

# Wprowadzenie do technologii JavaServer Faces 2.2

na podstawie

<https://docs.oracle.com/javaee/7/JEETT.pdf>

rozdział 7

## Technologie internetowe 2

# Technologia JavaServer Faces (JSF) (1)

Technologia JavaServer Faces jest środowiskiem do tworzenia interfejsu użytkownika w aplikacjach internetowych.

Głównymi składnikami technologii **JavaServer Faces** są:

- framework do budowy GUI zbudowanego z komponentów.
- elastyczny model do generowania elementów strony www w postaci różnych rodzajów znaczników HTML lub innych języków i technologii opartych na znacznikach. Obiekt typu `Renderer` generuje znaczniki do renderowania komponentu i konwertuje dane przechowywane w obiektach modelu do typów, które mogą być reprezentowane w widoku.
- standard `RenderKit` do generowania znaczników HTML.
- język wyrażeń (EL – Expression Language)

# Technologia JavaServer Faces (JSF) (2)

Wszystkie te funkcje są dostępne za pomocą standardowego API Javy i plików konfiguracyjnych typu XML.

W platformie Java EE 7, nowe funkcje JavaServer Faces obejmują:

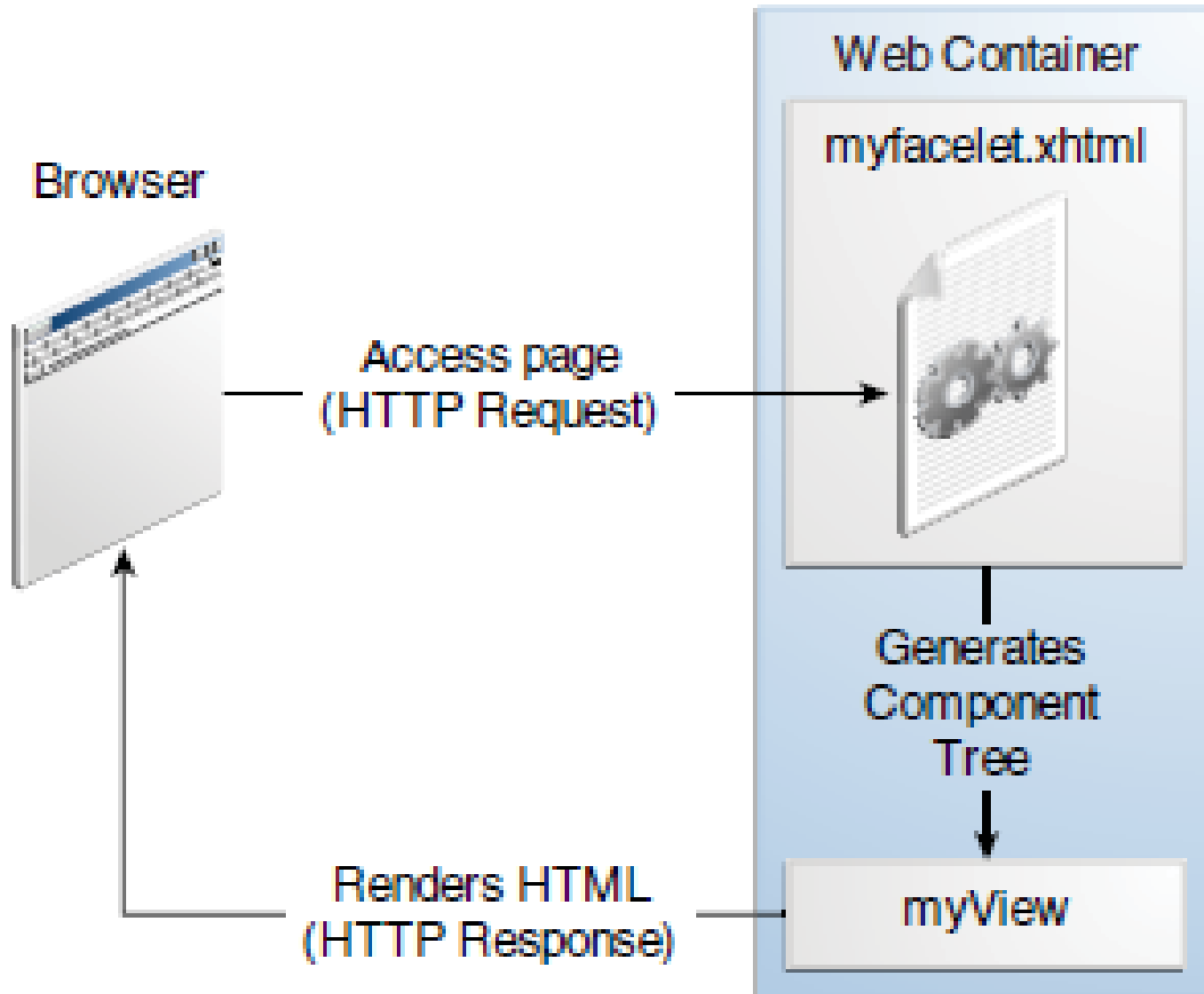
- wykorzystanie HTML5 podczas tworzenie stron www
- zastąpienie technologii wyświetlania opartą na stronach JSP(JavaServer Pages) technologią Facelets opartą na plikach XHTML
- zastosowanie biblioteki zasobów

Platforma Java EE 7 wymaga JavaServer Faces 2.2 i języka wyrażeń EL 3.0.

**Table 9-1 Example Expressions**

<b>EL Expression</b>	<b>Result</b>
<code>#{1 &gt; (4/2)}</code>	false
<code>#{4.0 &gt;= 3}</code>	true
<code>#{100.0 == 100}</code>	true
<code>#{(10*10) ne 100}</code>	false
<code>#{'a' &gt; 'b'}</code>	false
<code>#{'hip' lt 'hit'}</code>	true
<code>#{4 &gt; 3}</code>	true
<code>#{1.2E4 + 1.4}</code>	12001.4
<code>#{3 div 4}</code>	0.75
<code>#{10 mod 4}</code>	2
<code>#{((x, y) -&gt; x + y)(3, 5.5)}</code>	8.5
<code>[1,2,3,4].stream().sum()</code>	10
<code>[1,3,5,2].stream().sorted().toList()</code>	[1, 2, 3, 5]
<code>#{!empty param.Add}</code>	False if the request parameter named Add is null or an empty string
<code>#{pageContext.request.contextPath}</code>	The context path
<code>#{sessionScope.cart.numberOfItems}</code>	The value of the <code>numberOfItems</code> property of the session-scoped attribute named <code>cart</code>
<code>#{param['mycom.productId']}</code>	The value of the request parameter named <code>mycom.productId</code>
<code>#{header["host"]}</code>	The host
<code>#{departments[deptName]}</code>	The value of the entry named <code>deptName</code> in the <code>departments</code> map
<code>#{requestScope['javax.servlet.forward.servlet_path']}</code>	The value of the request-scoped attribute named <code>javax.servlet.forward.servlet_path</code>
<code>#{customer.lName}</code>	Gets the value of the property <code>lName</code> from the <code>customer</code> bean during an initial request; sets the value of <code>lName</code> during a postback
<code>#{customer.calcTotal}</code>	The return value of the method <code>calcTotal</code> of the <code>customer</code> bean

# Proces „żądanie-odpowiedź” w technologii JavaServer Faces



# Zadania realizowane podczas cyklu życia aplikacji typu JavaServer Faces

- Obsługa żądań
- Dekodowanie parametrów
- Modyfikacja i zachowanie stanu
- Renderowania stron internetowych i wysłanie ich do przeglądarki internetowej

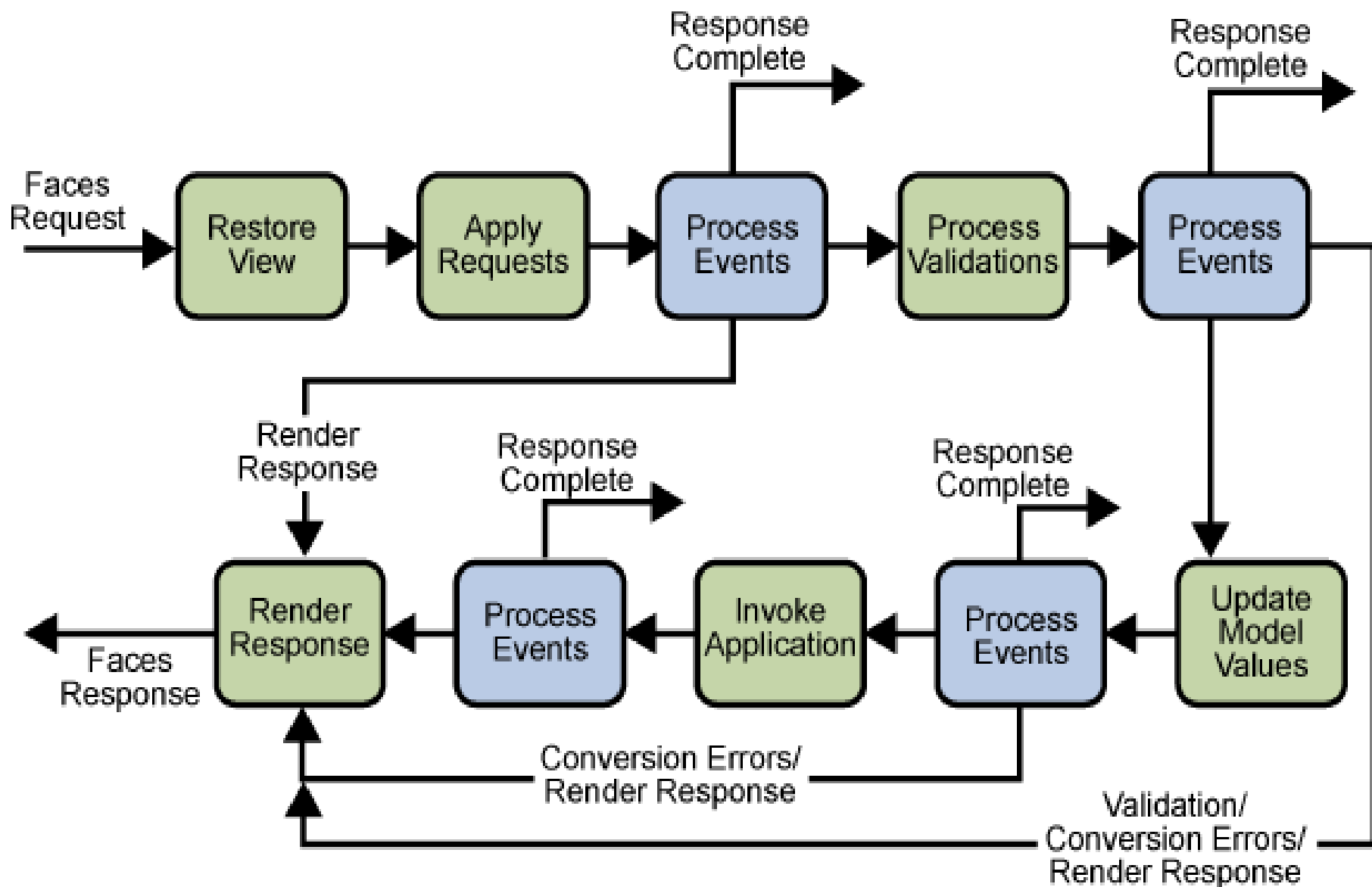
Framework aplikacji typu **JavaServer Faces** zarządza **fazami cyklu życia**:

- **automatycznie w przypadku prostych aplikacji,**
- **ręcznie w przypadku złożonych aplikacji.**

# Modele JSF

1. Komponenty typu `javax.faces.component.UIComponent` przeznaczone do **specyfikacji stanu i zachowania komponentów UI** (interfejsu użytkownika)
2. Model sposobu **renderowania** komponentów JSF na elementy strony HTML
3. Model konwersji definiujący sposób rejestrowania **konwerterów danych** w komponencie
4. Model **obsługi zdarzeń** komponentów
5. Model walidacji definiujący sposób rejestrowania **walidatorów danych** w komponencie

# Standard cyklu życia „Request-Response” dla JavaServer Faces





# Opis faz cyklu życia JavaServer Faces (1)

- **Dwa typy cyklu życia:**
  - **initial request** (pierwsze wywołanie strony) – realizowane są tylko fazy **RestoreView** i **Render Response**
  - **postbacks** (obsługa formularza): realizowane są wszystkie fazy.
- **Akcje **Response Complete**** oznaczają odwołanie do innej części aplikacji, nie zawierającej komponentów Java Server Faces
- **Dwie główne fazy cyklu życia:**
  - **execute** – składa się z podfaz, wspierających obsługę drzewa komponentów: w zakresie konwertowania i walidacji danych, obsługi zdarzeń komponentów oraz przekazania danych komponentów do obiektów typu Managed Bean w sposób uporządkowany
  - **Render**
- **Drzewo komponentów** reprezentuje stronę JSF, zwaną **view (widok)**.
- W przypadku, gdy komponent typu JavaServer Faces w procesie typu **submit przekierowuje żądanie do innej strony**, też w ustalony sposób realizowane są fazy cyklu życia (w celu konwersji i walidacji, zmiany stanu modelu i generowania widoku)

## Opis faz cyklu życia JavaServer Faces (2)

- Typy faz w przypadku pełnego cyklu życia np. wpłata gotówki
  - **Restore View:**
    - reakcja na zdarzenie wprowadzenia wartości na formularzu w polu typu TextField i wysłanie przez klienta strony (tworzenie **widoku strony** jako drzewa komponentów UI i łączenie ich z walidatorami, konwerterami i słuchaczami zdarzeń
    - aktualizacja tego widoku w **javax.faces.context.FacesContext** dla cyklu typu **postback**
    - lub zapamiętanie pustego widoku w przypadku cyklu **initial request**. Pusty widok w fazie **RenderResponse** jest **renderowany z wykorzystaniem komponentów powiązanych ze znacznikami strony**. Pierwsze żądanie powoduje utworzenie nowego drzewa komponentów i przechowanie go w obiekcie typu **javax.faces.context.FacesContext**, który reprezentuje wszystkie informacje związane z procesem obsługi żądania i tworzenia odpowiedzi

## Opis faz cyklu życia JavaServer Faces (3)

### – Apply Request Values:

- konwersja danych i zachowanie ich wartości, wiązanie zdarzeń ze słuchaczami,
- obsługa konwersji, walidacji i zdarzeń w przypadku, gdy atrybut **immediate** ma wartość **true**, wtedy konwersje, walidacje oraz zdarzenia powiązane z takim komponentem zachodzą w tej fazie - możliwość akcji **Response Complete**,
- możliwość przejścia do fazy **Render Response** jako wynik obsługi błędów.
- W przypadku żądania przekierowania do innego zasobu aplikacji internetowej np. do usług internetowych lub generowanie odpowiedzi, która nie zawiera komponentów typu JavaServer Faces, wówczas program w celu pominięcia fazy **RenderResponse** musi wywołać metodę **facesContext.responseComplete**
- Na końcu fazy komponenty zawierają dane zaktualizowane
- Częściowy cykl życia aplikacji typu JSF oraz częściowy rendering

# Opis faz cyklu życia JavaServer Faces (4)

## – Process Validations:

- obsługa walidacji za pomocą walidatorów umieszczonych w komponentach widoku – użycie metody `validate` (`processValidators`), konwersji i zdarzeń,
- zapamiętanie wartości wprowadzonych w formularzu,
- możliwość przejścia do fazy **Render Response** (obsługa błędów lub normalna reakcja) – jeśli metoda `validate` zostanie wywołana przez walidator lub słuchacze zdarzeń wywołają metodę `renderResponse` od bieżącego obiektu typu `FacesContext`. W przypadku błędu, do obiektu typu `FacesContext` zostaną wstawione komunikaty o błędach i renderowanie strony z informacją o błędzie
- W przypadku żądania przekierowania do innego zasobu aplikacji internetowej np. do usług internetowych lub generowanie odpowiedzi, która nie zawiera komponentów typu JavaServer Faces, wówczas program w celu pominięcia fazy **RenderResponse** musi wywołać metodę `facesContext.responseComplete`
- Częściowy cykl życia aplikacji typu JSF oraz częściowy rendering

# Opis faz cyklu życia JavaServer Faces (5)

## –Update Model Values:

- Jeśli dane są poprawne, wtedy wybrane właściwości komponentów w widoku są zaktualizowane - w przeciwnym wypadku następuje przejście do fazy **RenderResponse** i renderowanie strony z informacją o błędzie
- Jeśli jakikolwiek komponent wywoła metodę **updateModels** lub **renderResponse** od aktualnego obiektu typu **FacesContext** następuje przejście do fazy **RenderResponse**
- W przypadku żądania przekierowania do innego zasobu aplikacji internetowej np. do usług internetowych lub generowanie odpowiedzi, która nie zawiera komponentów typu JavaServer Faces, wówczas program w celu pominięcia fazy **RenderResponse** musi wywołać metodę **facesContext.responseComplete**

# Opis faz cyklu życia JavaServer Faces (6)

## – Invoke Application:

- realizacja zdarzeń np. typu **submit** formularza – często obsługa dotyczy jedynie rekonstrukcji widoku
- lub połączenie z inną stroną,
- możliwość przejścia do fazy **Render Response** (obsługa błędów lub normalna reakcja)
- W przypadku żądania przekierowania do innego zasobu aplikacji internetowej np. do usług internetowych lub generowanie odpowiedzi, która nie zawiera komponentów typu JavaServer Faces, wówczas program w celu pominięcia fazy **RenderResponse** musi wywołać metodę **facesContext.responseComplete**

# Opis faz cyklu życia JavaServer Faces (7)

## – Render Response:

- ustalenie zawartości strony w przypadku cyklu typu **postback** - komunikaty jako normalna reakcja
- lub komunikaty o błędach, pochodzące z faz **ApplyRequest**, **ProcessValidations** lub **UpdateModel**. Jeśli strona zawiera znaczniki **h:message** lub **h:messages**, błędy wyświetlane są za pomocą renderowania typ znaczników
- ustalenie nowej zawartości kontenera JSP zawierającego drzewo komponentów UI – przebudowanie istniejącego **widoku strony** utworzonego podczas **Restore View** dla cyklu typu **postbacks**

