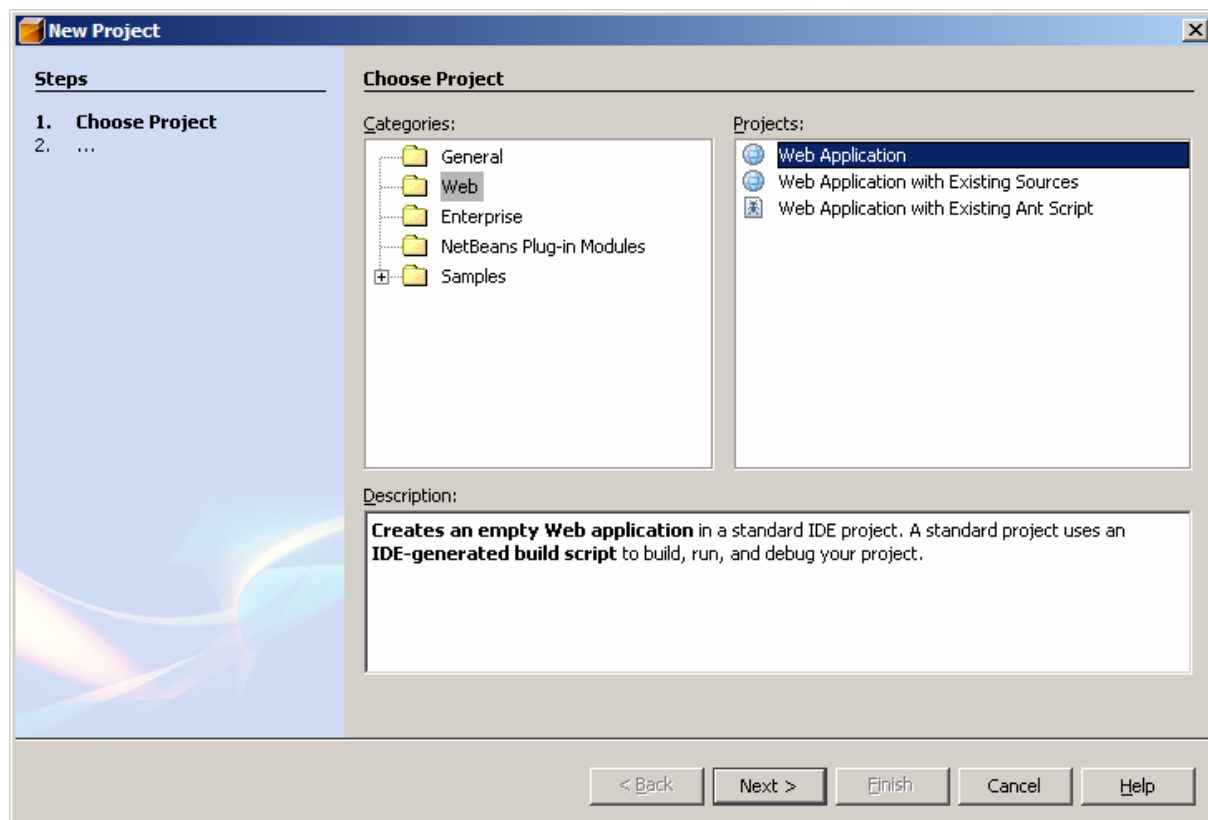


# Zaawansowane aplikacje internetowe - laboratorium

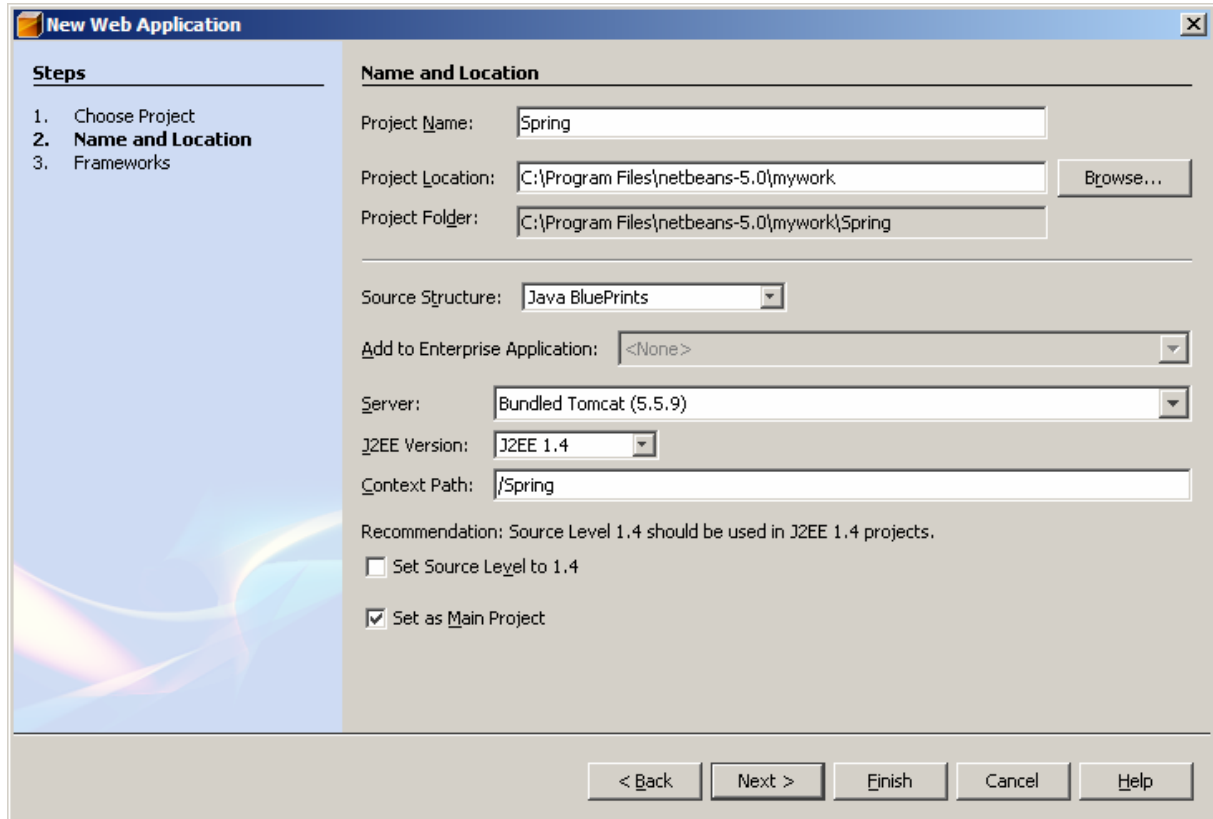
## Architektura Spring.

