Component
167 }
```
- Status bar: 163:3 INS
- Bottom panel: HTTP Monitor

# Definicja atrybutu **wylacz** typu **boolean** w klasie **SessionBean1**

The screenshot displays the NetBeans IDE 6.1 interface. The main editor window shows the source code for the `SessionBean1` class, which extends `AbstractSessionBean`. The code defines a private boolean attribute `wylacz` and two public methods: `isWylacz()` and `setWylacz(boolean wylacz)`. The `isWylacz()` method returns the value of `wylacz`, and `setWylacz()` sets the value of `wylacz` to the provided parameter.

```
23 public class SessionBean1 extends AbstractSessionBean {
24
25     /**...*/
34
37     private boolean wylacz=true;
38
39     public boolean isWylacz() {
40         return wylacz;
41     }
42
43     public void setWylacz(boolean wylacz) {
44         this.wylacz = wylacz;
45     }
46 }
```

The left sidebar shows the project structure for `WebWypożyczalnia3` and the `isWylacz - Navigator` window, which displays the `Members View` of the class. The `isWylacz() : boolean` method is highlighted in the navigator.

Powiązanie w klasie **Tytuly.java** strony aplikacji właściwości **rendered** przycisku **dodajtytul** typu **Button** z atrybutem **wylacz** typu **boolean** w klasie **SessionBean1**. Domyślna wartość zmiennej **wylacz** jest równa **true**, natomiast po uwierzytelnieniu bieżącego klienta o nazwie „klient” wartość atrybutu jest ustawiona na **false** – wtedy kod html przycisku jest usunięty ze strony w fazie **response**.

The screenshot shows an IDE interface for a web application. The main window displays a JSP page titled "Przykład wielowarstwowy" (Multi-layered example). The page contains a navigation menu with links like "Strona główna", "Dodaj tytuły w aplikacji", and "Dodaj książki w aplikacji". Below the menu is a form with input fields for "Tytuł", "Autor", "ISBN", "Wydawnictwo", and "Aktor", along with a dropdown menu. The "rendered" property of the "dodajtytul: Dodaj tytuł" button is set to `#{SessionBean1.wylacz}`. The Properties palette on the right shows the "rendered" property is checked, and a tooltip explains that this attribute indicates whether the HTML code for the component should be included in the rendered HTML page.


**dodajtytul: Dodaj tytuł - Properties**

disabled	<input type="checkbox"/>
reset	<input type="checkbox"/>
toolTip	<input type="checkbox"/>
visible	<input checked="" type="checkbox"/>
Accessibility	
alt	<input type="checkbox"/>
tabIndex	<input type="checkbox"/>
JavaScript	
onBlur	<input type="checkbox"/>
onClick	<input type="checkbox"/>
onDbClick	<input type="checkbox"/>
onFocus	<input type="checkbox"/>
onKeyDown	<input type="checkbox"/>
onKeyPress	<input type="checkbox"/>
onKeyUp	<input type="checkbox"/>
onMouseDown	<input type="checkbox"/>
onMouseMove	<input type="checkbox"/>
onMouseOut	<input type="checkbox"/>
onMouseOver	<input type="checkbox"/>
onMouseUp	<input type="checkbox"/>
Advanced	
actionListenerExp	<input type="checkbox"/>
immediate	<input type="checkbox"/>
<b>rendered</b>	<input checked="" type="checkbox"/> <code>#{SessionBean1.wylacz}</code>

**rendered** attribute to indicate whether the HTML code for the component should be included in the rendered HTML page

Po uwierzytelnieniu klienta o nazwie „klient” nie wyświetla się przycisk **dodajtytuł**

Łączenie z localhost



Serwer localhost w lokalizacji file wymaga nazwy użytkownika i hasła.


Nazwa użytkownika: klient

Hasło: .....

Zapamiętaj moje hasło

OK Anuluj

Prz



Strona główna

Dodaj tytuły w aplikacji

Dodaj książki w aplikacji

Dodaj tytuł do bazy

Dodaj książkę do bazy

Przepisz tytuły do bazy

Przepisz książki do bazy

Internet | Tryb chroniony: włączony

https://localhost:17113/WebWypożyczalnia3/faces/Tytuły.jsp - Windows Internet E...

https://localhost:17113/Wel

Google


Go

Bookmarks

Settings

Strona

Przykład wielowarstwowej aplikacji



Strona główna

Dodaj tytuły w aplikacji

Dodaj książki w aplikacji

Dodaj tytuł do bazy

Dodaj książkę do bazy

Przepisz tytuły do bazy

Przepisz książki do bazy

Tytuł

Autor

ISBN

Wydawnictwo

Aktor

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1


Internet | Tryb chroniony: włączony

100%



Po uwierzytelnieniu klienta o nazwie „administrator” wyświetla się przycisk **dodajtytuł**

Łączenie z localhost



Serwer localhost w lokalizacji file wymaga nazwy użytkownika i hasła.

Nazwa użytkownika: administrator

Hasło: .....

Zapamiętaj moje hasło

OK Anuluj

localhost:17113/WebWypożyczalnia3/faces/Tytuly.jsp - Windows Internet Exp...

https://localhost:17113/WebV Google

Go Bookmarks Settings

https://localhost:17113/...

Przykład wielowarstwowej aplikacji

Strona główna

Dodaj tytuły w aplikacji

Dodaj książki w aplikacji

Dodaj tytuł do bazy

Dodaj książkę do bazy

Przepisz tytuły do bazy

Przepisz książki do bazy

Tytuł

Autor

ISBN

Wydawnictwo

Aktor

Dodaj tytuł

Tytuł: 1 Autor: 1 ISBN: 1 Wydawnictwo: 1

Internet | Tryb chroniony: włączony 100%