# Częściowa cykl życia aplikacji typu JSF oraz częściowy rendering

- Takie przetwarzanie jest możliwe,



# Przykład 2 aplikacji typu JavaServer Faces

# Definicja formularza – domyślna strona startowa index.xhtml po edycji

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h:form>
      <h2> Hello from Facelets </h2>
      <h:inputText
        id="name"
        title="Imie: "
        value="#{personalia.name}"
        required="true"
        requiredMessage="Bład: Podaj imie."
        maxLength="10"
      />
      <p>
      </p>
      <h:commandButton id="ok"
        value="Ok"
        action="rezultat">
      </h:commandButton>
    </h:form>
  </h:body>
</html>
```

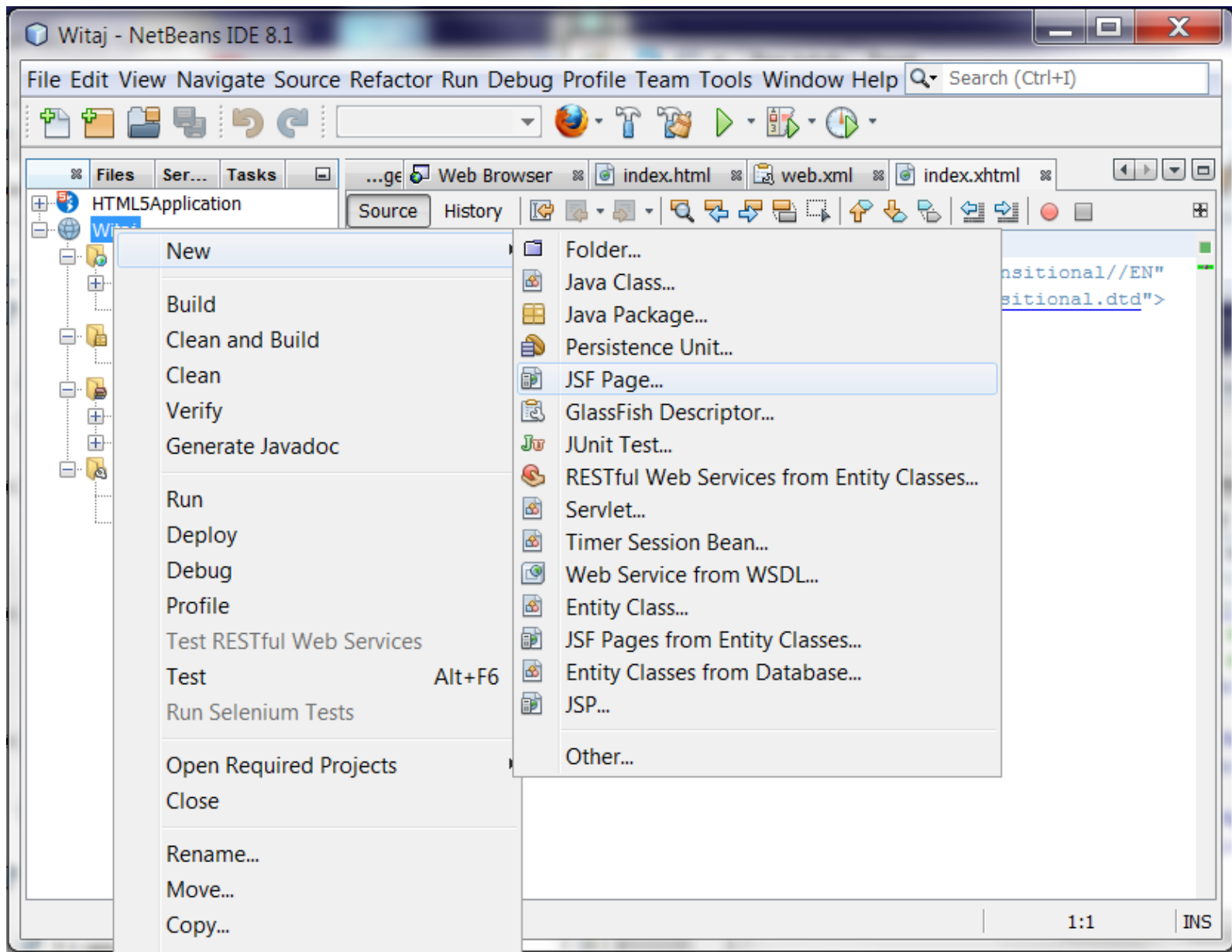
Pole tekstowe wejściowe (**inputText**) przypisuje dane wejściowe do atrybutu **name** obiektu **personalia** (**value**)

Wypełnienie pola jest wymagane (**required="true"**), w przeciwnym wypadku wyświetlany jest komunikat: Bład: Podaj imie. (**requiredMessage**)

Wprowadzone słowo nie może mieć więcej niż 10 znaków (**maxLength**),

Przycisk wywołujący akcję **Request**; w fazie **Response** zwracana jest strona **rezultat.xhtml**

# Dodanie nowej strony xhtml – wybór **New** z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce **Projects**, następnie wybór z kolejnej listy pozycji **JSF Page**



# Nadanie nazwy pliku rezultat (**File Name**) i wybór katalogu pliku (**Location**) i wybór **Next**

**Steps**

1. Choose File Type
- 2. Name and Location**

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

Options:

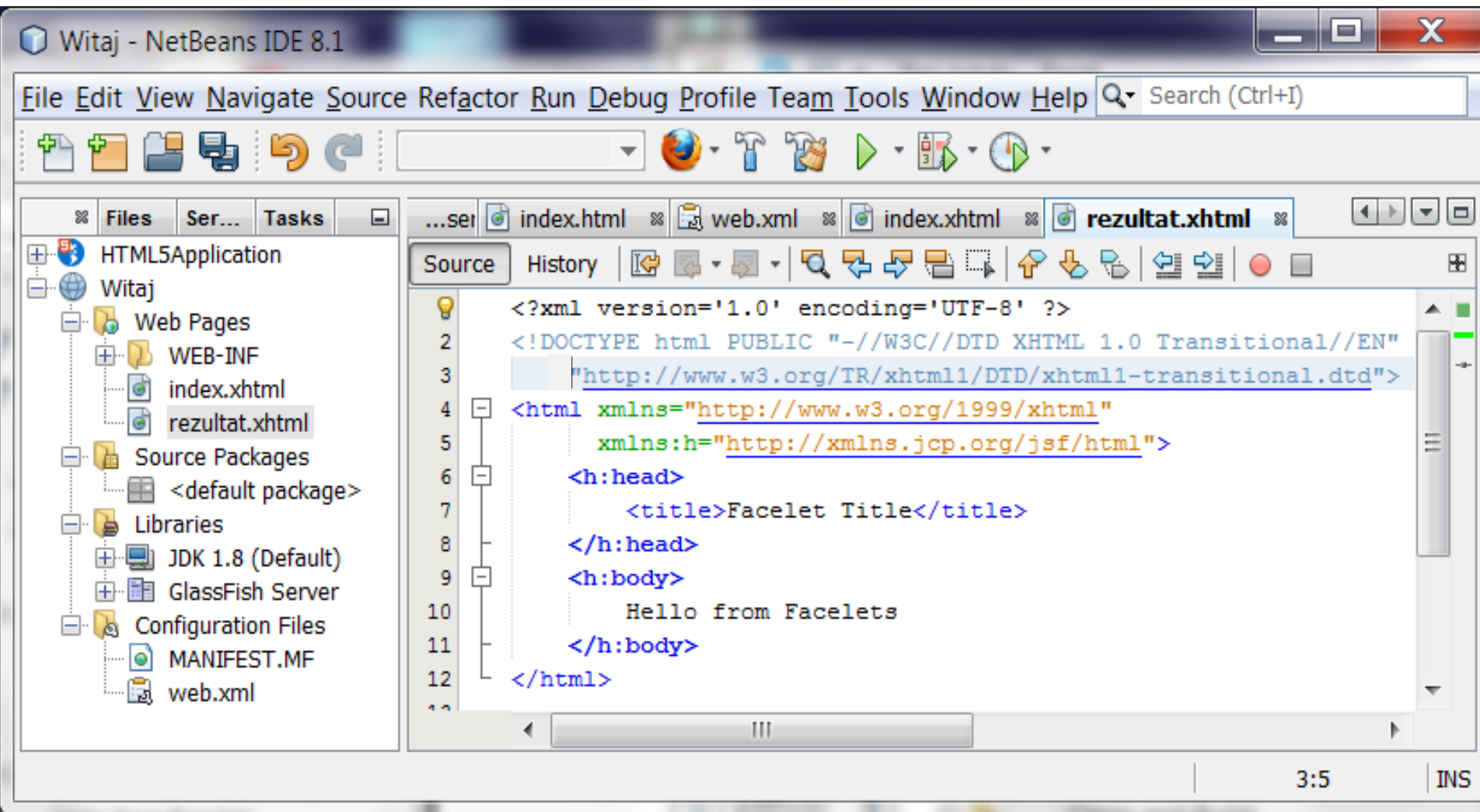
Facelets

JSP File (Standard Syntax)  Create as a JSP Segment

Description:

< Back   Next >   **Finish**   Cancel   Help

# Zawartość pliku rezultat.xhtml

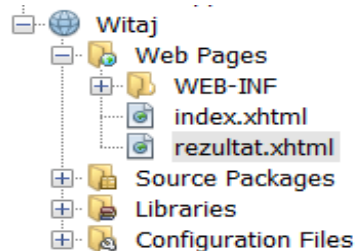


The screenshot shows the NetBeans IDE 8.1 interface. The title bar reads "Witaj - NetBeans IDE 8.1". The menu bar includes "File", "Edit", "View", "Navigate", "Source", "Refactor", "Run", "Debug", "Profile", "Team", "Tools", "Window", and "Help". A search bar on the right contains "Search (Ctrl+I)". The toolbar contains various icons for file operations and development. The left sidebar shows a project tree for "HTML5Application" with a sub-project "Witaj". Under "Witaj", there are "Web Pages" (containing "index.xhtml" and "rezultat.xhtml") and "Source Packages" (containing "<default package>"). The main editor window displays the source code of "rezultat.xhtml". The code is as follows:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    Hello from Facelets
  </h:body>
</html>
```

The status bar at the bottom right shows "3:5" and "INS".

# Edycja strony `rezultat.xhtml` przez uzupełnienie kodu, wyświetlana w fazie `Response` po fazie `Request` ze strony `index.xhtml` – naciśnięcie przycisku OK



Dodanie  
znacznika do  
wyświetlania  
danych na  
stronie:  
**<h:outputLabel>**

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h2> Hello from Facelets </h2>
    <h:form>
      <h
        <h:message>
        <h:messages>
        <h:outputFormat>
        <h:outputLabel>
        <h:outputLink>
        <h:outputScript>
        <h:outputStylesheet>
        <h:outputText>
        <h:panelGrid>
        <h:panelGroup>
        <h:selectBooleanCheckbox>
        <h:selectManyCheckbox>
        <h:selectManyListbox>
        <h:selectManyMenu>
        <h:selectOneListbox>
        <h:selectOneMenu>
        <h:selectOneRadio>
```

Library: `http://xmlns.jcp.org/jsf/html, http://java.sun.com/jsf/html (Html Basic)`

## outputLabel

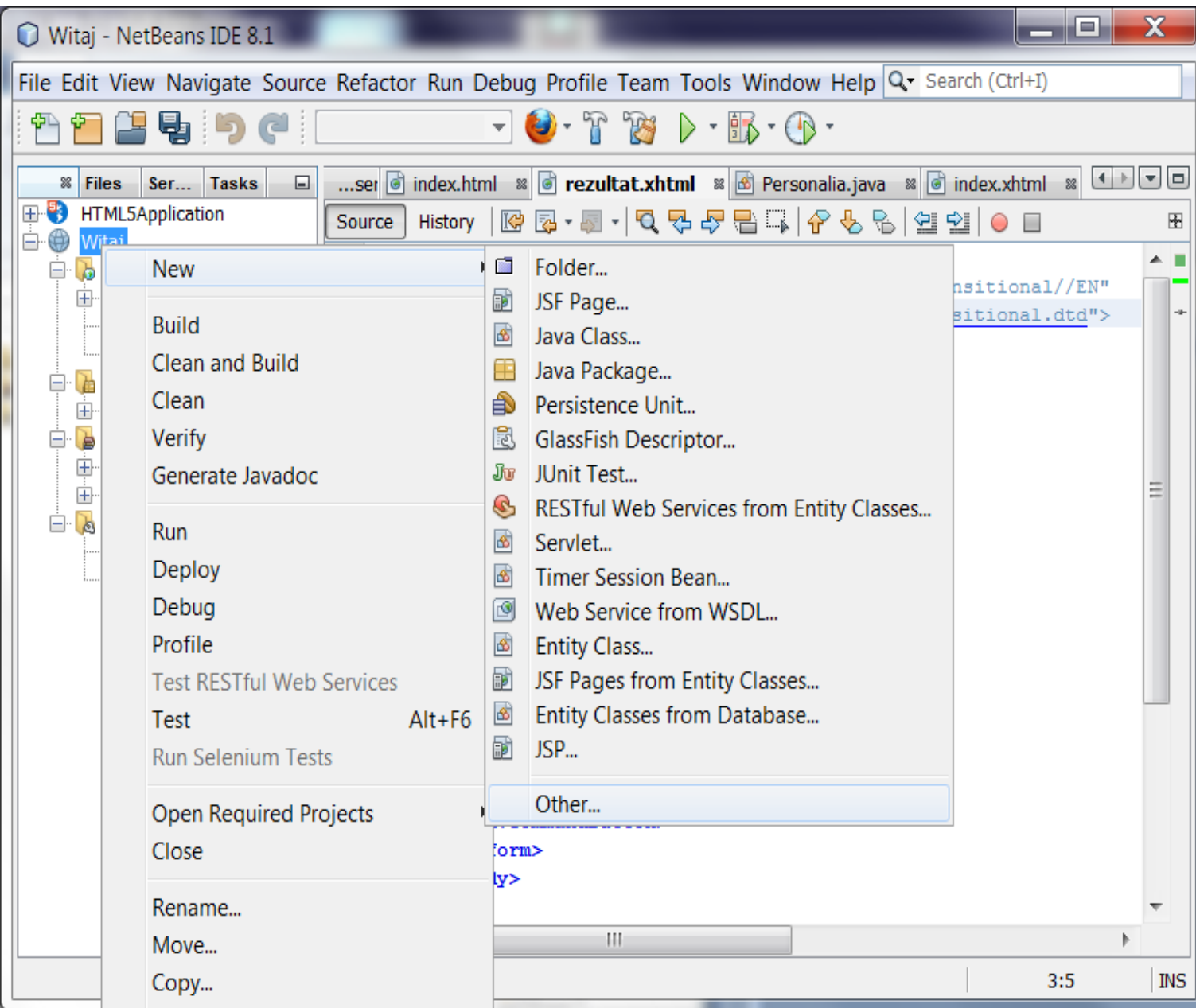
Renders an HTML "label" element. Render the current value of the component as label text if it is specified. If a "for" attribute is specified, find the component specified by the value of the "for" attribute, and render its client id as the value of the "for" attribute. If "styleClass" attribute is specified, render its value as the value of the "class" attribute.

# Strona rezultat.xhtml po edycji

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h2> Hello from Facelets </h2>
    <h:form>
      <h:outputLabel
        id="rezultat"
        title="rezultat"
        value="Witaj: #{personalia.name}">
      </h:outputLabel>
      <p></p>
      <h:commandButton id="Powrot"
        value="Powrot"
        action="index">
      </h:commandButton>
    </h:form>
  </h:body>
</html>
```

Pole tekstowe wyjściowe typu etykieta **outputLabel**, które wyświetla dane pobrane z atrybutu **name** obiektu **personalia** (**value**)

Przycisk wywołujący akcję **Request**; w fazie **Response** zwracana jest strona **index.xhtml**



**Dodanie obiektu typu ManagedBean - wybór New z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce Projects, następnie wybór pozycji Other z kolejnej listy**



# Wybór JavaServer Faces/JSF ManagedBean (2)

New File

**Steps**

1. Choose File Type
2. ...

**Choose File Type**

Project:

Filter:

**Categories:**

- Web
- HTML5/JavaScript
- JavaServer Faces
- Bean Validation
- Struts
- Spring Framework
- Enterprise JavaBeans
- Contexts and Dependenc

**File Types:**

- JSF Page
- JSF Managed Bean
- JSF Faces Configuration
- JSF Composite Component
- JSF Pages from Entity Classes
- JSF Resource Library Contract
- JSF Faces Component
- Facelets Template
- Facelets Template Client

**Description:**

Creates a new managed bean class.

< Back   Next >   Finish   Cancel   Help

## Wybór czasu życia obiektu typu **ManagedBean** (3)

New JSF Managed Bean

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name: Personalia

Project: Witaj

Location: Source Packages

Package: warstwa\_internetowa

Created File: C:\Studia\JSF\JavaPK\Witaj\src\java\Personalia.java

Add data to configuration file

Configuration File:

Name: personalia

Scope:

- dependent
- dependent
- application
- request
- session
- conversation
- flow
- view

26

# Czas życia komponentów typu ManagedBean

- **Application (javax.enterprise.context.ApplicationScoped):** zachowanie danych z interakcji wszystkich użytkowników aplikacji internetowej.
- **Session (javax.enterprise.context.SessionScoped):** zachowanie danych wielu interakcji użytkownika aplikacji internetowej określonych czasem sesji
- **Flow (javax.faces.flows.FlowScoped):** zachowanie danych interakcji użytkownika aplikacji internetowej pochodzących z określonego przepływu danych w tej aplikacji
- **Request (javax.enterprise.context.RequestScoped):** zachowanie danych jednej interakcji „żądanie-odpowiedź” użytkownika aplikacji internetowej
- **Dependent (javax.enterprise.context.Dependent):** czas życia danego komponentu ManagedBean zależy od innych komponentów typu ManagedBean

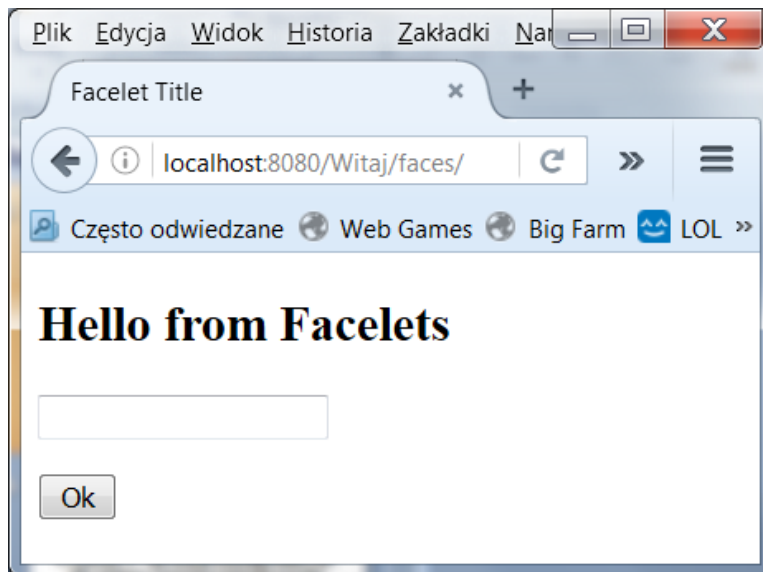
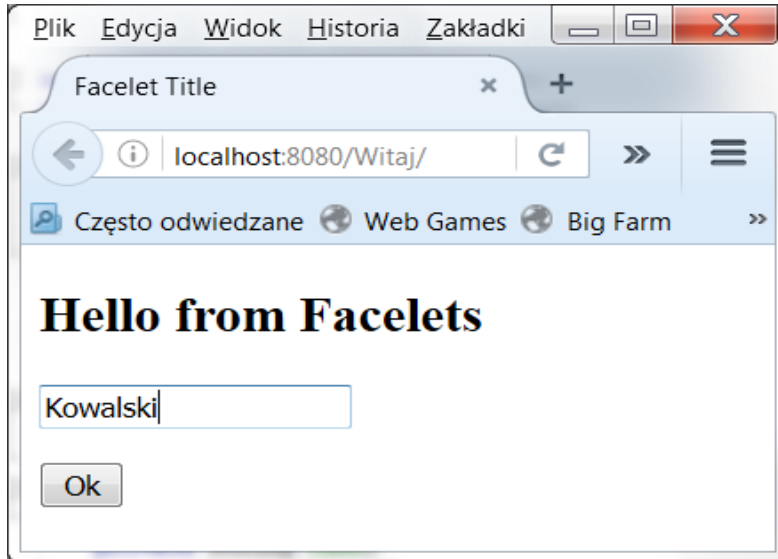
## Zawartość klasy typu ManagedBean (4)