Celem ćwiczenia jest przygotowanie prostej aplikacji internetowej wykorzystującej architekturę Spring. Aplikacja składa się z jednego zarządcy (ang. dispatcher) i jednego kontrolera wyświetlającego stronę JSP. Dodatkowo, aplikacja wykorzystuje mechanizm fabryki komponentów do połączenia się z bazą danych i wyświetlenia listy pracowników. Do wykonania ćwiczenia potrzebne jest zintegrowane środowisko programistyczne NetBeans IDE 5.0 (do pobrania z <http://www.netbeans.org>), środowisko J2SE 1.4.2 (lub wyższe), oraz Spring Framework 1.2.8 (do pobrania z <http://www.springframework.org/download>).

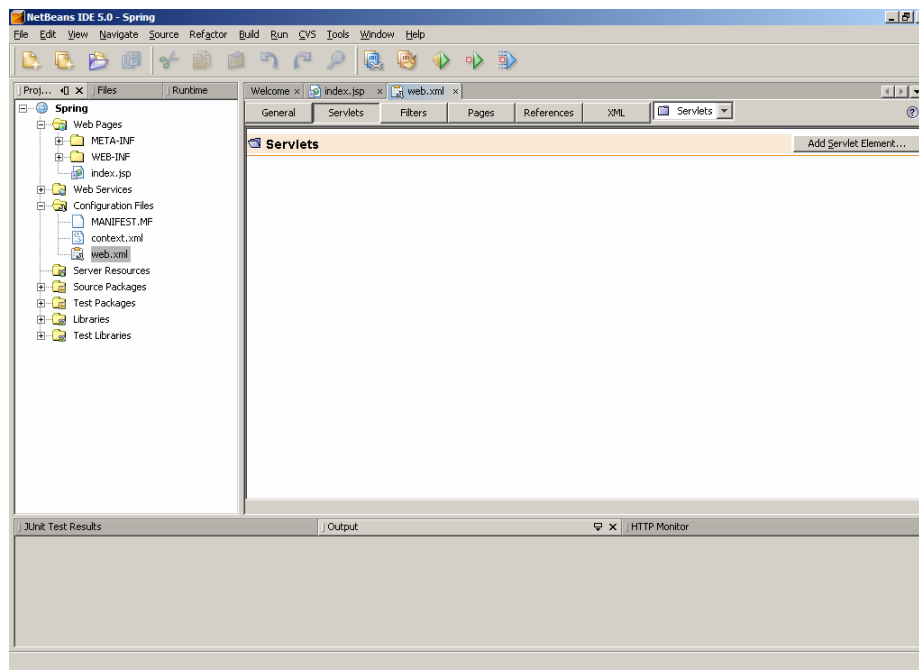
1. Uruchom narzędzie NetBeans IDE 5.0
2. Z menu głównego wybierz File→New Project. Wybierz kategorię Web i typ projektu Web Application. Kliknij przycisk **Next >**.



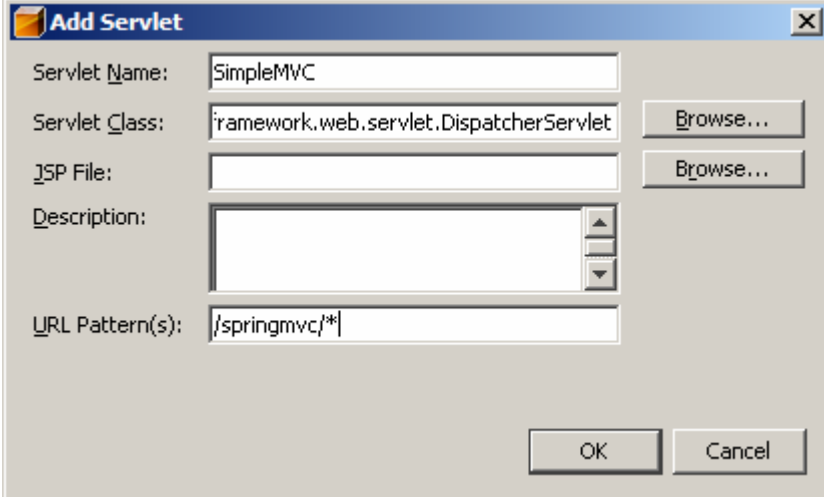
3. Podaj nazwę projektu, np. „Spring”. Upewnij się, że opcja Set Source Level to 1.4 jest odznaczona. Kliknij przycisk **Finish**.