The screenshot displays the NetBeans IDE 8.1 interface. The left sidebar shows a project tree for 'Witaj' under 'Web Pages', with 'Personalia.java' selected in the 'warstwa\_internetowa' source package. The main editor window shows the source code of 'Personalia.java' with the following content:

```
1 package warstwa_internetowa;
2
3 import java.io.Serializable;
4 import javax.faces.bean.ManagedBean;
5 import javax.faces.bean.SessionScoped;
6
7 @ManagedBean
8 @SessionScoped
9 public class Personalia implements Serializable{
10     private static final long serialVersionUID = 5443351151396868724L;
11     public Personalia() { }
12     private String name;
13
14     public String getName() { return name; }
15
16     public void setName(String name) { this.name = name; }
17 }
```

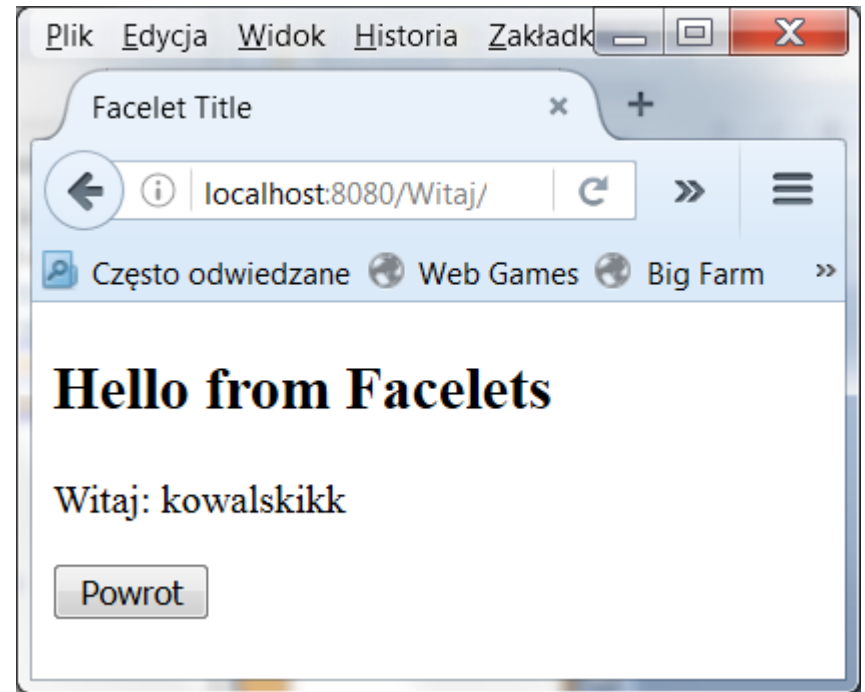
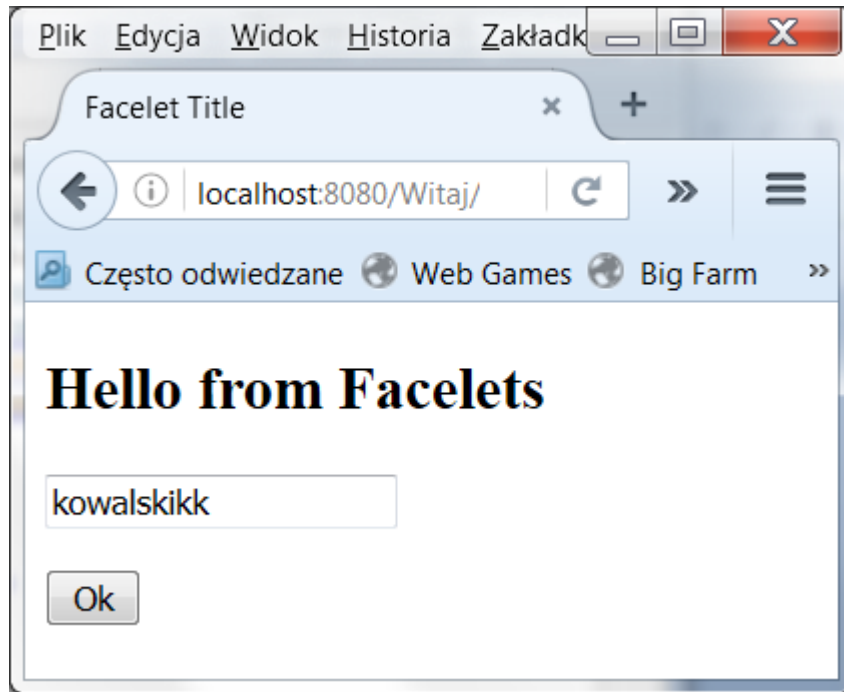
The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar with various icons, and a status bar at the bottom showing the time '14:43' and the text 'INS'.

# Aplikacja po uruchomieniu



Rezultat zastosowania atrybutów w znaczniku `input`:  
`required="true"` oraz `requiredMessage="Bład: Podaj imie."`

# Aplikacja po uruchomieniu



Można wstawić tylko 10 znaków dzięki zastosowaniu atrybutu w znaczniku **input**:  
**maxlength=10**

# Kod HTML formularza index.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head id="j_idt2">
    <title>Facelet Title</title>
  </head>
  <body>
    <form id="j_idt5" name="j_idt5" method="post" action="/Witaj/faces/index.xhtml"
      enctype="application/x-www-form-urlencoded">
      <input type="hidden" name="j_idt5" value="j_idt5" />
      <h2> Hello from Facelets </h2>
      <input id="j_idt5:name" type="text" name="j_idt5:name" maxlength="10" title="Imie: " />
      <p> </p>
      <input id="j_idt5:ok" type="submit" name="j_idt5:ok" value="Ok" />
      <input type="hidden" name="javax.faces.ViewState" id="j_id1:javax.faces.ViewState:0,,
        value="1734791381227717204:-5721288883428752962" autocomplete="off" />
    </form>
  </body>
</html>
```

# Kod HTML formularza rezultat.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head id="j_idt2">
    <title>Facelet Title</title>
  </head>
  <body>
    <h2> Hello from Facelets </h2>
    <form id="j_idt6" name="j_idt6" method="post" action="/Witaj/faces/rezultat.xhtml"
      enctype="application/x-www-form-urlencoded">
      <input type="hidden" name="j_idt6" value="j_idt6" />
      <label id="j_idt6:rezultat" title="rezultat">Witaj: Kowalski</label>
      <p></p>
      <input id="j_idt6:Powrot" type="submit" name="j_idt6:Powrot" value="Powrot" />
      <input type="hidden" name="javax.faces.ViewState"
        id="j_id1:javax.faces.ViewState:0"
        value="1734791381227717204:-8349896159163539121" autocomplete="off" />
    </form>
  </body>
</html>
```



## Dodatek do wykładu 2

### Przykład - aplikacja bazodanowa JSF

<http://netbeans.org/kb/docs/web/jsf20-crud.html>

(podobnie przebiega proces tworzenia aplikacji z wykorzystaniem platformy Java EE 7)

# 1. Tworzenie pustej bazy danych (1)

Zakładanie pustej bazy danych w systemie baz danych Derby: Database Name: Katalogksiazek, User Name: Katalogksiazek, Password: Katalogksiazek

The screenshot shows the NetBeans IDE 7.2 interface. The 'Databases' tree view on the left lists various database connections. A context menu is open over the 'Katalogksiazek' entry, with 'Create Database...' selected. The 'Create Java DB Database' dialog box is open in the foreground, with the following fields:

- Database Name: Katalogksiazek
- User Name: Katalogksiazek
- Password: [masked with dots]
- Confirm Password: [masked with dots]
- Database Location: C:\Users\kruczkiewicz\.netbeans-derby

The 'OK' button is highlighted, indicating the user is ready to create the database.

# 1. Połączenie z pustą bazą danych (2)

The screenshot displays the NetBeans IDE 7.2 interface. The 'Services' tab is active in the left-hand 'Projects' pane, showing a list of database services. The service 'jdbc:derby://localhost:1527/Katalog\_ksiazek1 [Katalog\_ksiazek1 on KATALOG\_KSIAZEK1]' is selected, and a context menu is open over it, with 'Connect...' highlighted. The main editor area shows the 'Output' window for the 'Java DB Database Process', displaying the message: 'DRDA\_SecurityInstalled.I' and 'Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony'. Below the output window is the 'HTTP Server Monitor' window, which is currently empty. The bottom status bar shows the page number '35'.

NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

localhost

Projects Files Services Favorites

Databases

- MySQL Server at localhost:3306 [root] (disconnected)
- Java DB
- Drivers
- jdbc:derby://localhost:1527/alarmsystem [alarm on ALARM]
- jdbc:derby://localhost:1527/consult [consult on CONSULT]
- jdbc:derby://localhost:1527/JsfJpa [JsfJpa on JSFJPA]
- jdbc:derby://localhost:1527/Katalog [katalog on KATALOG]
- jdbc:derby://localhost:1527/Katalog\_Klientow [Katalog\_Klientow on KATALOG\_KLIENTOW]
- jdbc:derby://localhost:1527/Katalog\_ksiazek [Katalog\_ksiazek on KATALOG\_KSIAZEK]
- jdbc:derby://localhost:1527/Katalog\_ksiazek1 [Katalog\_ksiazek1 on KATALOG\_KSIAZEK1]
- jdbc:derby://localhost:1527/Katalog\_ksiazek2 [Katalog\_ksiazek2 on KATALOG\_KSIAZEK2]
- jdbc:derby://localhost:1527/Katalog\_ksiazek3 [Katalog\_ksiazek3 on KATALOG\_KSIAZEK3]
- jdbc:derby://localhost:1527/Katalog\_ksiazek4 [Katalog\_ksiazek4 on KATALOG\_KSIAZEK4]
- jdbc:derby://localhost:1527/Katalog\_ksiazek5 [Katalog\_ksiazek5 on KATALOG\_KSIAZEK5]
- jdbc:derby://localhost:1527/Katalog\_ksiazek6 [Katalog\_ksiazek6 on KATALOG\_KSIAZEK6]
- jdbc:derby://localhost:1527/Katalog\_ksiazek7 [Katalog\_ksiazek7 on KATALOG\_KSIAZEK7]
- jdbc:derby://localhost:1527/Katalog\_ksiazek8 [Katalog\_ksiazek8 on KATALOG\_KSIAZEK8]
- jdbc:derby://localhost:1527/Katalog\_ksiazek9 [Katalog\_ksiazek9 on KATALOG\_KSIAZEK9]
- jdbc:derby://localhost:1527/Katalog\_ksiazek10 [Katalog\_ksiazek10 on KATALOG\_KSIAZEK10]
- jdbc:derby://localhost:1527/Katalog\_ksiazek11 [Katalog\_ksiazek11 on KATALOG\_KSIAZEK11]
- jdbc:derby://localhost:1527/Katalog\_ksiazek12 [Katalog\_ksiazek12 on KATALOG\_KSIAZEK12]
- jdbc:derby://localhost:1527/Katalog\_ksiazek13 [Katalog\_ksiazek13 on KATALOG\_KSIAZEK13]
- jdbc:derby://localhost:1527/Katalog\_ksiazek14 [Katalog\_ksiazek14 on KATALOG\_KSIAZEK14]
- jdbc:derby://localhost:1527/Katalog\_ksiazek15 [Katalog\_ksiazek15 on KATALOG\_KSIAZEK15]
- jdbc:derby://localhost:1527/Katalog\_ksiazek16 [Katalog\_ksiazek16 on KATALOG\_KSIAZEK16]
- jdbc:derby://localhost:1527/Katalog\_ksiazek17 [Katalog\_ksiazek17 on KATALOG\_KSIAZEK17]
- jdbc:derby://localhost:1527/Katalog\_ksiazek18 [Katalog\_ksiazek18 on KATALOG\_KSIAZEK18]
- jdbc:derby://localhost:1527/Katalog\_ksiazek19 [Katalog\_ksiazek19 on KATALOG\_KSIAZEK19]
- jdbc:derby://localhost:1527/Katalog\_ksiazek20 [Katalog\_ksiazek20 on KATALOG\_KSIAZEK20]
- jdbc:derby://localhost:1527/Katalog\_ksiazek21 [Katalog\_ksiazek21 on KATALOG\_KSIAZEK21]
- jdbc:derby://localhost:1527/Katalog\_ksiazek22 [Katalog\_ksiazek22 on KATALOG\_KSIAZEK22]
- jdbc:derby://localhost:1527/Katalog\_ksiazek23 [Katalog\_ksiazek23 on KATALOG\_KSIAZEK23]
- jdbc:derby://localhost:1527/Katalog\_ksiazek24 [Katalog\_ksiazek24 on KATALOG\_KSIAZEK24]
- jdbc:derby://localhost:1527/Katalog\_ksiazek25 [Katalog\_ksiazek25 on KATALOG\_KSIAZEK25]
- jdbc:derby://localhost:1527/Katalog\_ksiazek26 [Katalog\_ksiazek26 on KATALOG\_KSIAZEK26]
- jdbc:derby://localhost:1527/Katalog\_ksiazek27 [Katalog\_ksiazek27 on KATALOG\_KSIAZEK27]
- jdbc:derby://localhost:1527/Katalog\_ksiazek28 [Katalog\_ksiazek28 on KATALOG\_KSIAZEK28]
- jdbc:derby://localhost:1527/Katalog\_ksiazek29 [Katalog\_ksiazek29 on KATALOG\_KSIAZEK29]
- jdbc:derby://localhost:1527/Katalog\_ksiazek30 [Katalog\_ksiazek30 on KATALOG\_KSIAZEK30]
- jdbc:derby://localhost:1527/Katalog\_ksiazek31 [Katalog\_ksiazek31 on KATALOG\_KSIAZEK31]
- jdbc:derby://localhost:1527/Katalog\_ksiazek32 [Katalog\_ksiazek32 on KATALOG\_KSIAZEK32]
- jdbc:derby://localhost:1527/Katalog\_ksiazek33 [Katalog\_ksiazek33 on KATALOG\_KSIAZEK33]
- jdbc:derby://localhost:1527/Katalog\_ksiazek34 [Katalog\_ksiazek34 on KATALOG\_KSIAZEK34]
- jdbc:derby://localhost:1527/Katalog\_ksiazek35 [Katalog\_ksiazek35 on KATALOG\_KSIAZEK35]
- jdbc:derby://localhost:1527/Katalog\_ksiazek36 [Katalog\_ksiazek36 on KATALOG\_KSIAZEK36]
- jdbc:derby://localhost:1527/Katalog\_ksiazek37 [Katalog\_ksiazek37 on KATALOG\_KSIAZEK37]
- jdbc:derby://localhost:1527/Katalog\_ksiazek38 [Katalog\_ksiazek38 on KATALOG\_KSIAZEK38]
- jdbc:derby://localhost:1527/Katalog\_ksiazek39 [Katalog\_ksiazek39 on KATALOG\_KSIAZEK39]
- jdbc:derby://localhost:1527/Katalog\_ksiazek40 [Katalog\_ksiazek40 on KATALOG\_KSIAZEK40]
- jdbc:derby://localhost:1527/Katalog\_ksiazek41 [Katalog\_ksiazek41 on KATALOG\_KSIAZEK41]
- jdbc:derby://localhost:1527/Katalog\_ksiazek42 [Katalog\_ksiazek42 on KATALOG\_KSIAZEK42]
- jdbc:derby://localhost:1527/Katalog\_ksiazek43 [Katalog\_ksiazek43 on KATALOG\_KSIAZEK43]
- jdbc:derby://localhost:1527/Katalog\_ksiazek44 [Katalog\_ksiazek44 on KATALOG\_KSIAZEK44]
- jdbc:derby://localhost:1527/Katalog\_ksiazek45 [Katalog\_ksiazek45 on KATALOG\_KSIAZEK45]
- jdbc:derby://localhost:1527/Katalog\_ksiazek46 [Katalog\_ksiazek46 on KATALOG\_KSIAZEK46]
- jdbc:derby://localhost:1527/Katalog\_ksiazek47 [Katalog\_ksiazek47 on KATALOG\_KSIAZEK47]
- jdbc:derby://localhost:1527/Katalog\_ksiazek48 [Katalog\_ksiazek48 on KATALOG\_KSIAZEK48]
- jdbc:derby://localhost:1527/Katalog\_ksiazek49 [Katalog\_ksiazek49 on KATALOG\_KSIAZEK49]
- jdbc:derby://localhost:1527/Katalog\_ksiazek50 [Katalog\_ksiazek50 on KATALOG\_KSIAZEK50]

Connect... Disconnect Execute Command... Refresh Delete Rename... Properties

Java ME SDK Start Page Start Page

NetBeansIDE

Learn & Discover My NetBeans

News & Tutorials Blogs

Output - Java DB Database Process

```
DRDA_SecurityInstalled.I
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony
```

HTTP Server Monitor

Client and Server		Headers	
Request	Cookies	Session	Context
All Records			
Current Records			
Saved Records			

<No View Available>

35

INS

# 1. Połączenie z pustą bazą danych (3)

The screenshot displays the NetBeans IDE 7.2 interface. The main window is titled 'Java ME SDK Start Page' and 'Start Page'. The 'Databases' tree on the left shows a connection to 'Katalogksiazek' on 'KATALOGKSIAZEK' selected. The 'Output' window shows the server startup logs for Apache Derby. The 'HTTP Server Monitor' window is also visible at the bottom right.

**Projects**

- Databases
  - MySQL Server at localhost:3306 [root] (disconnected)
  - Java DB
  - Drivers
    - jdbc:derby://localhost:1527/alarmsystem [alarm on ALARM]
    - jdbc:derby://localhost:1527/consult [consult on CONSULT]
    - jdbc:derby://localhost:1527/JsfJpa [JsfJpa on JSFJPA]
    - jdbc:derby://localhost:1527/Katalog [katalog on KATALOG]
    - jdbc:derby://localhost:1527/Katalog\_Klientow [Katalog\_Klientow on KATALOG\_KLIENTOW]
    - jdbc:derby://localhost:1527/Katalog\_ksiazek [Katalog\_ksiazek on KATALOG\_KSIAZEK]
    - jdbc:derby://localhost:1527/Katalog\_ksiazek1 [Katalog\_ksiazek1 on KATALOG\_KSIAZEK1]
    - jdbc:derby://localhost:1527/Katalogksiazek [Katalogksiazek on KATALOGKSIAZEK]**
  - APP
  - KATALOGKSIAZEK**
    - Tables
    - Views
    - Procedures
  - NULLID
  - SQLJ
  - SYS
  - SYSCAT
  - SYSCS\_DIAG
  - SYSCS\_UTIL
  - SYSFUN
  - SYSIBM
  - SYSPROC
  - SYSSTAT
- jdbc:derby://localhost:1527/sample [app on APP]

**Output - Java DB Database Process**

```
DRDA_SecurityInstalled.I
Server sieciowy Apache Derby - 10.4.2.1
```

**HTTP Server Monitor**

Headers

Client and Server	
Session	Context
Request	Cookies

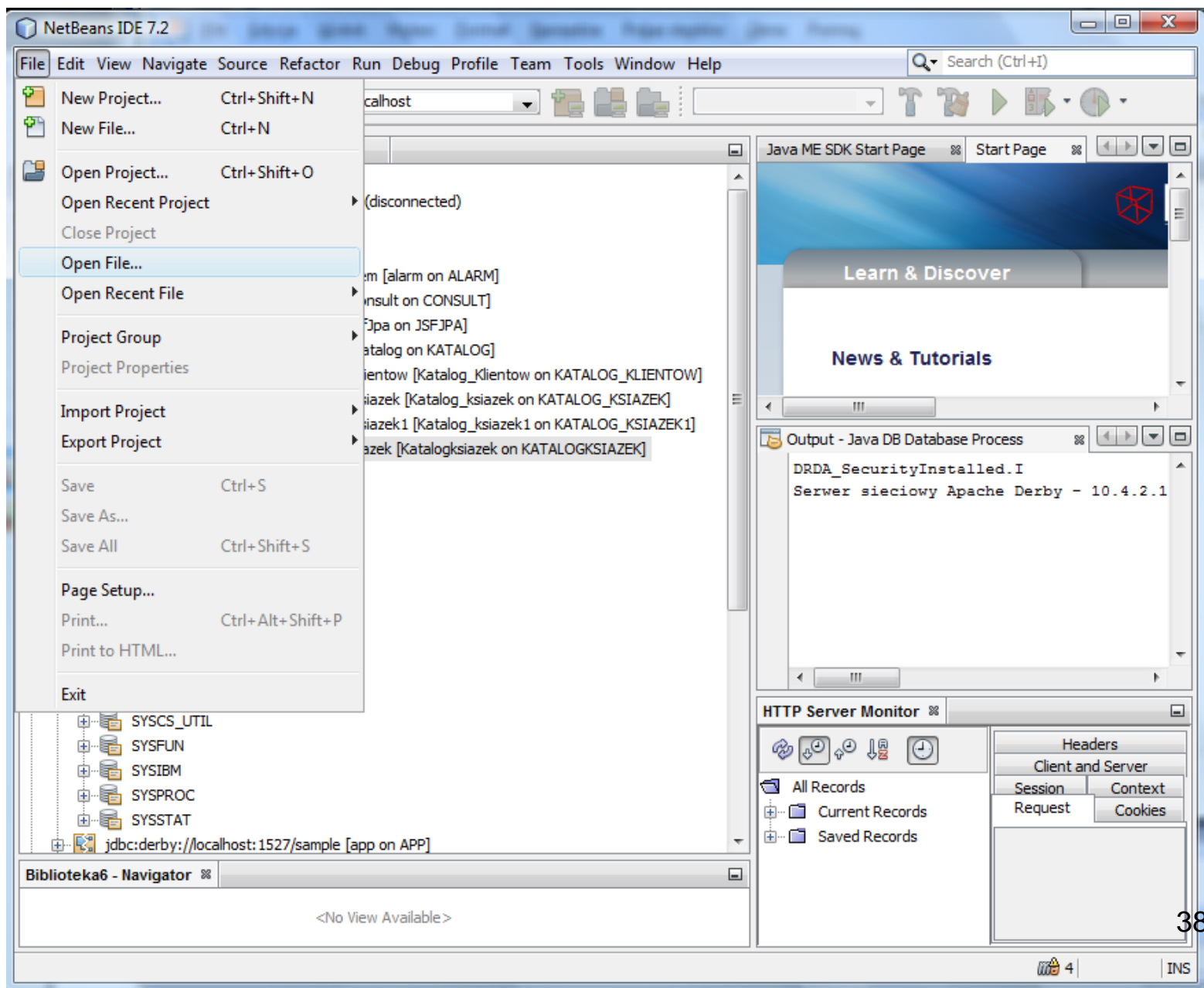
36

# Wykonanie tabel w bazie danych – skrypt typu sql (1)

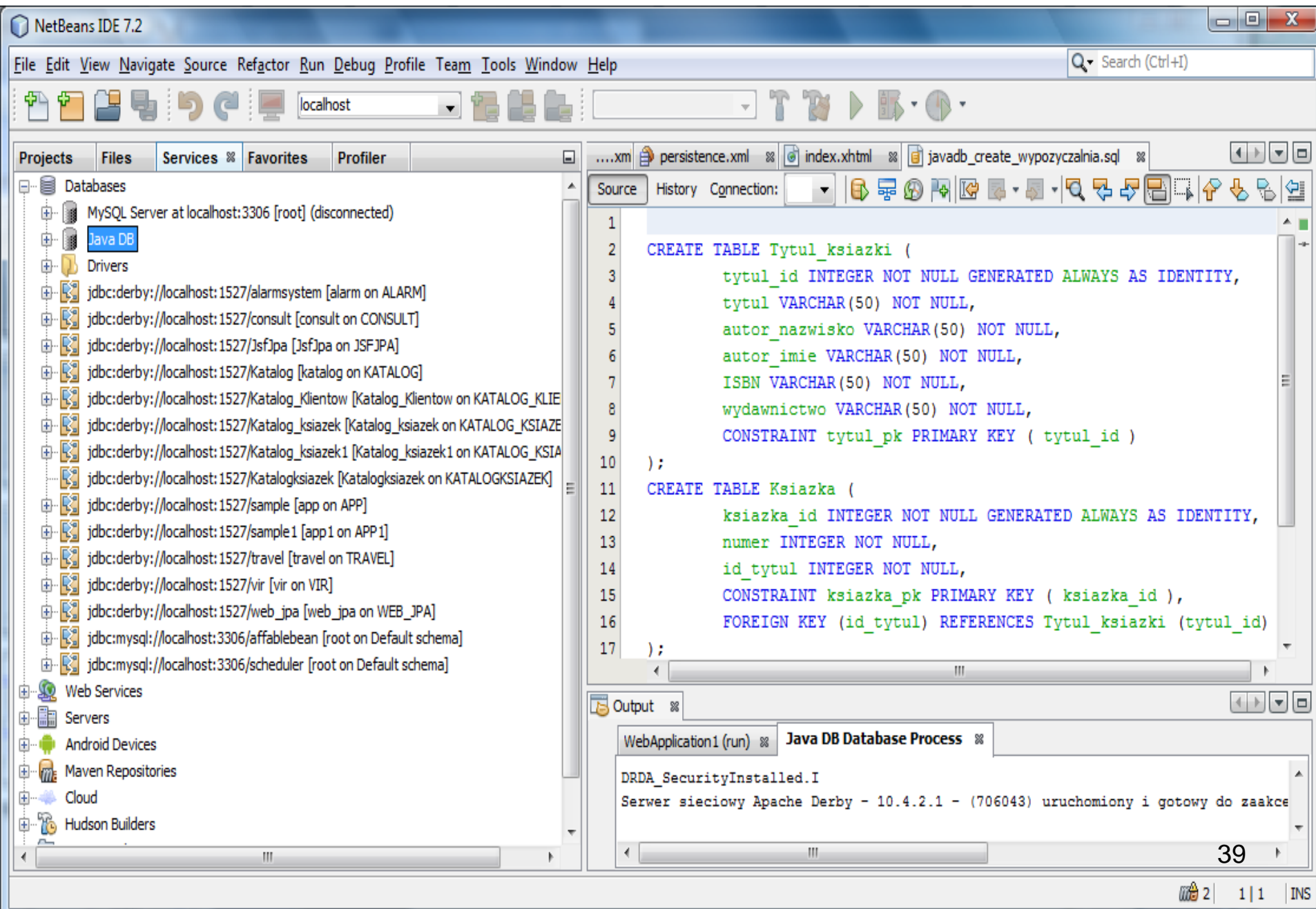
```
CREATE TABLE Tytul_ksiazki (  
  tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
  tytul VARCHAR(50) NOT NULL,  
  autor_nazwisko VARCHAR(50) NOT NULL,  
  autor_imie VARCHAR(50) NOT NULL,  
  ISBN VARCHAR(50) NOT NULL,  
  wydawnictwo VARCHAR(50) NOT NULL,  
  CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )  
);
```

```
CREATE TABLE Ksiazka (  
  ksiazka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
  numer INTEGER NOT NULL,  
  id_tytul INTEGER NOT NULL,  
  CONSTRAINT ksiazka_id PRIMARY KEY ( ksiazka_id ),  
  FOREIGN KEY (id_tytul) REFERENCES Tytul_ksiazki (tytul_id)  
);
```

# Wykonanie tabel w bazie danych (2) – otwieranie pliku zawierającego polecenia SQL do tworzenia tabel



# Wczytanie skryptu typu sql : File/Open File (3)



The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows a SQL script with the following content:

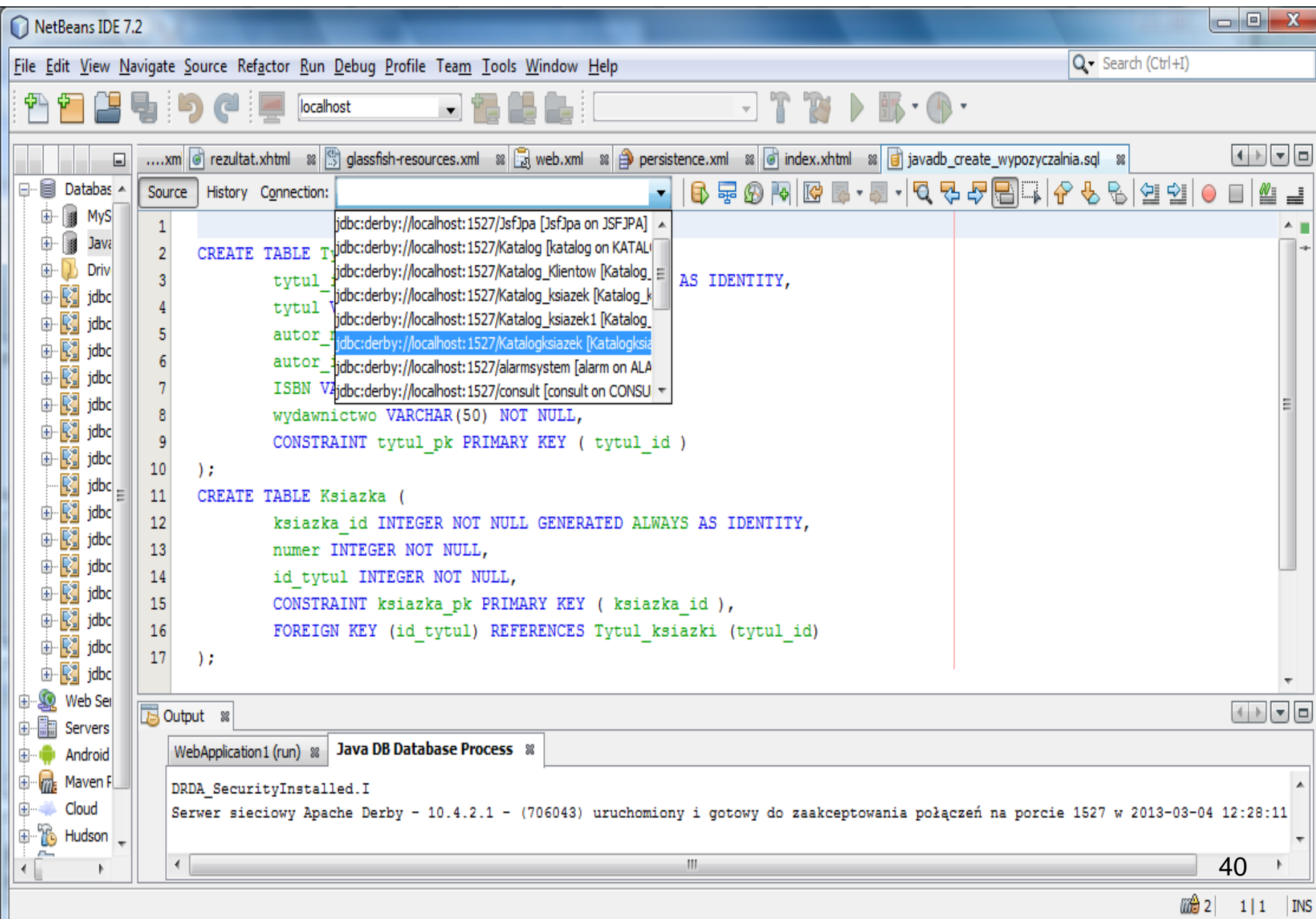
```
1  
2 CREATE TABLE Tytul_książki (  
3     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
4     tytul VARCHAR(50) NOT NULL,  
5     autor_nazwisko VARCHAR(50) NOT NULL,  
6     autor_imie VARCHAR(50) NOT NULL,  
7     ISBN VARCHAR(50) NOT NULL,  
8     wydawnictwo VARCHAR(50) NOT NULL,  
9     CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )  
10 );  
11 CREATE TABLE Książka (  
12     książka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
13     numer INTEGER NOT NULL,  
14     id_tytul INTEGER NOT NULL,  
15     CONSTRAINT książka_pk PRIMARY KEY ( książka_id ),  
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_książki (tytul_id)  
17 );
```

The Output window at the bottom shows the following message:

```
WebApplication1 (run) Java DB Database Process  
DRDA_SecurityInstalled.I  
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakce
```

The page number 39 is visible in the bottom right corner of the IDE window.

# Połączenie skryptu typu sql z bazą danych (4)



The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows a SQL script with the following content:

```
1 jdbc:derby://localhost:1527/JsfJpa [JsfJpa on JSFJPA]
2 CREATE TABLE Tytul_ksiazki (
3     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,
4     tytul VARCHAR(255) NOT NULL,
5     autor VARCHAR(255) NOT NULL,
6     isbn VARCHAR(13) NOT NULL,
7     isbn VARCHAR(13) NOT NULL,
8     wydawnictwo VARCHAR(50) NOT NULL,
9     CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )
10 );
11 CREATE TABLE Ksiazka (
12     ksiazka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,
13     numer INTEGER NOT NULL,
14     id_tytul INTEGER NOT NULL,
15     CONSTRAINT ksiazka_pk PRIMARY KEY ( ksiazka_id ),
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_ksiazki (tytul_id)
17 );
```

The Output window at the bottom shows the following message:

```
WebApplication1 (run) Java DB Database Process
DRDA_SecurityInstalled.I
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania połączeń na porcie 1527 w 2013-03-04 12:28:11
```

The status bar at the bottom right indicates the current page is 40 out of 111, with the text "INS" next to it.



# Uruchomienie skryptu typu sql (5)

The screenshot shows the NetBeans IDE 7.2 interface. The main editor window displays an SQL script named `javadb_insert_data_wypożyczalnia.sql`. The script contains two `CREATE TABLE` statements. The first statement defines a table `Tytul_książki` with columns `tytul_id` (INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY) and `tytul` (VARCHAR(50) NOT NULL). The second statement defines a table `Książka` with columns `książka_id` (INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY), `numer` (INTEGER), `id_tytul` (INTEGER), and a foreign key constraint `FOREIGN KEY (id_tytul) REFERENCES Tytul_książki (tytul_id)`.

A context menu is open over the script, with the `Run File` option (Shift+F6) highlighted. Other options include `Navigate`, `Format` (Alt+Shift+F), `Run Statement`, `SQL History` (Ctrl+Alt+Shift+H), `Cut` (Ctrl+X), `Copy` (Ctrl+C), `Paste` (Ctrl+V), and `Select in Projects`.

The Output window at the bottom shows the execution results for `Java DB Database Process`. The output text is: `DRDA_SecurityInstalled.I` and `Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania połączeń na porcie`.

The status bar at the bottom right shows the page number 41, a magnifying glass icon, and the text 2 | 4 | 23 | INS.

# Utworzenie tabel za pomocą skryptu typu sql (6)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the 'Projects' pane shows a database project named 'jdbc:derby://localhost:1527/Katalogksiazek'. Underneath, a 'Tables' folder contains a 'KSIAZKA' table with columns: KSIAZKA\_ID, NUMER, ID\_TYTUL, and a 'TYTUL\_KSIAZKI' table with columns: TYTUL\_ID, TYTUL, AUTOR\_NAZWISKO, AUTOR\_IMIE, ISBN, and WYDAWNICTWO. The main editor window shows a SQL script with the following content:

```
1  
2 CREATE TABLE Tytul_ksiazki (  
3     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
4     tytul VARCHAR(50) NOT NULL,  
5     autor_nazwisko VARCHAR(50) NOT NULL,  
6     autor_imie VARCHAR(50) NOT NULL,  
7     ISBN VARCHAR(50) NOT NULL,  
8     wydawnictwo VARCHAR(50) NOT NULL,  
9     CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )  
10 );  
11 CREATE TABLE Ksiazka (  
12     ksiazka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
13     numer INTEGER NOT NULL,  
14     id_tytul INTEGER NOT NULL,  
15     CONSTRAINT ksiazka_pk PRIMARY KEY ( ksiazka_id ),  
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_ksiazki (tytul_id)  
17 );
```

The 'Output' window at the bottom shows the execution results for the script 'javadb\_create\_wypozyczalnia.sql':

```
Java DB Database Process  javadb_create_wypozyczalnia.sql execution  
  
Executed successfully in 0,41 s, 0 rows affected.  
Line 2, column 1  
  
Executed successfully in 0,643 s, 0 rows affected.  
Line 11, column 1  
  
Execution finished after 1,053 s, 0 error(s) occurred.
```

# Utworzone tabele oraz możliwe do utworzenia klasy typu Entity

Tabela	Entity	Opis
Tytul_ksiazki	Tytul_ksiazki	Przechowują dane tytułu książki; relacja 1 do wiele z tabelą (encją) Ksiazka
Ksiazka	Ksiazka	Przechowują numer książki Relacja wiele do jeden z tabelą (encją) Tytul_ksiazki

## Dodanie danych do tabel za pomocą skryptu typu sql (1)

**SET SCHEMA** Katalogksiazek;

**INSERT INTO** Tytul\_ksiazki (tytul, autor\_nazwisko, autor\_imie, ISBN, wydawnictwo)  
**VALUES** ('Krzyzacy', 'Sienkiewicz', 'Henryk', '1234567', 'PWN');

**INSERT INTO** Ksiazka (numer,id\_tytul) **VALUES** (1,1);

## Dodanie danych do tabel za pomocą skryptu typu sql (2)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the Project Explorer shows a database project named 'KATALOGKSIAZEK' with several tables and indexes. The main editor window shows a SQL script named 'javadb\_insert\_data\_wypożyczalnia.sql'. The script contains the following SQL statements:

```
1 SET SCHEMA Katalogksiazek;
2
3 INSERT INTO Tytul_ksiazki (tytul, autor_nazwisko, autor_imie, ISBN, wydawnictwo)
4     VALUES ('Kryzacy', 'Sienkiewicz', 'Henryk', '1234567', 'PWN');
5
6 INSERT INTO Ksiazka (numer, id_tytul) VALUES (1,1);
```

A context menu is open over the script, with the 'Run File' option highlighted. The menu items and their shortcuts are:

- Navigate
- Format (Alt+Shift+F)
- Run Statement
- SQL History (Ctrl+Alt+Shift+H)
- Run File (Shift+F6)
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Select in Projects

The Output window at the bottom shows the execution results:

```
WebApplication1 (run) | Java DB Database
Executed successfully in 0,059 s
Line 2, column 1

Executed successfully in 0,04 s, 0 rows affected.
Line 11, column 1

Execution finished after 0,099 s, 0 error(s) occurred.
```

The status bar at the bottom right shows the page number 45 and the text '2 | 6 | 27 | INS'.

## Dodanie danych do tabel za pomocą skryptu typu sql (2)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the 'Files' view shows a project structure for 'jdbc:derby://localhost:1527/Katalogksiazek'. Under the 'Tables' folder, there are two tables: 'KSIAZKA' and 'TYTUL\_KSIAZKI'. The 'KSIAZKA' table has columns: KSIAZKA\_ID, NUMER, ID\_TYTUL, and Indexes. The 'TYTUL\_KSIAZKI' table has columns: TYTUL\_ID, TYTUL, AUTOR\_NAZWISKO, AUTOR\_IMIE, ISBN, WYDAWNICTWO, Indexes, and Foreign Keys.