4. W nawigatoryze projektu (okno po lewej stronie) rozwiń drzewo Configuration Files i kliknij dwukrotnie na pliku web.xml. W oknie edytora przejdź na zakładkę Servlets (pasek z przyciskami u góry okna edycji).



- Kliknij przycisk **Add Servlet Element...** . Wprowadź nazwę serwletu zarządzcy, "SimpleMVC". Jako zarządzcę wykorzystamy serwlet dostarczony przez architekturę Spring. Jako klasę serwletu zarządzcy wprowadź następującą klasę: `org.springframework.web.servlet.DispatcherServlet`. Jako wzorzec URL żądań kierowanych do serwletu zarządzcy wprowadź: `/springmvc/*`. Kliknij przycisk **OK**.



**Add Servlet**

Servlet Name: SimpleMVC

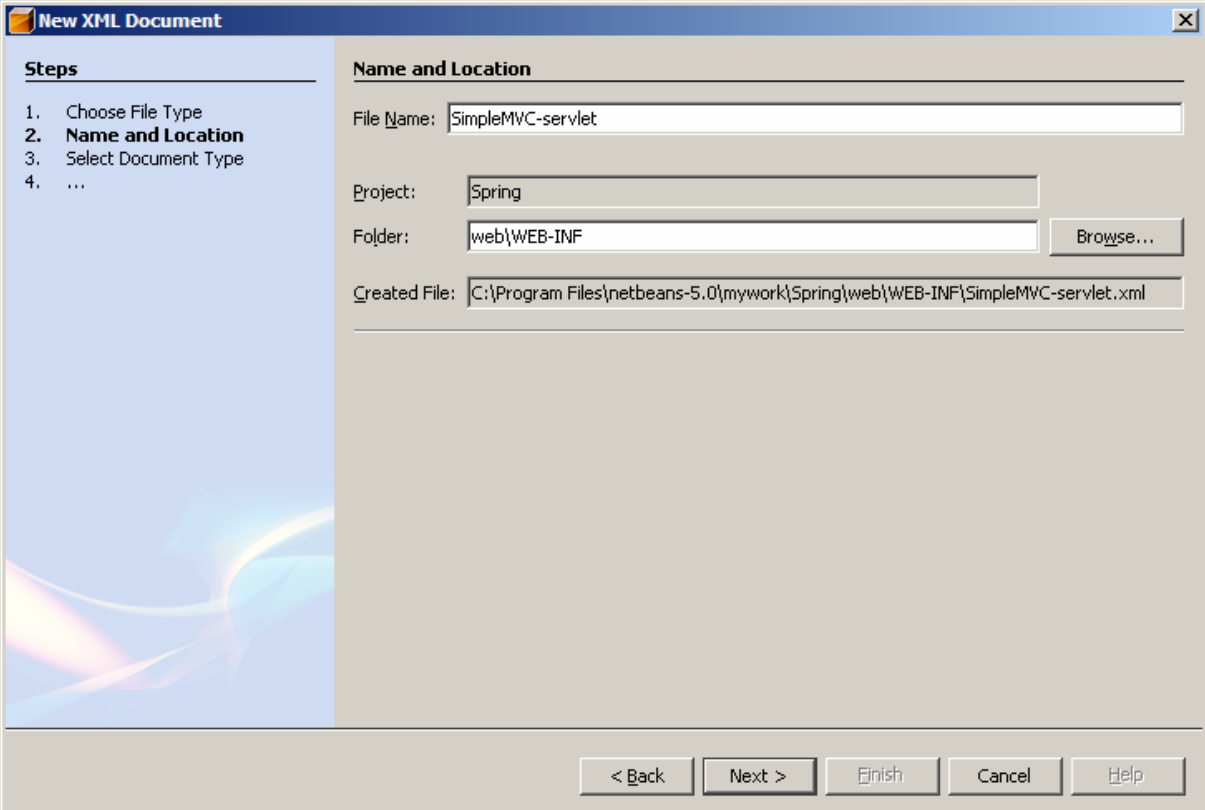
Servlet Class: ramework.web.servlet.DispatcherServlet

JSP File:

Description:

URL Pattern(s): /springmvc/\*

- Plikiem konfiguracyjnym i kontekstowym dla naszego serwletu jest plik o nazwie `/WEB-INF/[servletname]-servlet.xml`. Kliknij prawym przyciskiem myszy na katalogu `WEB-INF` i z menu wybierz **New**→**XML Document**. Wprowadź nazwę pliku: `SimpleMVC-servlet` (nie dodawaj rozszerzenia, zostanie ono dodane automatycznie!). Kliknij przycisk **Next >**.