The central 'Source' window shows the following SQL script:

```
1 SET SCHEMA Katalogksiazek;  
2  
3 INSERT INTO Tytul_książki (tytul, autor_nazwisko, autor_imie, ISBN, wydawnictwo)  
4     VALUES ('Krzyzacy', 'Sienkiewicz', 'Henryk', '1234567', 'PWN');  
5  
6 INSERT INTO Książka (numer,id_tytul) VALUES (1,1);
```

The bottom 'Output' window shows the execution results:

```
Java DB Database Process » javadb_create_wypożyczalnia.sql execution » javadb_insert_data_wypożyczalnia.sql execution  
Executed successfully in 0,002 s, 0 rows affected.  
Line 1, column 1  
  
Executed successfully in 0,067 s, 1 rows affected.  
Line 3, column 1  
  
Executed successfully in 0,026 s, 1 rows affected.  
Line 6, column 1
```

The status bar at the bottom right shows '2 | 3 | 1 | INS' and the page number '46'.

# Wykonanie zapytania wyświetlającego zawartość wybranej tabeli

The screenshot shows the NetBeans IDE 7.2 interface. On the left, the 'Projects' pane shows a database connection 'jdbc:derby://localhost:1527/Katalogksi...' with a table 'KSIAZKA' selected. A context menu is open over the table, with 'View Data...' selected. The central pane shows the SQL Command window with the query: `select * from KATALOGKSIAZEK.KSIAZKA`. The bottom pane shows the 'Output' window with the following text: `Executed successfully in 0,109 s. Line 1, column 1 Execution finished after 0,109 s, 0 error(s) occurred.`

ZKA_ID	NUMER	ID_TYTUL
3	1	1

# Zawartość tabeli Tytul\_książki

NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

localhost

Projects Files Ser... Favorit...

jdbc:derby://localhost:1527/Katalogksi

APP

KATALOGKSIAZEK

Tables

KSIAZKA

KSIAZKA\_ID

NUMER

ID\_TYTUL

Indexes

SQL 130302234036

SQL 130302234036

Foreign Keys

SQL 130302234036

TYTUL\_KSIAZKI

TYTUL\_ID

TYTUL

AUTOR\_NAZWISKO

AUTOR\_IMIE

ISBN

WYDAWNICTWO

Indexes

SQL 130302233158

Foreign Keys

Views

Procedures

NULLID

SQLJ

SYS

SYSCAT

SYSCS\_DIAG

Source History Connection: j... SQL Command 1 SQL Command 2

```
1 select * from KATALOGKSIAZEK.TYTUL_KSIAZKI
```

select \* from KATALOGKSIA...

Page Size: 20 Total Rows: 1 Page: 1 of 1 Matching Rows:

#	TYTUL_ID	TYTUL	AUTOR_NAZWISKO	AUTOR_IMIE	ISBN	WYDAWNICTWO
1	1	Krzyżacy	Sienkiewicz	Henryk	1234567	PWN

Output

process javadb\_insert\_data\_wypożyczalnia.sql execution SQL Command 1 execution SQL Command 2 execution

Executed successfully in 0,109 s.  
Line 1, column 1

Execution finished after 0,109 s, 0 error(s) occurred.

48

4 | 1 | 1 | INS



# Zawartość tabeli Ksiazka

NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

localhost

Projects Files Ser... Favorit...

jdbc:derby://localhost:1527/Katalogksi...

APP

**KATALOGKSIAZEK**

Tables

**KSIAZKA**

- KSIAZKA\_ID
- NUMER
- ID\_TYTUL
- Indexes
  - SQL 130302234036
  - SQL 130302234036
- Foreign Keys
  - SQL 130302234036

TYTUL\_KSIAZKI

- TYTUL\_ID
- TYTUL
- AUTOR\_NAZWISKO
- AUTOR\_IMIE
- ISBN
- WYDAWNICTWO
- Indexes
  - SQL 130302233158
- Foreign Keys

Views

Procedures

NULLID

SQLJ

SYS

SYSCAT

SYSCS\_DIAG

SQL Command 1

```
select * from KATALOGKSIAZEK.KSIAZKA
```

select \* from KATALOGKSIAZEK.KSIAZKA

#	KSIAZKA_ID	NUMER	ID_TYTUL
1		3	1

Output

javadb\_insert\_data\_wypożyczalnia.sql execution

```
Executed successfully in 0,109 s.  
Line 1, column 1  
  
Execution finished after 0,109 s, 0 error(s) occurred.
```

49

4 | 1 | 1 | INS

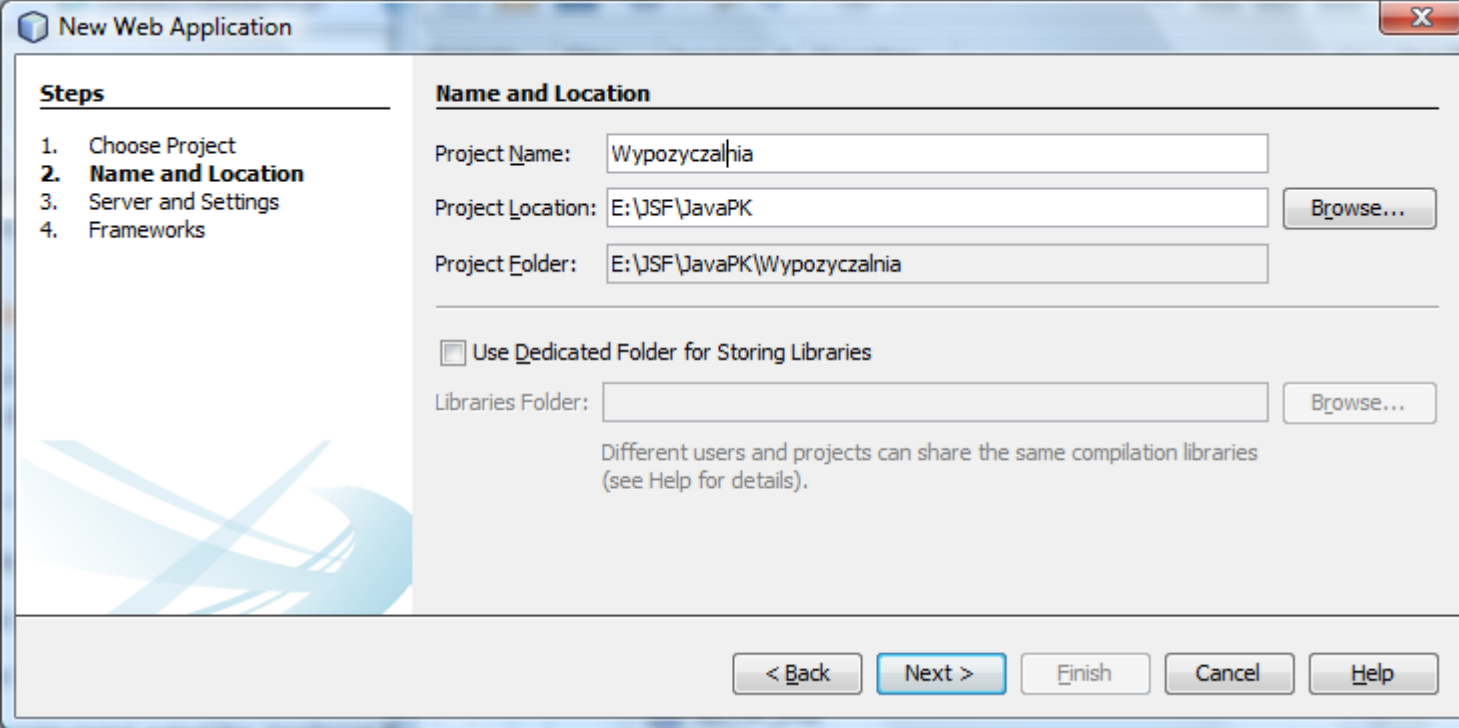
# Tworzenie aplikacji typu JavaServer Faces – File/New Project/Java Web/Web Application (1)

The screenshot displays an IDE interface with the 'File' menu open, highlighting 'New Project...'. The background code editor shows SQL code:

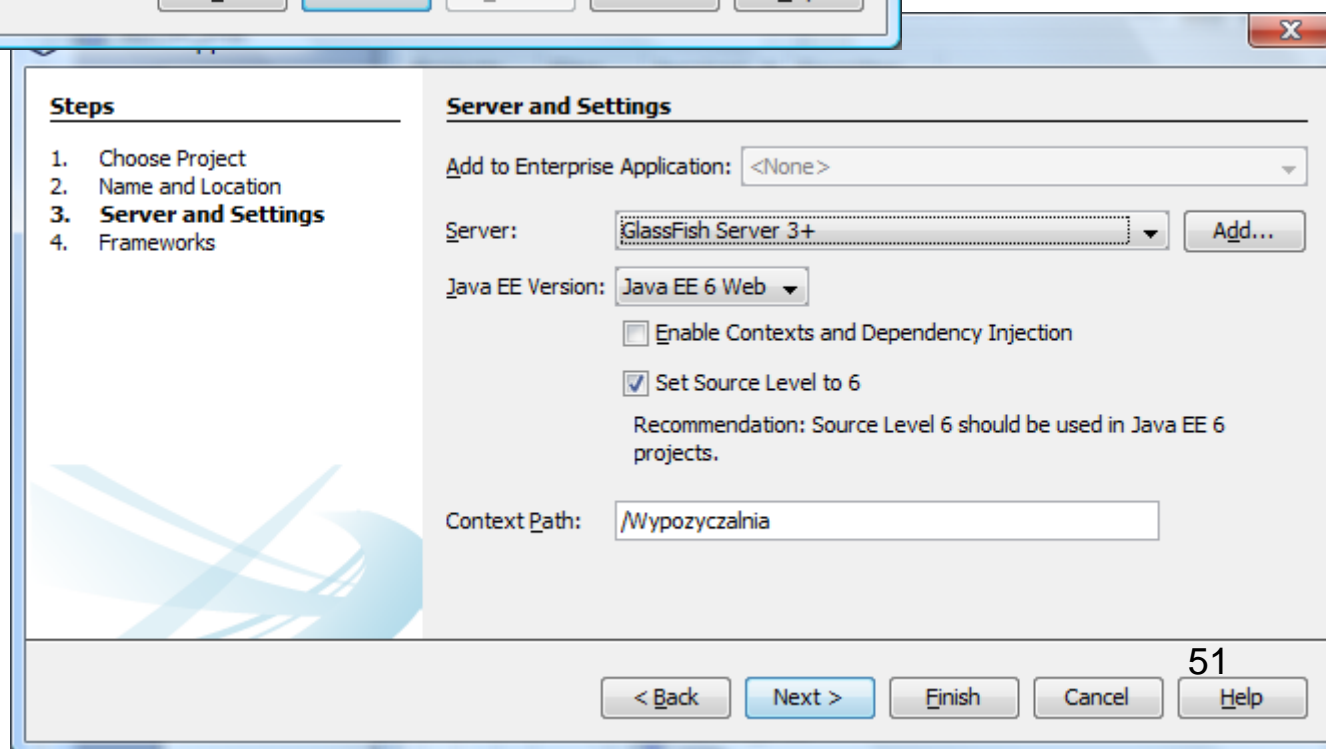
```
1 SET SCHEMA Katalogksiazek;  
2  
3 INSERT INTO Tytul_ksiazki (tytul, autor_nazwisko, autor_  
4 VALUES ('Kryzacy', 'Sienkiewicz', 'Henryk', '123  
5  
6
```

The 'New Project' dialog box is open, showing the 'Choose Project' step. The 'Categories' list includes Java, JavaFX, Java Web, Java EE, Java Card, Java ME, and Maven. The 'Projects' list includes Web Application, Web Application with Existing Sources, and Web Free-Form Application. The 'Web Application' project is selected. The description for 'Web Application' is: 'Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.'

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.



Tworzenie aplikacji internetowej JSF (2) o nazwie Wypozyczalnia



# Tworzenie aplikacji internetowej JSF (3)

New Web Application

**Steps**

1. Choose Project
2. Name and Location
3. Server and Settings
- 4. Frameworks**

**Frameworks**

Select the frameworks you want to use in your web application.

- Spring Web MVC
- JavaServer Faces
- Struts 1.3.10
- Hibernate 3.2.5

JavaServer Faces Configuration

Libraries Configuration Components

Server Library: JSF 2.1

Registered Libraries: Search Libraries...

Create New Library

JSF Folder or JAR:  Browse...

Library Name:

< Back Next > Finish Cancel Help

# Wykonanie klas typu **Entity** z bazy danych Katalogksiazek (1) - wybór **New** z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce **Projects**, następnie wybór z kolejnej listy pozycji **Other**

The screenshot shows an IDE interface with a context menu open over a project named 'Wypożyczalnia' in the 'Projects' view. The menu is divided into two columns. The left column contains standard IDE actions like 'New', 'Build', 'Run', and 'Test'. The right column contains options for generating code from database entities, including 'Entity Classes from Database...', which is highlighted. Below this, the 'Other...' option is also highlighted. The background shows a code editor with XML content and a 'Palette' window on the right with various UI components.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

localhost

Files Servic... FAVORI... ...age Start Page index.xhtml

Test\_Fit\_S  
TestingBot  
Watki\_z\_s  
WebApplic  
weld-gues  
Witaj  
Wykres\_sl  
Wypożycz  
Web P  
W  
inc  
Source  
<c  
Librari  
JD  
Gl  
Config  
MA  
we

New  
Build  
Clean and Build  
Clean  
Verify  
Generate Javadoc  
Run  
Deploy  
Debug  
Profile  
Test RESTful Web Services  
Test Alt+F6  
Open Required Projects  
Close  
Rename...  
Move...  
Copy...  
Delete Delete  
Find... Ctrl+F  
Inspect and Transform...  
Versioning

Validation Constraint...  
JSF Managed Bean...  
JSF Page...  
Java Class...  
Java Package...  
JavaBeans Component...  
JFrame Form...  
Property Editor...  
Frame Form...  
JSF Pages from Entity Classes...  
Entity Classes from Database...  
Facelets Template Client...  
Facelets Template...  
JSF Page...  
TestNG Test Case...  
Other...

```
<?xml version='1.0' encoding='UTF-8' ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" xml:base="http://www.w3.org/2002/xhtml" >  
<head>  
<title>  
</title>
```

HTML  
Table Ordered List  
Unordered List Image  
Link Meta data  
HTML Forms  
Form  
Text Input  
Multi-line Input  
Drop-down List  
Checkbox  
Radio Button  
File Select  
Button  
JSF  
Metadata  
JSF Form  
JSF Form From Entity  
JSF Data Table  
JSF Data Table From Entity

4 | 8 | 13 | INS

# Wykonanie klas typu **Entity** z bazy danych Katalogksiazek (2) – wybór **Persistence/Entity Classes from Database**

The screenshot shows the 'New File' wizard in an IDE. The 'Project' field is set to 'Wypożyczalnia'. Under 'Categories', 'Persistence' is selected. Under 'File Types', 'Entity Classes from Database' is selected. The description at the bottom explains that this template creates Java Persistence API entity classes based on an existing relational database.

**Steps**

1. **Choose File Type**
2. ...

**Choose File Type**

Project:

**Categories:**

- Contexts and Dependency
- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Selenium
- Persistence**
- Groovy
- Hibernate

**File Types:**

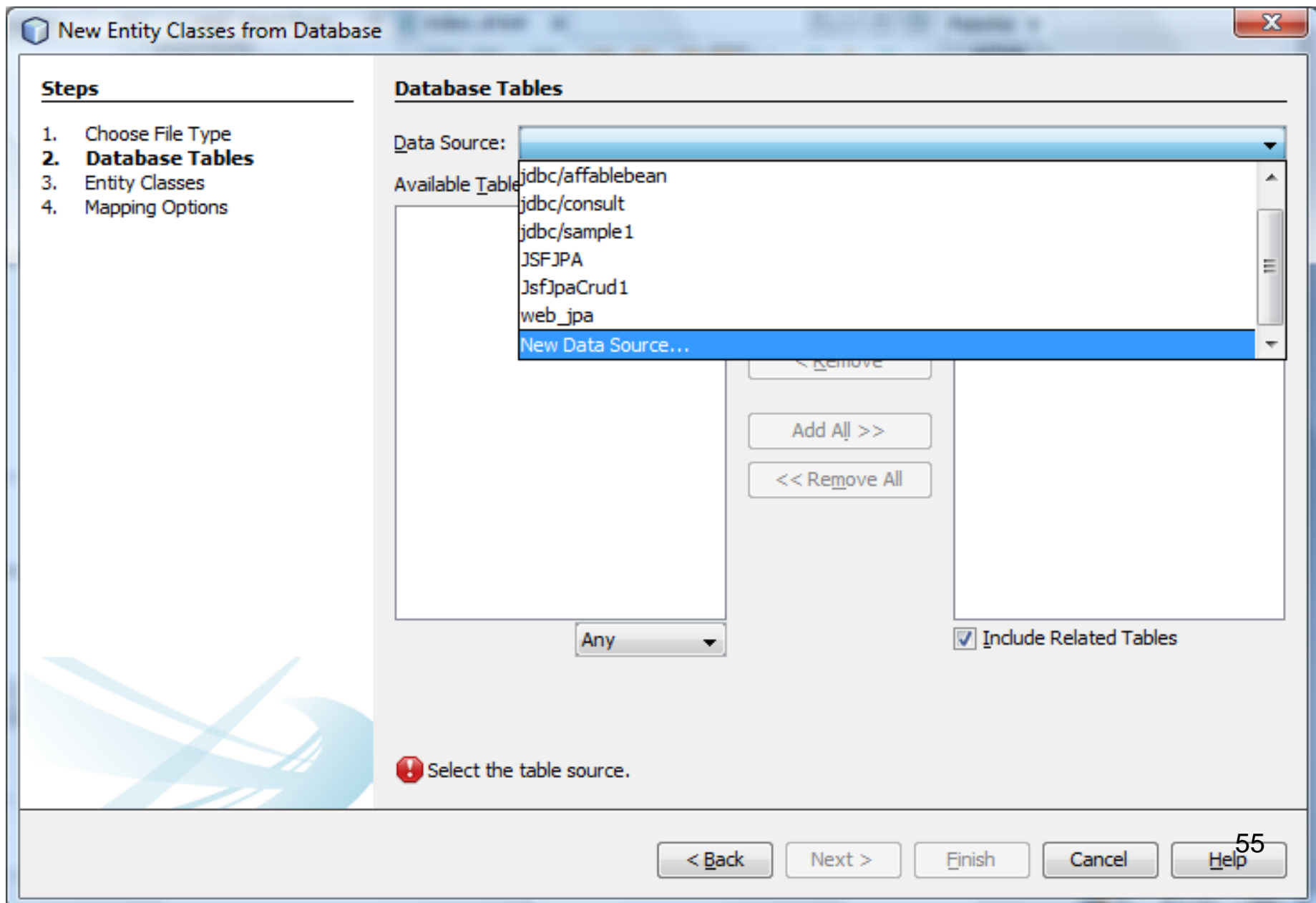
- Entity Class
- Entity Classes from Database**
- JPA Controller Classes from Entity Classes
- JSF Pages from Entity Classes
- Persistence Unit
- Database Schema
- Session Beans For Entity Classes
- RESTful Web Services from Entity Classes
- RESTful Web Services from Database

**Description:**

Creates Java Persistence API entity classes based on an existing relational database. Entity classes are used to represent objects whose lifespan is longer than a typical program execution. This template creates an entity class for each selected table, complete with named query annotations, fields representing columns, and relationships representing foreign keys.

< Back   **Next >**   Finish   Cancel   Help

# Wykonanie klas typu **Entity** z bazy danych (3)– wykonanie odniesienia typu JNDI do bazy danych Katalogsiazek: **New Data Source**




## Wykonanie klas typu Entity z bazy danych (4) – Wybór połączenia z bazą danych Katalogksiazek (**Database Connection**) oraz podanie nazwy JNDI (**JNDI Name**)

Create Data Source

JNDI Name:

Database Connection:

 JNDI name cannot

- jdbc:derby://localhost:1527/Katalog [katalog on KATALOG]
- jdbc:derby://localhost:1527/Katalog\_Klientow [Katalog\_Klientow on KATALOG\_KLIENTOW]
- jdbc:derby://localhost:1527/Katalog\_ksiazek [Katalog\_ksiazek on KATALOG\_KSIAZEK]
- jdbc:derby://localhost:1527/Katalog\_ksiazek1 [Katalog\_ksiazek1 on KATALOG\_KSIAZEK1]
- jdbc:derby://localhost:1527/Katalogksiazek [Katalogksiazek on KATALOGKSIAZEK]**
- jdbc:derby://localhost:1527/sample [app on APP]

Create Data Source

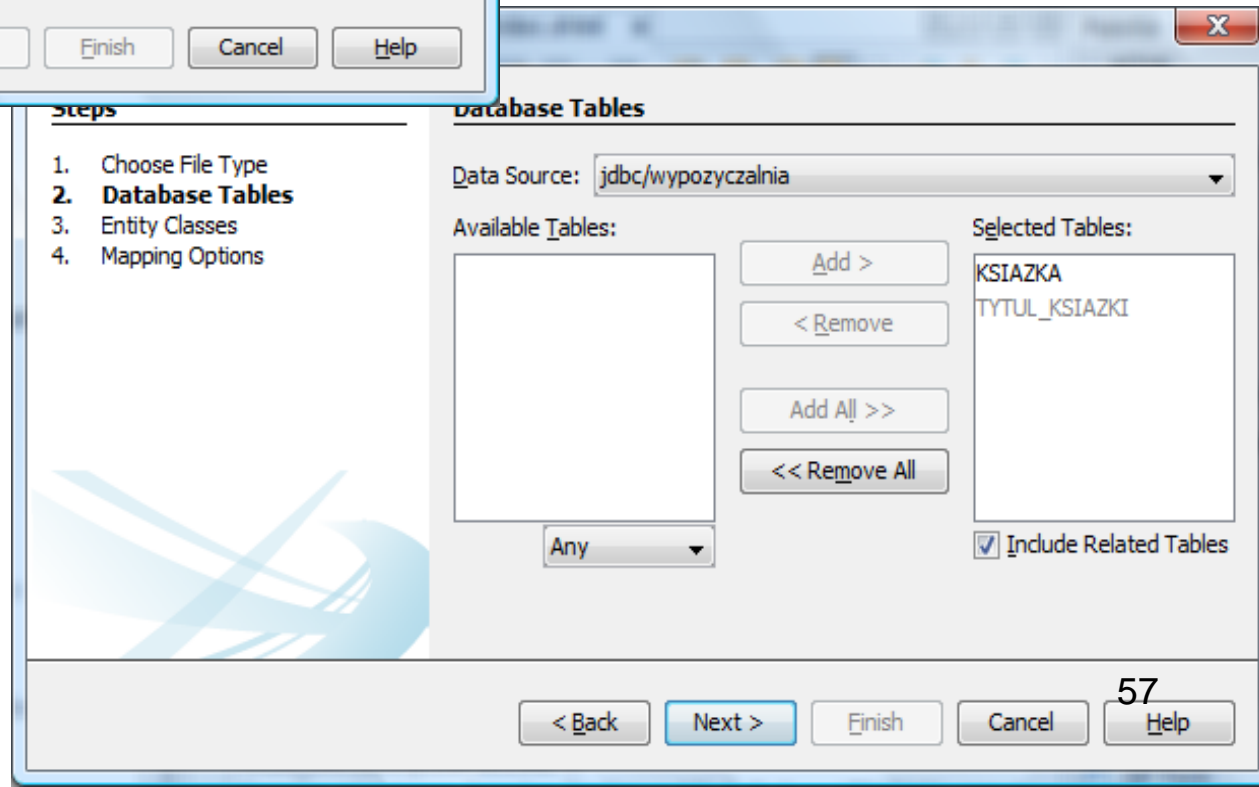
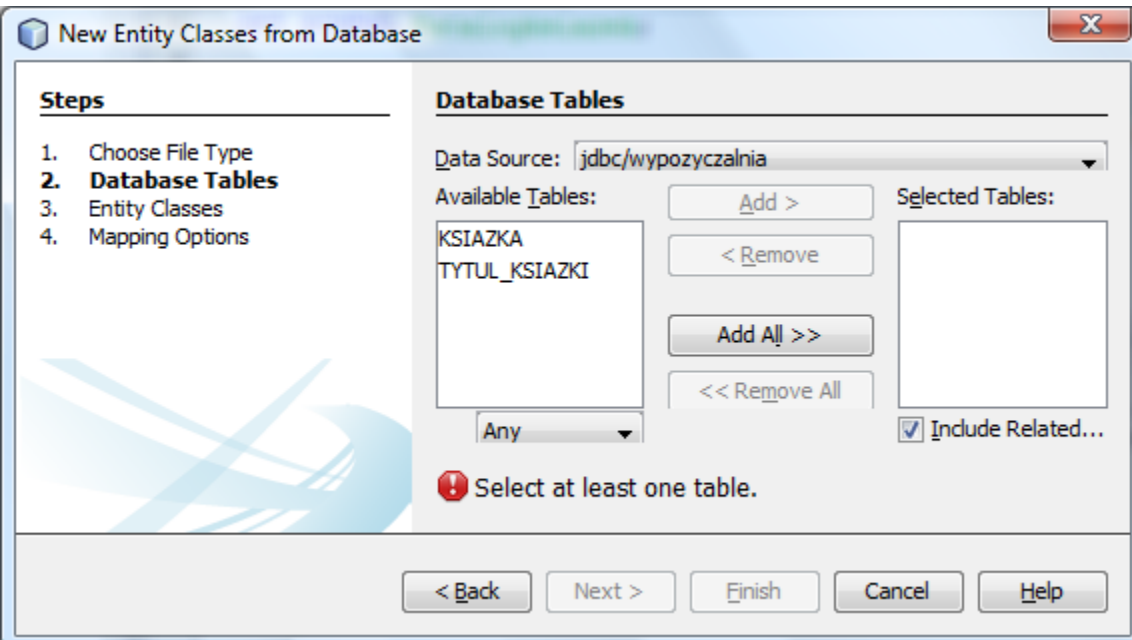
JNDI Name:

Database Connection:

OK Cancel Help



# Wybór tabel do mapowania - wybór **Add All**



# Wykonanie klas typu Entity z bazy danych (5)

New Entity Classes from Database

**Steps**

1. Choose File Type
2. Database Tables
3. **Entity Classes**
4. Mapping Options

**Entity Classes**

Specify the names and the location of the entity classes.

Class Names:

Database Table	Class Name	Generation Type
KSIAZKA	Ksiazka	New
TYTUL_KSIAZKI	TytulKsiazki	New

Project: Wypożyczalnia

Location: Source Packages

Package: jpa

Generate Named Query Annotations for Persistent Fields

Generate JAXB Annotations

Create Persistence Unit

< Back   Next >   Finish   Cancel   Help

# Wygenerowana klasa typu **Entity** Ksiazka

```
package jpa;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.xml.bind.annotation.XmlRootElement;
```

@Entity

@Table(name = "KSIAZKA")

@XmlRootElement

@NamedQueries({

    @NamedQuery(name = "Ksiazka.findAll", query = "SELECT k FROM Ksiazka k"),

    @NamedQuery(name = "Ksiazka.findByKsiazkald", query = "SELECT k FROM Ksiazka k  
    WHERE k.ksiazkald = :ksiazkald"),

    @NamedQuery(name = "Ksiazka.findByNumer", query = "SELECT k FROM Ksiazka k  
    WHERE k.numer = :numer"))})

**public class** Ksiazka **implements** Serializable {

**private static final** long serialVersionUID = 1L;

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Basic(optional = false)

    @Column(name = "KSIAZKA\_ID")

**private** Integer ksiazkald;

    @Basic(optional = false)

    @NotNull

    @Column(name = "NUMER")

**private** int numer;

    @JoinColumn(name = "ID\_TYTUL", referencedColumnName = "TYTUL\_ID")

    @ManyToOne(optional = false)

**private** TytulKsiazki idTytul;

```
public Ksiazka() { }
public Ksiazka(Integer ksiazkald) {
    this.ksiazkald = ksiazkald;
}
public Ksiazka(Integer ksiazkald, int numer) {
    this.ksiazkald = ksiazkald;
    this.numer = numer;
}
public Integer getKsiazkald() {
    return ksiazkald;
}
public void setKsiazkald(Integer ksiazkald) {
    this.ksiazkald = ksiazkald;
}
public int getNumer() {
    return numer;
}
public void setNumer(int numer) {
    this.numer = numer;
}
public TytulKsiazki getIdTytul() {
    return idTytul;
}
public void setIdTytul(TytulKsiazki idTytul) {
    this.idTytul = idTytul;
}
}
```

@Override

```
public int hashCode() {  
    int hash = 0;  
    hash += (ksiazkaId != null ? ksiazkaId.hashCode() : 0);  
    return hash;  
}
```

@Override

```
public boolean equals(Object object) {  
    // TODO: Warning - this method won't work in the case the id fields are not set  
    if (!(object instanceof Ksiazka)) {  
        return false;  
    }  
    Ksiazka other = (Ksiazka) object;  
    if ((this.ksiazkaId == null && other.ksiazkaId != null) || (this.ksiazkaId != null &&  
        !this.ksiazkaId.equals(other.ksiazkaId))) {  
        return false;  
    }  
    return true;  
}
```

@Override

```
public String toString() {  
    return "jpa.Ksiazka[ ksiazkaId=" + ksiazkaId + " ]";  
}
```

```
}  
  
Technologie internetowe 2, Zofia  
Kruczkiewicz
```

# Wygenerowana klasa typu **Entity**Tytul\_ksiazki

```
package jpa;
import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;
```

@Entity

@Table(name = "TYTUL\_KSIAZKI")

@XmlRootElement

@NamedQueries({

@NamedQuery(name = "TytulKsiazki.findAll", query = "SELECT t FROM TytulKsiazki t"),

@NamedQuery(name = "TytulKsiazki.findByTytulId", query = "SELECT t FROM TytulKsiazki t WHERE t.tytulId = :tytulId"),

@NamedQuery(name = "TytulKsiazki.findByTytul", query = "SELECT t FROM TytulKsiazki t WHERE t.tytul = :tytul"),

@NamedQuery(name = "TytulKsiazki.findByAutorNazwisko", query = "SELECT t FROM TytulKsiazki t WHERE t.autorNazwisko = :autorNazwisko"),

@NamedQuery(name = "TytulKsiazki.findByAutorImie", query = "SELECT t FROM TytulKsiazki t WHERE t.autorImie = :autorImie"),

@NamedQuery(name = "TytulKsiazki.findByIsbn", query = "SELECT t FROM TytulKsiazki t WHERE t.isbn = :isbn"),

@NamedQuery(name = "TytulKsiazki.findByWydawnictwo", query = "SELECT t FROM TytulKsiazki t WHERE t.wydawnictwo = :wydawnictwo"))})



```

public class TytulKsiazki implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "TYTUL_ID")
    private Integer tytulId;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "TYTUL")
    private String tytul;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "AUTOR_NAZWISKO")
    private String autorNazwisko;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "AUTOR_IMIE")
    private String autorImie;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "ISBN")
    private String isbn;

```

```

@Basic(optional = false)
@NotNull
@Size(min = 1, max = 50)
@Column(name = "WYDAWNICTWO")
private String wydawnictwo;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "idTytul")
private Collection<Ksiazka> ksiazkaCollection

public TytulKsiazki() {
}
public TytulKsiazki(Integer tytulId) {
    this.tytulId = tytulId;
}
public TytulKsiazki(Integer tytulId, String tytul, String autorNazwisko, String autorImie, String isbn,
String wydawnictwo) {
    this.tytulId = tytulId;
    this.tytul = tytul;
    this.autorNazwisko = autorNazwisko;
    this.autorImie = autorImie;
    this.isbn = isbn;
    this.wydawnictwo = wydawnictwo;
}
public String getTytul() {
    return tytul;
}
public void setTytul(String tytul) {
    this.tytul = tytul;
}
}

```

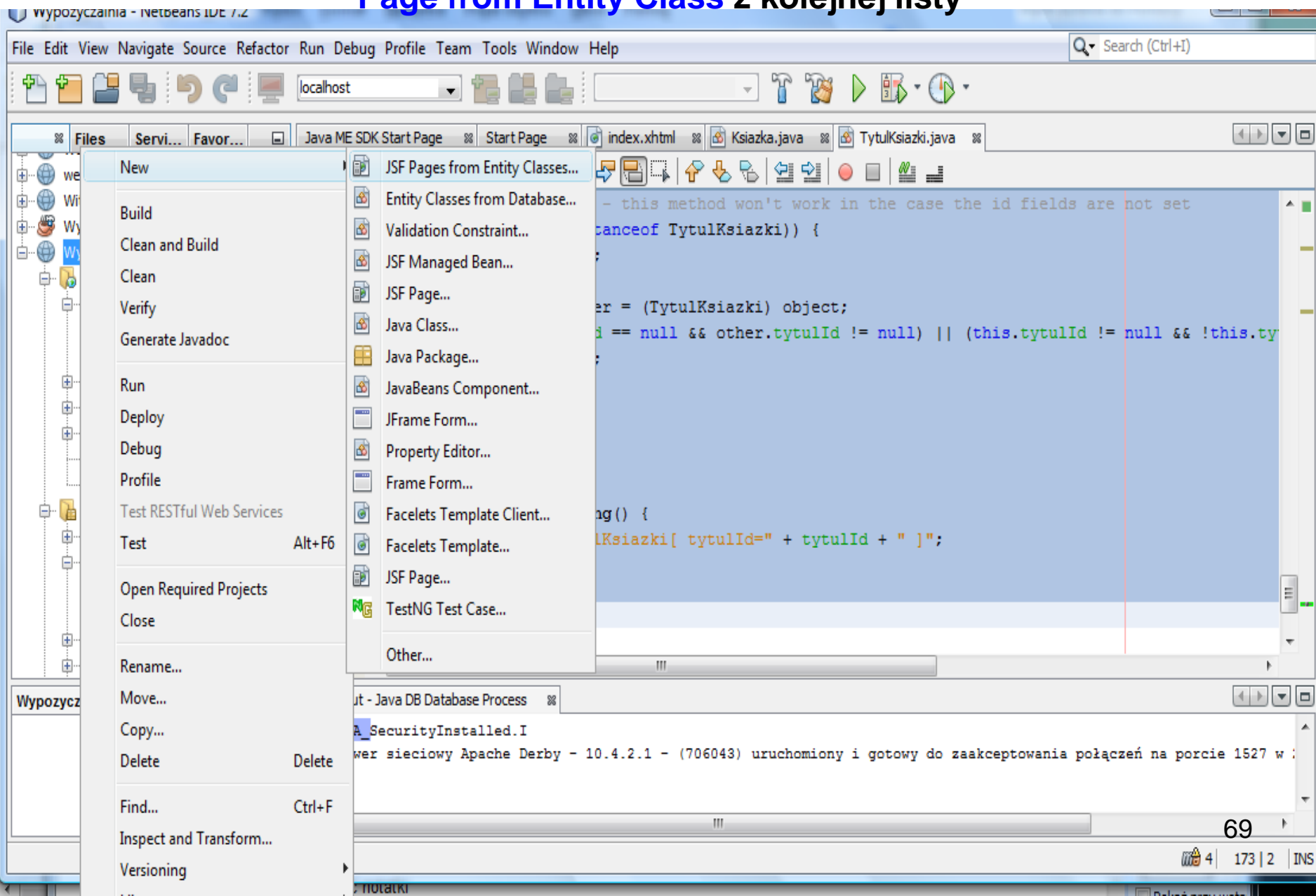
```
public String getAutorNazwisko() {
    return autorNazwisko;
}
public void setAutorNazwisko(String autorNazwisko) {
    this.autorNazwisko = autorNazwisko;
}
public String getAutorImie() {
    return autorImie;
}
public void setAutorImie(String autorImie) {
    this.autorImie = autorImie;
}
public String getIsbn() {
    return isbn;
}
public void setIsbn(String isbn) {
    this.isbn = isbn;
}
public String getWydawnictwo() {
    return wydawnictwo;
}
public void setWydawnictwo(String wydawnictwo) {
    this.wydawnictwo = wydawnictwo;
}
public Integer getTytulId() {
    return tytulId;
}
public void setTytulId(Integer tytulId) {
    this.tytulId = tytulId;
}
}
```

```

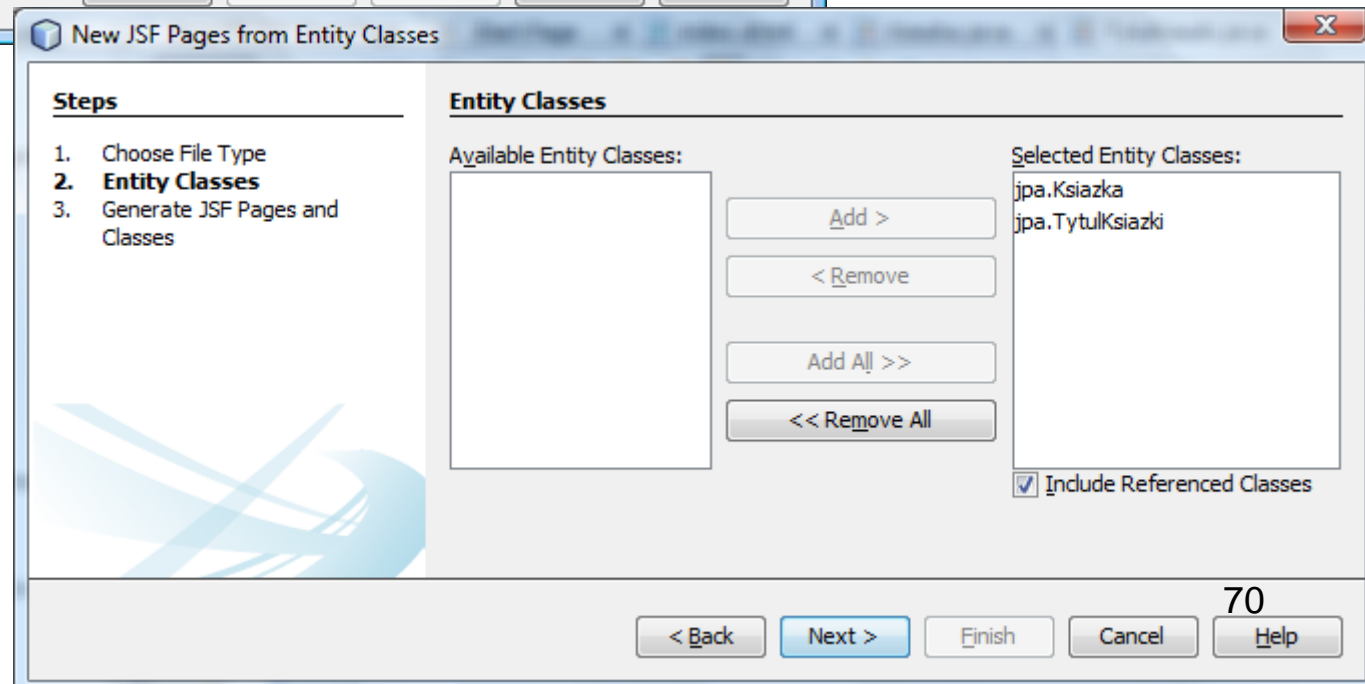
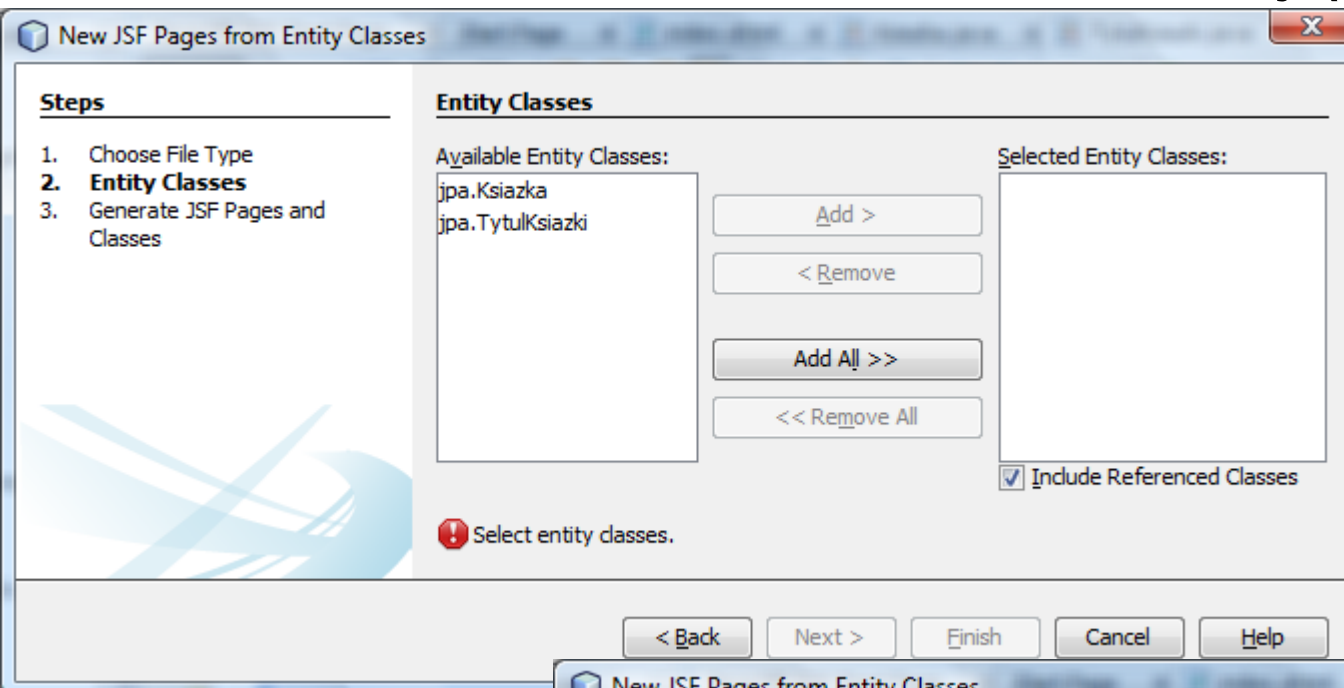
@XmlTransient
public Collection<Ksiazka> getKsiazkaCollection() {
    return ksiazkaCollection;
}
public void setKsiazkaCollection(Collection<Ksiazka> ksiazkaCollection) {
    this.ksiazkaCollection = ksiazkaCollection;
}
@Override
public int hashCode() {
    int hash = 0;
    hash += (tytulId != null ? tytulId.hashCode() : 0);
    return hash;
}
@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof TytulKsiazki)) {
        return false;
    }
    TytulKsiazki other = (TytulKsiazki) object;
    if ((this.tytulId == null && other.tytulId != null) || (this.tytulId != null &&
!this.tytulId.equals(other.tytulId))) {
        return false;
    }
    return true;
}
@Override
public String toString() {
    return "jpa.TytulKsiazki[ tytulId=" + tytulId + " ]";
}
}