**New XML Document**

**Steps**

- Choose File Type
- Name and Location**
- Select Document Type
- ...

**Name and Location**

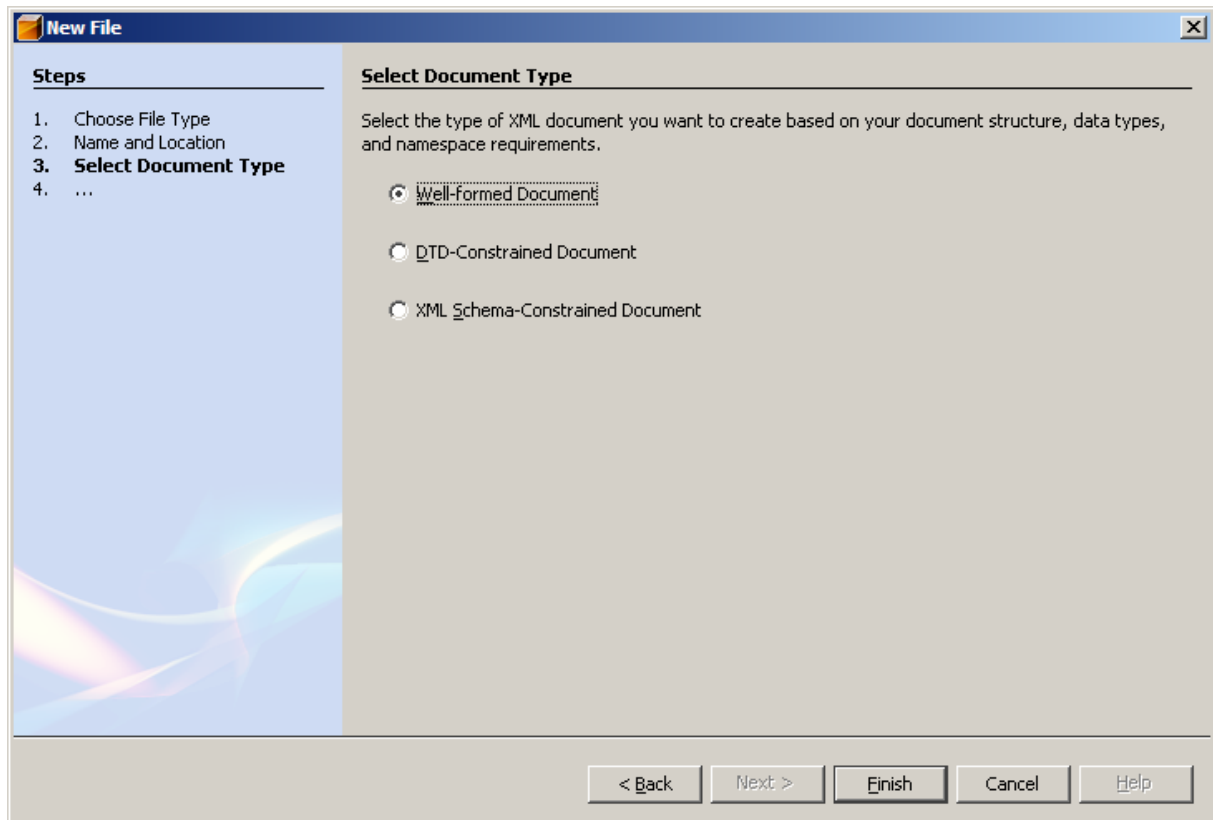
File Name: SimpleMVC-servlet

Project: Spring

Folder: web\WEB-INF

Created File: C:\Program Files\netbeans-5.0\mywork\Spring\web\WEB-INF\SimpleMVC-servlet.xml

7. Wybierz opcję Well-formed Document i kliknij przycisk Finish.



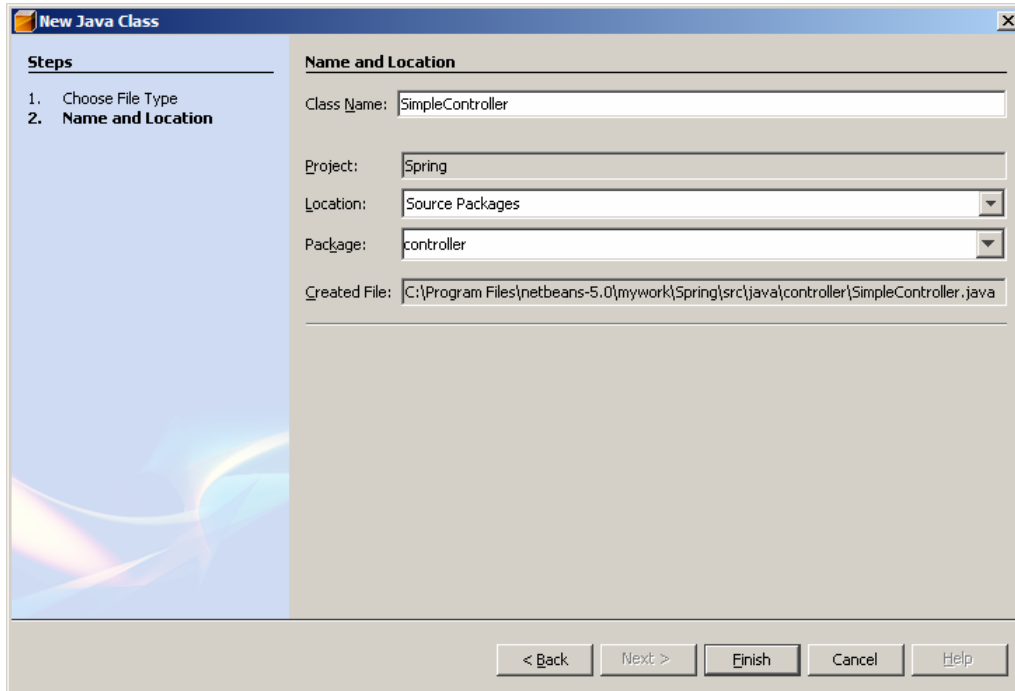
8. Wprowadź do pliku SimpleMVC-servlet.xml następujący kod.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
    "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>
  <bean name="/index.html" class="controller.SimpleController"/>

  <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix"><value>/WEB-INF/jsp/</value></property>
    <property name="suffix"><value>.jsp</value></property>
    <property name="viewClass">
      <value>org.springframework.web.servlet.view.JstlView</value>
    </property>
  </bean>
</beans>
```

9. Powyższa konfiguracja zleca kierowanie żądań o URL `/simplemvc/index.html` do kontrolera o nazwie `controller.SimpleController`. Kolejnym krokiem jest implementacja tego kontrolera. Kliknij prawym przyciskiem myszy na ikonie `Source Packages` i z menu kontekstowego wybierz `New→Java Class...` Wprowadź nazwę klasy: `SimpleController` i nazwę pakietu: `controller`. Kliknij przycisk `Finish`.



10. Plik `SimpleController.java` zostanie załadowany do edytora. Wprowadź do pliku następujący kod. Kontroler otrzymuje żądanie, tworzy nowy obiekt wynikowy `ModelAndView`, dodaje do modelu obiekt biznesowy (w tym przypadku prosty obiekt `java.util.Date`) i zwraca obiekt wynikowy. Nazwa obiektu wynikowego „welcome” jest tłumaczona na `/WEB-INF/jsp/welcome.jsp`.

```
package controller;

import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;
import javax.servlet.http.*;
import java.util.*;

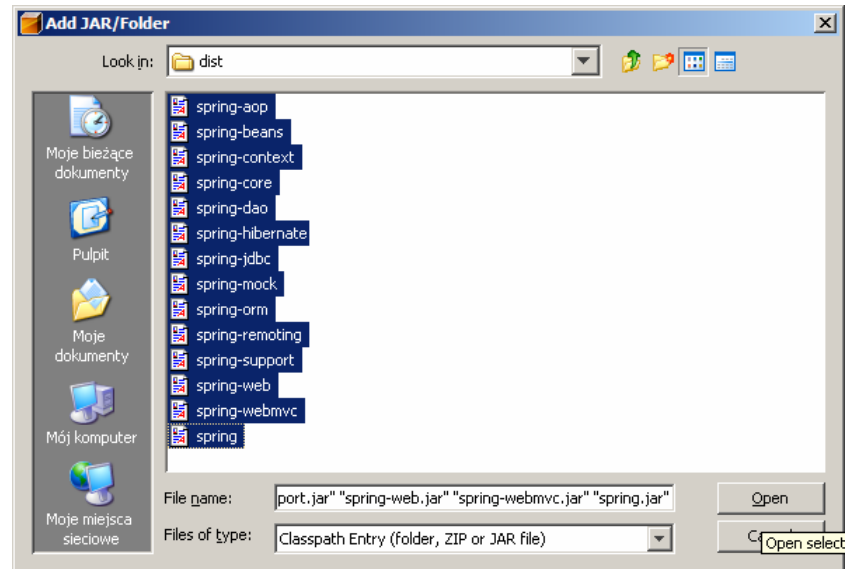
public class SimpleController extends AbstractController {