```

# Generowanie stron JSF z encji (1) – wybór **New** z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce **Projects**, następnie wybór pozycji **JSP Page from Entity Class** z kolejnej listy



# Generowanie stron JSF z encji (2)



# Generowanie stron JSF z encji (3)

New JSF Pages from Entity Classes

**Steps**

1. Choose File Type
2. Entity Classes
3. **Generate JSF Pages and Classes**

**Generate JSF Pages and Classes**

Specify the package of existing or new EJBs and the package of JSF classes.

Project:

Location:

Session Bean Package:

JSF Classes Package:

Specify the location of new JSF pages.

JSF Pages Folder:

Localization Bundle Name:

Override existing files

Customize Template:

< Back   Next >   **Finish**   Cancel   Help 71

# Zawartość katalogu projektu aplikacji internetowej (1)

The screenshot displays the NetBeans IDE 7.2 interface. The main window shows the source code for `index.xhtml`. The code is as follows:

```
html h:body
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/I
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://java.sun.com/jsf/html">
5     <h:head>
6         <title>Facelet Title</title>
7         <h:outputStylesheet name="css/jsfcrud.css"/>
8     </h:head>
9     <h:body>
10        Hello from Facelets
11        <br />
12        <h:link outcome="/ksiazka/List" value="Show All Ksiazka Items"/>
13        <br />
14        <h:link outcome="/tytulKsiazki/List" value="Show All TytulKsiazki Items"/>
15    </h:body>
16
17 </html>
18
19
20
```

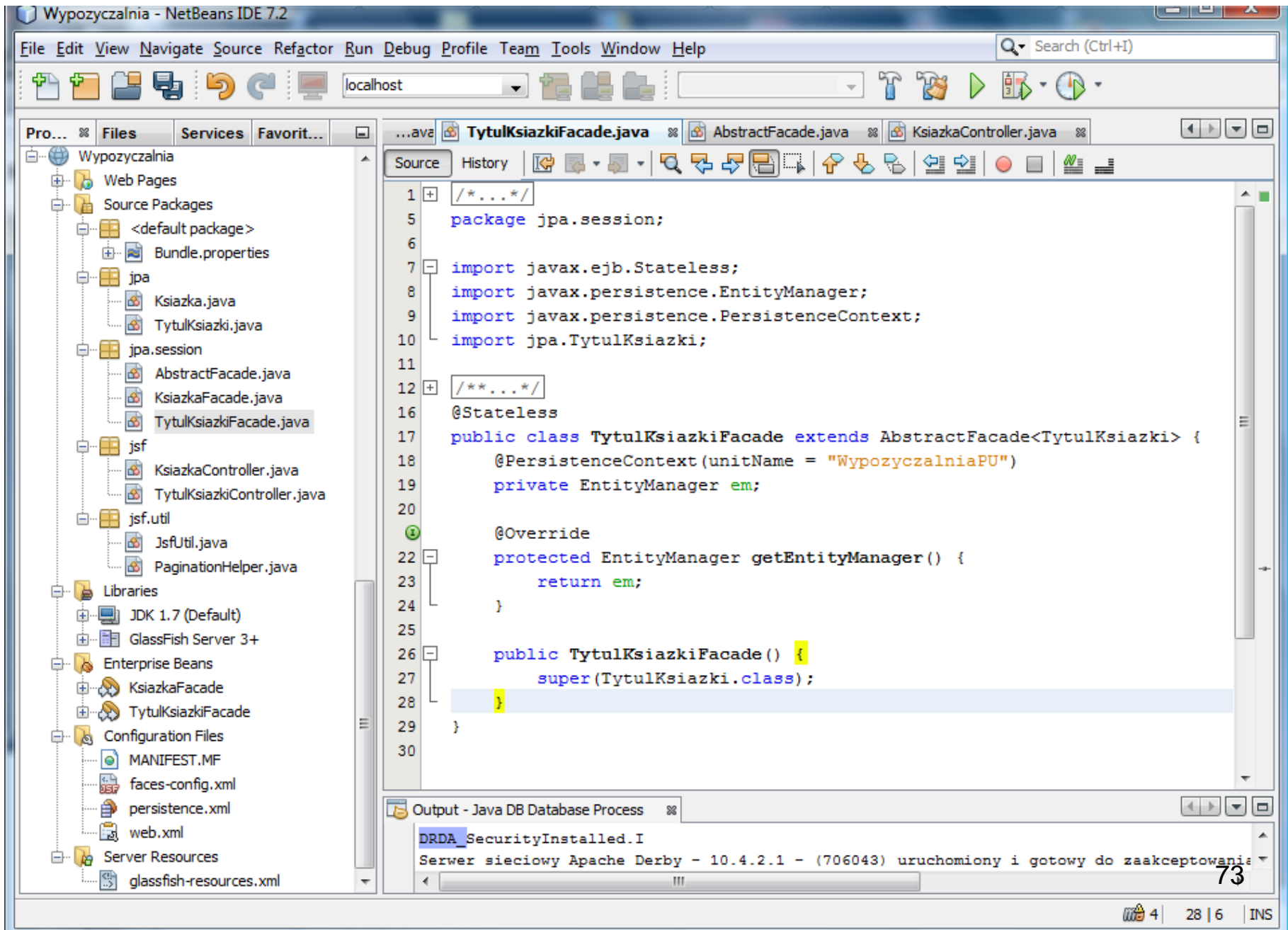
The left sidebar shows the project structure for "Wypożyczalnia". The "Web Pages" folder contains "WEB-INF" (with "faces-config.xml" and "web.xml") and "ksiazka" (with "Create.xhtml", "Edit.xhtml", "List.xhtml", and "View.xhtml"). The "resources" folder contains "css" (with "jsfcrud.css") and "tytulKsiazki" (with "Create.xhtml", "Edit.xhtml", "List.xhtml", and "View.xhtml"). The "index.xhtml" file is selected in the "Web Pages" folder.

The right sidebar shows the "Palette" with various HTML and JSF components. The "HTML" section includes Table, Ordered List, Unordered List, Image, Link, and Meta data. The "HTML Forms" section includes Form, Text Input, Multi-line Input, Drop-down List, Checkbox, Radio Button, File Select, and Button. The "JSF" section includes Metadata, JSF Form, JSF Form From Entity, JSF Data Table, and JSF Data Table From Entity.

The bottom status bar shows the output window with the message: "DRDA\_SecurityInstalled.I Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania połączeń n".



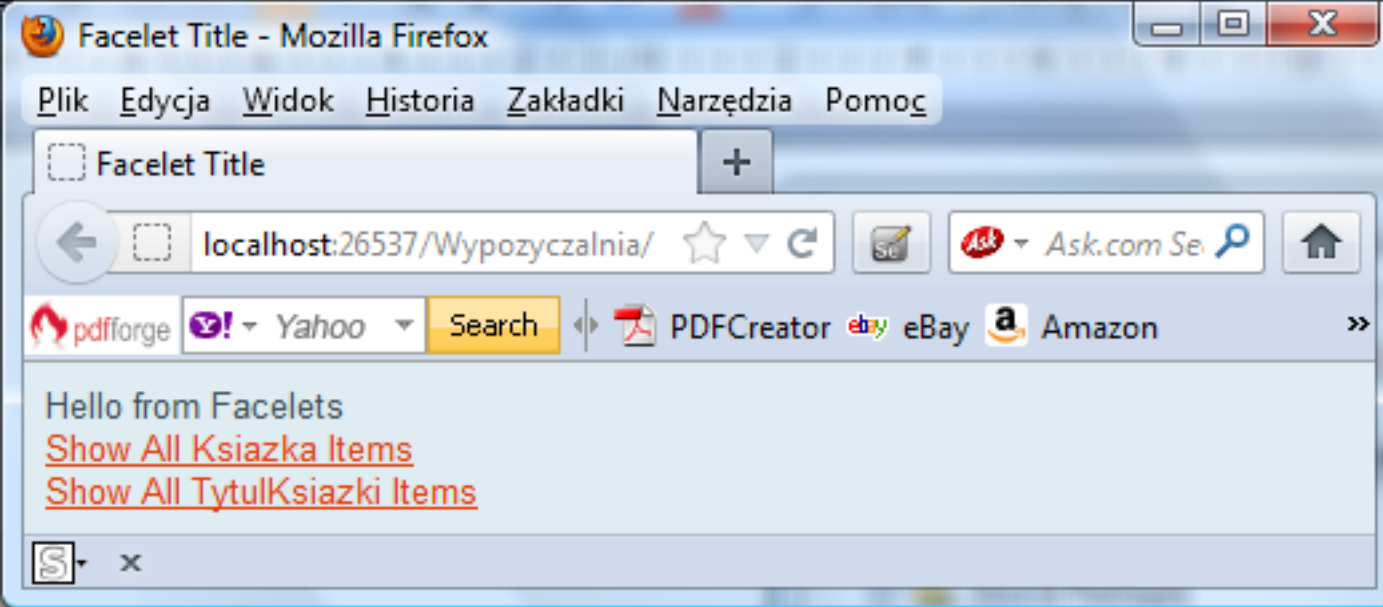
# Zawartość katalogu projektu aplikacji internetowej (2)



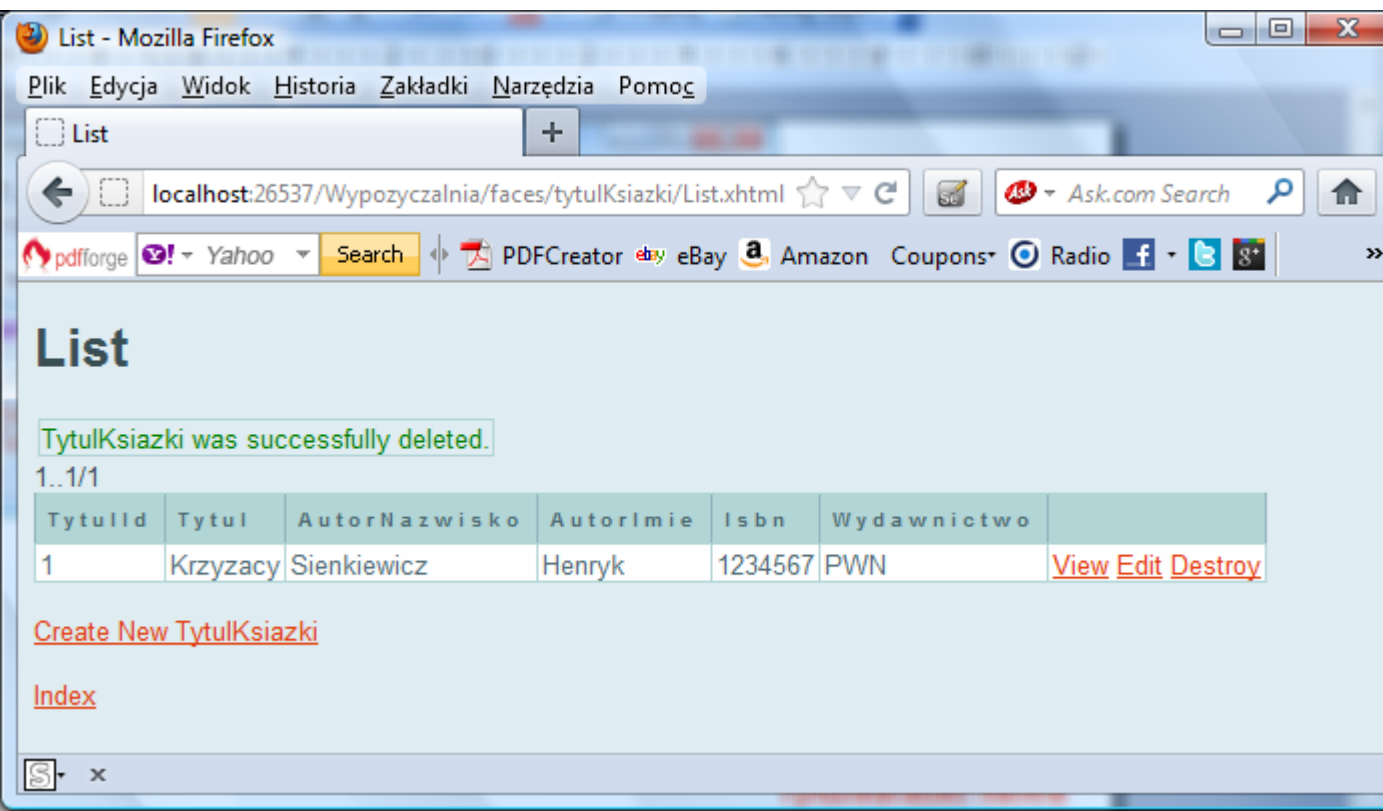
The screenshot displays the NetBeans IDE 7.2 interface. The left sidebar shows a project tree for 'Wypożyczalnia' with a structure including 'Web Pages', 'Source Packages', 'jpa', 'jpa.session', 'jsf', and 'Libraries'. The main editor window shows the code for 'TytulKsiazkiFacade.java'. The code includes package declarations, imports for javax.ejb.Stateless, javax.persistence.EntityManager, javax.persistence.PersistenceContext, and jpa.TytulKsiazki. It defines a @Stateless class TytulKsiazkiFacade extending AbstractFacade<TytulKsiazki>, with a constructor and a protected getEntityManager() method.

```
1  /*...*/
5  package jpa.session;
6
7  import javax.ejb.Stateless;
8  import javax.persistence.EntityManager;
9  import javax.persistence.PersistenceContext;
10 import jpa.TytulKsiazki;
11
12 /*...*/
16 @Stateless
17 public class TytulKsiazkiFacade extends AbstractFacade<TytulKsiazki> {
18     @PersistenceContext(unitName = "WypożyczalniaPU")
19     private EntityManager em;
20
21     @Override
22     protected EntityManager getEntityManager() {
23         return em;
24     }
25
26     public TytulKsiazkiFacade() {
27         super(TytulKsiazki.class);
28     }
29 }
30
```

The bottom output window shows the message: 'DRDA\_SecurityInstalled.I Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania'.