    public SimpleController() {
    }

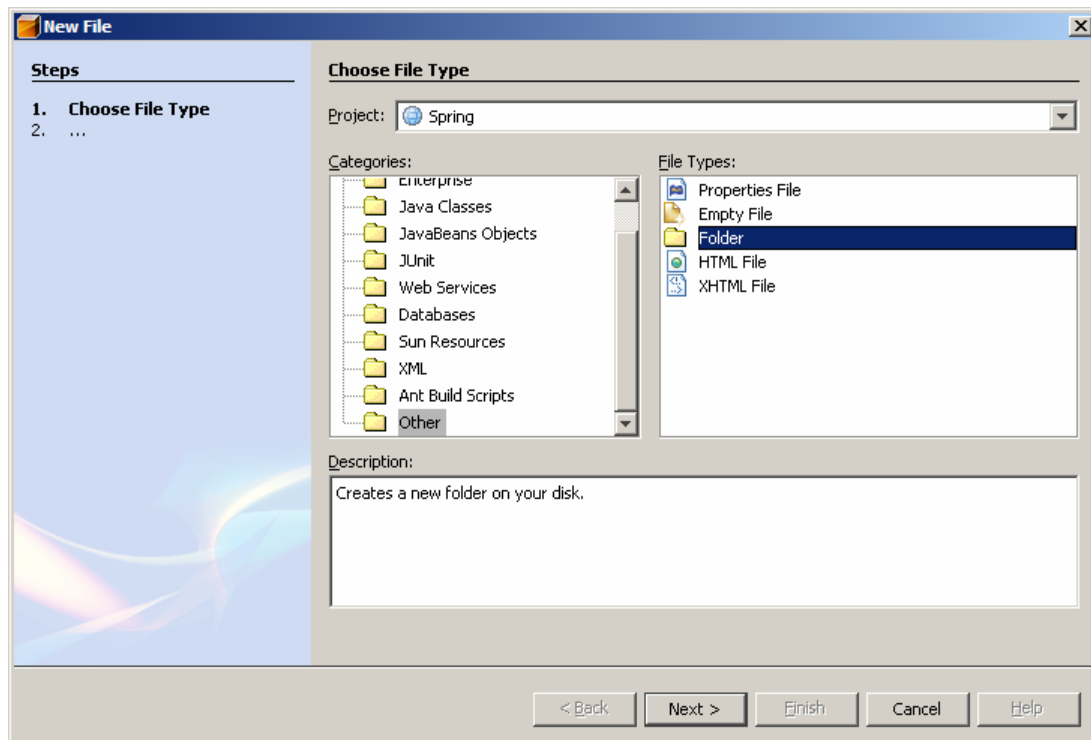
    public ModelAndView handleRequestInternal
        (HttpServletRequest request, HttpServletResponse response)
        throws Exception {