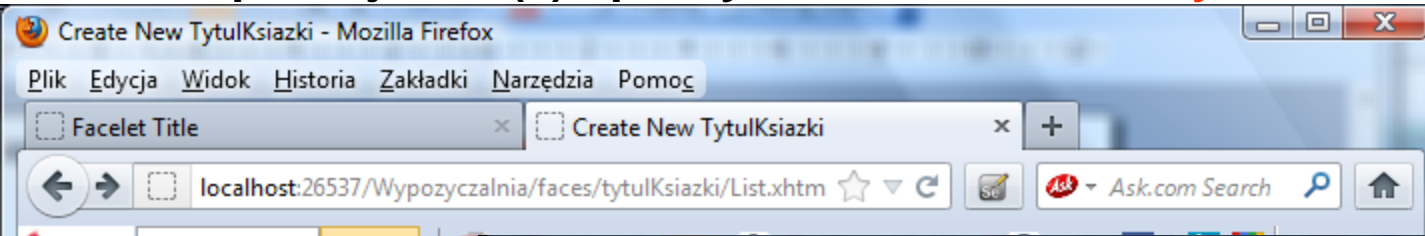


**Widok aplikacji JSF (1) - po uruchomieniu**

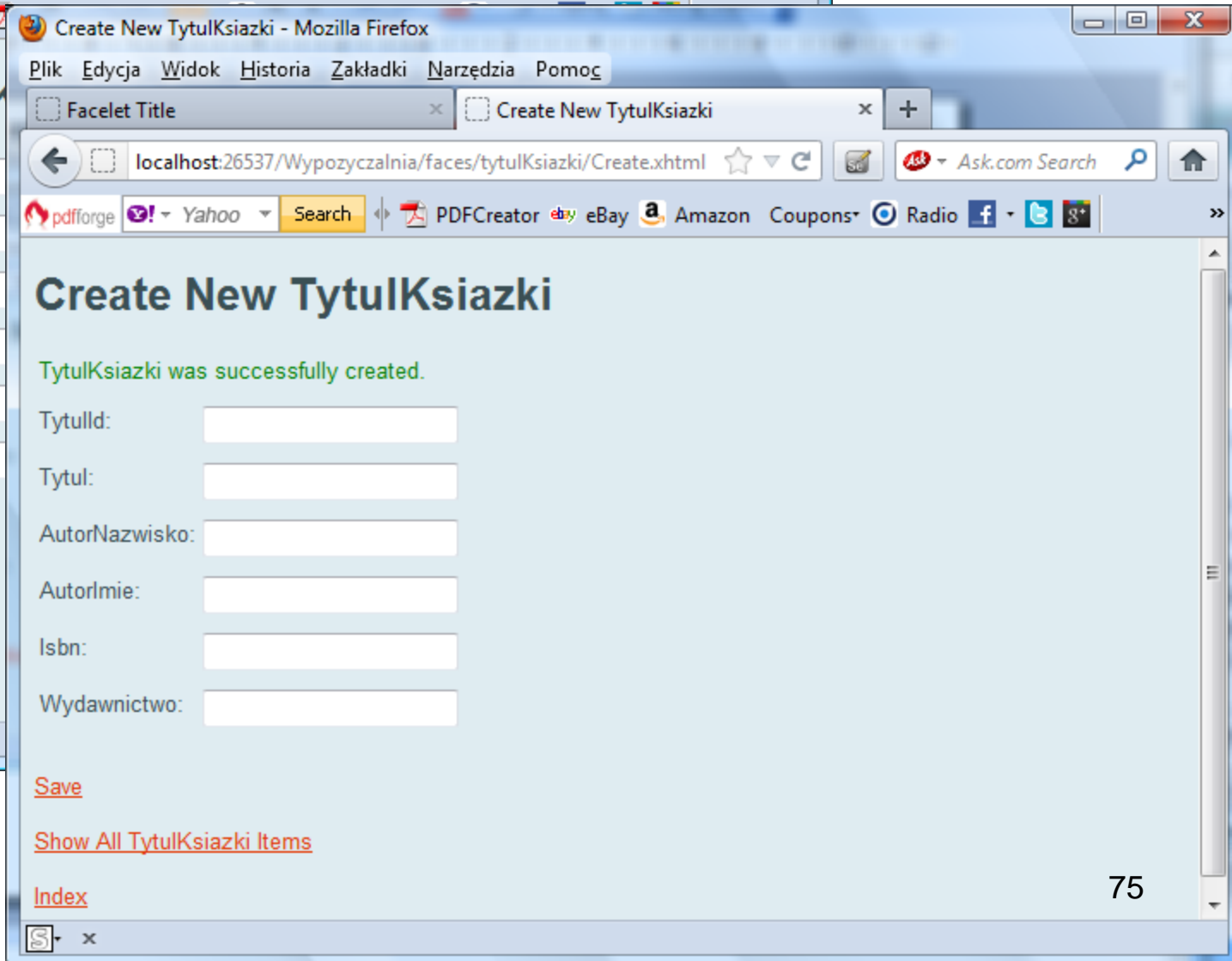
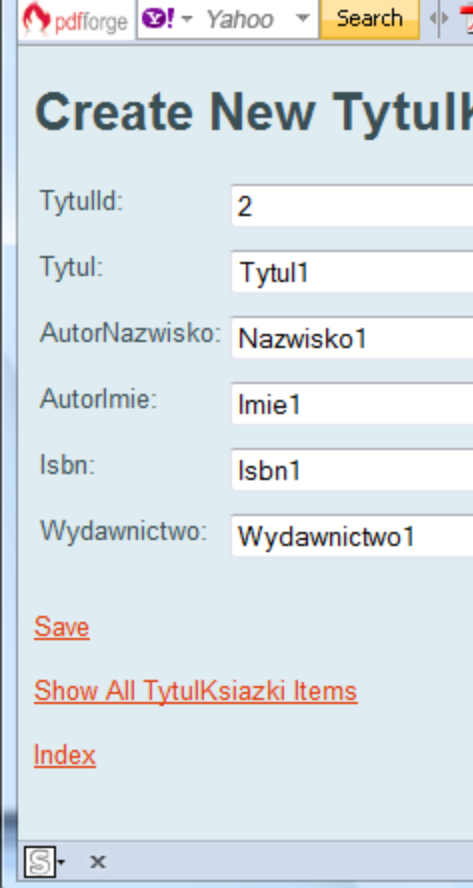


**Widok aplikacji JSF (2) - po wyborze Show All TytułKsiążki Items**

## Widok aplikacji JSF (3) - po wyborze **Create New TytułKsiazki**



Widok aplikacji JSF  
(4) - po wyborze  
**Save**



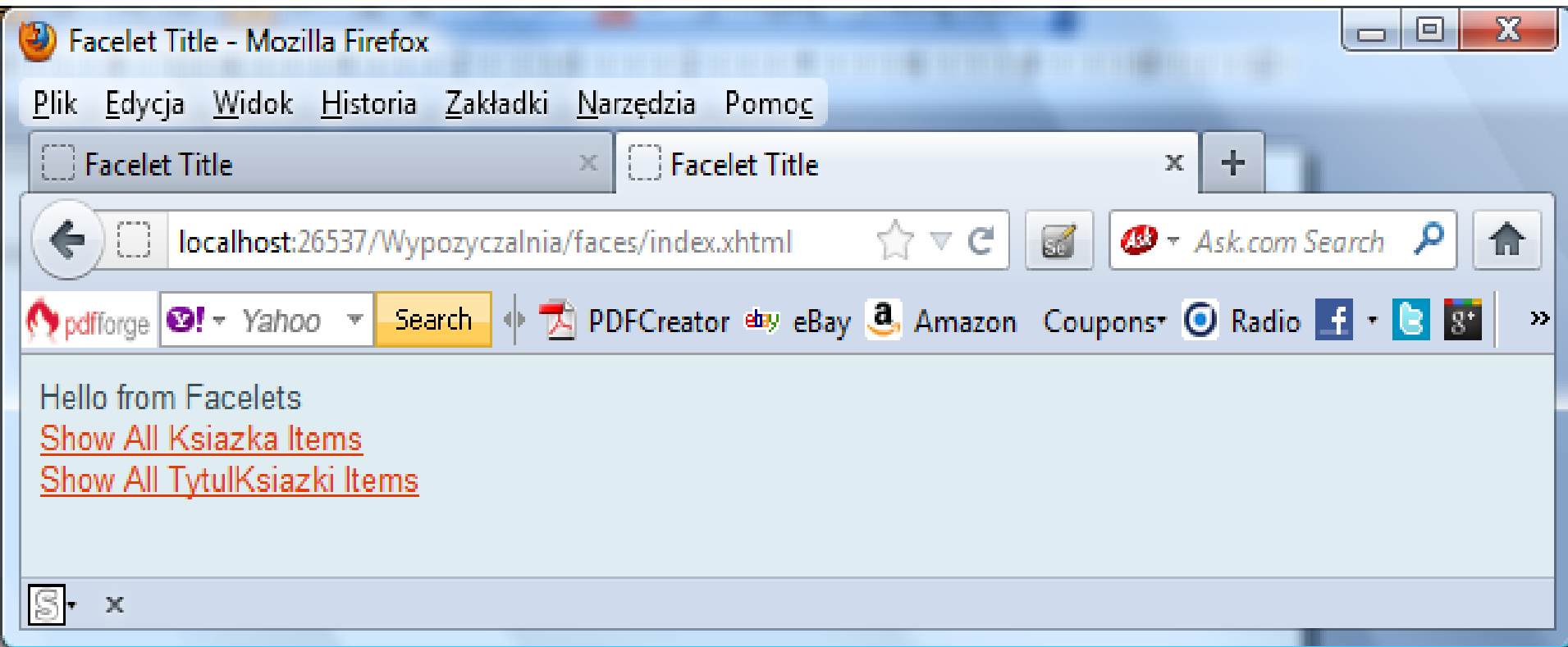
# Widok aplikacji JSF (5) - po wyborze **Show All TytulKsiazki Items**

The screenshot shows a Mozilla Firefox browser window with the title 'List - Mozilla Firefox'. The address bar contains the URL 'localhost:26537/Wypożyczalnia/faces/tytulKsiazki/Create.xhtml'. The page content includes a table with the following data:

TytuId	Tytuł	AutorNazwisko	AutorImie	Isbn	Wydawnictwo	
1	Krzyzacy	Sienkiewicz	Henryk	1234567	PWN	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
3	Tytuł1	Nazwisko1	Imie1	Isbn1	Wydawnictwo1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>

Below the table, there is a link 'Create New TytulKsiazki' and a link 'Index'.

## Widok aplikacji JSF (6) - po wyborze **Index**



# Kod HTML formularza index.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?> <!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head id="j_idt2">
    <title>Facelet Title</title>
    <link type="text/css" rel="stylesheet"
      href="/WebKatalogKsiazek1/faces/javax.faces.resource/css/jscrud.css;jsessionid=7017_986b1d5a46cef736ff282f69" />
  </head>
  <body>
    Hello from Facelets <br />
    <a href="/WebKatalogKsiazek1/faces/web_jsf/ksiazka/List.xhtml;jsessionid=7017986b1d5a46cef736ff282f69">
      Show All Ksiazka Items </a>
    <br />
    <a href="/WebKatalogKsiazek1/faces/web_jsf/tytulKsiazki/List.xhtml;jsessionid=7017986b1d5a46cef736ff282f69">
      Show All TytulKsiazki Items </a>
  </body>
</html>
```

# Widok aplikacji JSF (7) - po wyborze **Show All Ksiazka Items**

The screenshot shows a Mozilla Firefox browser window with the following details:

- Window title: List - Mozilla Firefox
- Address bar: localhost:26537/Wypożyczalnia/faces/ksiazka/List.xhtml
- Page title: List
- Page content:
  - 1..1/1

KsiazkaId	Numer	IdTytul	
1	1	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>

    - [Create New Ksiazka](#)
    - [Index](#)

## Widok aplikacji JSF (8) po wyborze **Create New Ksiazka**

Create New Ksiazka - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Create New Ksiazka

localhost:26537/Wypożyczalnia/faces/ksiazka/List.xhtml

# Create New Ksiazka

Ksiazkald:

Numer:

IdTytul:

- 
- jpa.TytulKsiazki[ tytulId=1 ]
- jpa.TytulKsiazki[ tytulId=3 ]

[Save](#)

[Show All Ksiazka Items](#)

[Index](#)

80



## Widok aplikacji JSF (9) - po wyborze **Create New Ksiazka**

The screenshot shows a Mozilla Firefox browser window with the following elements:

- Browser Title Bar:** "Create New Ksiazka - Mozilla Firefox"
- Menu Bar:** Plik, Edycja, Widok, Historia, Zakładki, Narzędzia, Pomoc
- Address Bar:** localhost:26537/Wypożyczalnia/faces/ksiazka/List.xhtml
- Search Bar:** Ask.com Search
- Toolbar:** pdfforge, Yahoo Search, PDFCreator, eBay, Amazon, Coupons, Radio, Facebook, Twitter, Google
- Page Content:**
  - # Create New Ksiazka
  - Form fields:
    - KsiazkaId:
    - Numer:
    - IdTytuł:
  - Buttons:
    - [Save](#)
    - [Show All Ksiazka Items](#)
    - [Index](#)
- Status Bar:** Shows a small icon and the number 81.

## Widok aplikacji JSF (10) - po wyborze **Save**

Create New Książka - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Create New Książka +

localhost:26537/Wypożyczalnia/faces/ksiazka/Create.xhtml

pdfforge Yahoo Search PDFCreator eBay Amazon Coupons Radio f t g+

# Create New Książka

Książka was successfully created.

Książkald:

Numer:

IdTytuł:

[Save](#)

[Show All Książka Items](#)

# Widok aplikacji JSF (11) - po wyborze **Show All Książka Items**

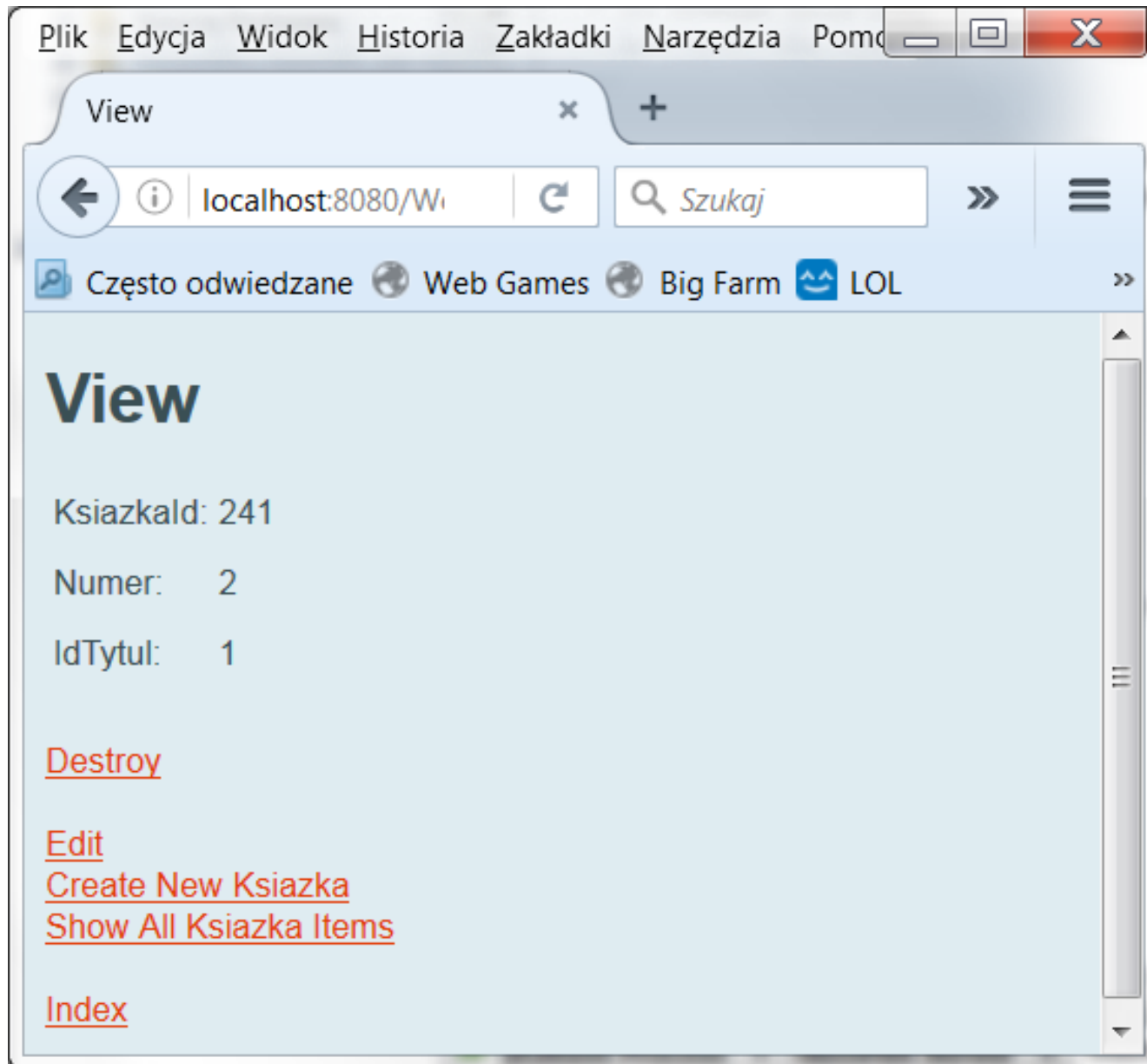
The screenshot shows a Mozilla Firefox browser window with the following details:

- Window title: List - Mozilla Firefox
- Address bar: localhost:26537/Wypożyczalnia/faces/ksiazka/Create.xhtml
- Page title: List
- Table content:

KsiazkaId	Numer	IdTytuł	
1	1	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
2	2	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>

Below the table, there are two links: [Create New Książka](#) and [Index](#).

# Formularz View.xhtml



# Kod HTML formularza View.xhtml (1)

```
<?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head id="j_idt2">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>View</title>
    <link type="text/css" rel="stylesheet" href="/WebKatalogKsiazek1/faces/javax.faces.resource/jsfcrud.css?ln=css" />
  </head>
  <body>
    <h1>View </h1>
    <p>
      <div id="messagePanel"></div>
      <form id="j_idt12" name="j_idt12" method="post"
        action="/WebKatalogKsiazek1/faces/web_jsf/ksiazka/View.xhtml"
        enctype="application/x-www-form-urlencoded">
        <input type="hidden" name="j_idt12" value="j_idt12" />
        <table id="j_idt12:ksiazka">
          <tbody>
            <tr>
              <td>KsiazkaId:</td>
              <td><span title="KsiazkaId">2</span></td>
            </tr>
            <tr>
              <td>Numer:</td>
              <td><span title="Numer">2</span></td>
            </tr>
            <tr>
              <td>IdTytul:</td>
              <td><span title="IdTytul">1</span></td>
            </tr>
          </tbody>
        </table>
        <br />
      </p>
    </div>
  </body>
</html>
```

## Kod HTML formularza View.xhtml (2)

```
<script type="text/javascript"
  src="/WebKatalogKsiazek1/faces/javax.faces.resource/jsf.js?ln=javax.faces&stage=Development">
</script>
<a href="#" onclick="mojarra.jsfcljs(document.getElementById('j_idt12'),
  {'j_idt12:j_idt20':'j_idt12:j_idt20'});return false"> Destroy</a>
<br /> <br />
<a href="/WebKatalogKsiazek1/faces/web_jsf/ksiazka/Edit.xhtml">Edit</a>
<br />
<a href="#" onclick="mojarra.jsfcljs(document.getElementById('j_idt12'),
  {'j_idt12:j_idt24':'j_idt12:j_idt24'});return false">Create New Ksiazka</a>
<br />
<a href="#" onclick="mojarra.jsfcljs(document.getElementById('j_idt12'),
  {'j_idt12:j_idt26':'j_idt12:j_idt26'});return false">Show All Ksiazka Items</a>
<br /> <br />
<a href="/WebKatalogKsiazek1/faces/index.xhtml">Index</a>
<input type="hidden" name="javax.faces.ViewState" id="j_id1:javax.faces.ViewState:0"
  value="-8909617770357827330:-8863361915686680743" autocomplete="off" />
</form>
</p>
</body>
</html>
```