        ModelAndView mav = new ModelAndView("welcome");
        mav.addObject("date", new Date());
        return mav;
    }
}
```

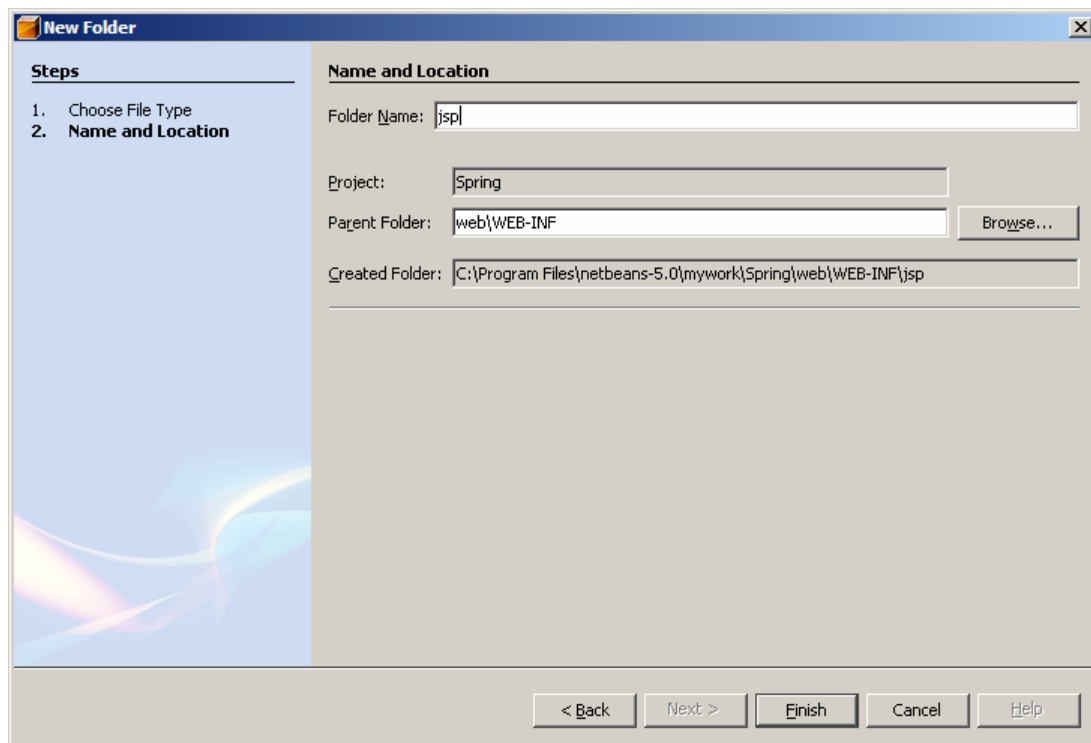
11. Błędy kompilacji są związane z tym, że do projektu nie dołączyliśmy jeszcze bibliotek Spring Framework. Kliknij prawym przyciskiem myszy na ikonie Libraries (u dołu w oknie nawigatora projektu). Z menu kontekstowego wybierz opcję Add JAR/Folder... Przejdź do katalogu lokalnego w którym zainstalowała(e)ś Spring Framework. Przejdź do podkatalogu dist i dodaj wszystkie znajdujące się tam biblioteki \*.jar. Kliknij przycisk **Open**.



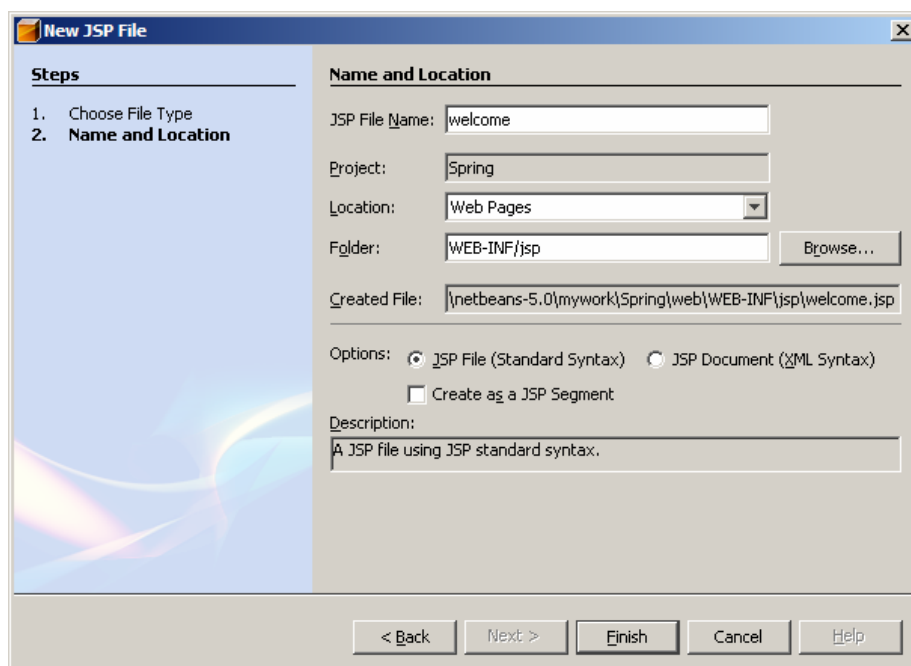
12. Kolejny krok to zbudowanie strony JSP wywoływanej przez kontroler. Kliknij prawym przyciskiem myszy na ikonie katalogu WEB-INF w nawigatorze projektu i z menu kontekstowego wybierz New→File/Folder... Przejdź do kategorii Other i zaznacz jako typ pliku Folder. Kliknij przycisk **Next >**.



13. Podaj nazwę folderu: `jsp` i kliknij przycisk **Finish**.



14. Kliknij prawym przyciskiem myszy na nowo utworzonym folderze `jsp` i z menu kontekstowego wybierz **New**→**JSP...**. Podaj nazwę dokumentu: `welcome` (nie podawaj rozszerzenia, zostanie dodane automatycznie!). Kliknij przycisk **Finish**.



15. Plik `welcome.jsp` zostanie otwarty w edytorze. Wprowadź następujący kod.

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

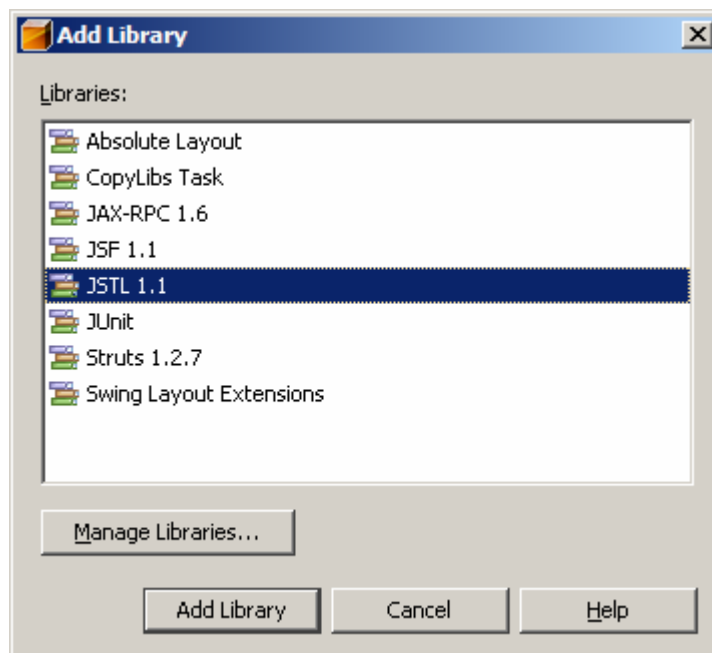
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Spring MVC</title>
  </head>
  <body>

    <h1>Congratulations!</h1>
    Now it is <c:out value="${date}"/>

  </body>
</html>
```

16. Powyższa strona JSP korzysta z biblioteki znaczników JSTL, należy zatem dodać odpowiednią bibliotekę do projektu. Kliknij prawym przyciskiem myszy na ikonie Libraries i z menu kontekstowego wybierz Add Library... Zaznacz bibliotekę JSTL 1.1 i kliknij przycisk **Add Library**.





17. Domyślnym plikiem projektu jest `index.jsp`. Zmodyfikuj zawartość tego pliku w taki sposób, aby odwołanie do `index.jsp` było natychmiast przekierowane do przygotowanego przez nas kontrolera. Kliknij dwukrotnie na pliku `index.jsp` i wprowadź następujący kod.

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

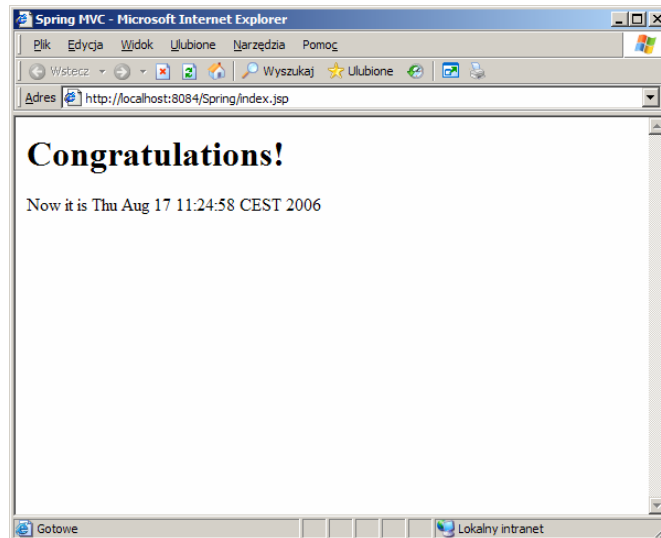
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Simple MVC with Spring</title>
  </head>
  <body>

    <jsp:forward page="/simplemvc/index.html"/>

  </body>
</html>
```

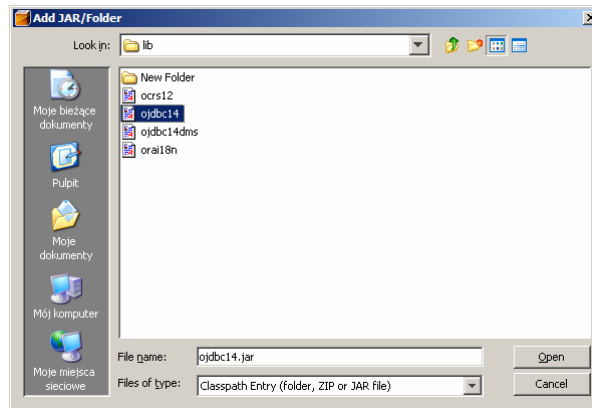
18. Uruchom aplikację za pomocą kombinacji klawiszy Shift+F6.



19. Kolejny krok to dodanie do projektu komponentu odpowiedzialnego za nawiązanie połączenia z bazą danych. Kliknij dwukrotnie na pliku `SimpleMVC-servlet.xml` i wewnątrz znaczników `<beans></beans>` wprowadź następującą definicję komponentu `dbBean`, oczywiście podając prawidłowe parametry połączenia (*host*, *port*, *sid*, *username*, *password*).

```
<bean id="dbBean"
  class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName"
    value="oracle.jdbc.driver.OracleDriver"/>
  <property name="url"
    value="jdbc:oracle:thin:@host:port:sid"/>
  <property name="username" value="username"/>
  <property name="password" value="password"/>
</bean>
```

20. Nawiązanie połączenia z bazą danych wymaga dołączenia do projektu właściwego sterownika JDBC. Kliknij prawym przyciskiem myszy na ikonie Libraries i z menu kontekstowego wybierz Add JAR/Folder... Przejdź do lokalnego katalogu zawierającego sterownik JDBC i zaznacz plik ze sterownikiem. Kliknij przycisk **Open**.



21. Nawiązanie połączenia, wykonanie zapytania i odczytanie wyników zapytania jest realizowane w kontrolerze. Kliknij dwukrotnie plik SimpleController.java. Dodaj kod obsługi połączenia z bazą danych. Ostateczna postać kontrolera powinna być następująca.

```
package controller;

import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.AbstractController;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.datasource.DriverManagerDataSource;
import javax.servlet.http.*;
import java.util.*;

public class SimpleController extends AbstractController {

    public SimpleController() {
    }

    public ModelAndView handleRequestInternal(HttpServletRequest request,
        HttpServletResponse response) throws Exception {

        ApplicationContext ctx = this.getApplicationContext();
        DriverManagerDataSource dbBean =
            (DriverManagerDataSource)ctx.getBean("dbBean");
        JdbcTemplate template = new JdbcTemplate(dbBean);

        List employees = template.queryForList("select * from emp");
        List empMap = new ArrayList();
        for (Object o : employees)
            empMap.add((Map)o);

        ModelAndView mav = new ModelAndView("welcome");
        mav.addObject("date", new Date());
        mav.addObject("employees", empMap);
        return mav;
    }
}
```

22. Zmodyfikuj wynikową stronę JSP w taki sposób, aby wyświetlała wynik zapytania przesłany przez kontroler w obiekcie „employees”. Kliknij dwukrotnie na pliku welcome.jsp. Ostateczna postać pliku powinna być następująca.

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Spring MVC</title>
</head>
<body>

<h1>Congratulations!</h1>
Now it is <c:out value="\${date}"/>

<fmt:setLocale value="en_US"/>

<h1>Employee list</h1>
<table border="1">
<tr><th>Name</th><th>Job</th><th>Salary</th></tr>

<c:forEach var="emp" items="\${employees}">
<tr>
<td><c:out value="\${emp["ENAME"]}"/></td>
<td><c:out value="\${emp["JOB"]}"/></td>
<td><fmt:formatNumber type="currency" value="\${emp["SAL"]}"/></td>
</tr>
</c:forEach>
</table>

</body>
</html>
```

23. Wróć do pliku `index.jsp` i uruchom aplikację za pomocą kombinacji klawiszy `Shift+F6`.

