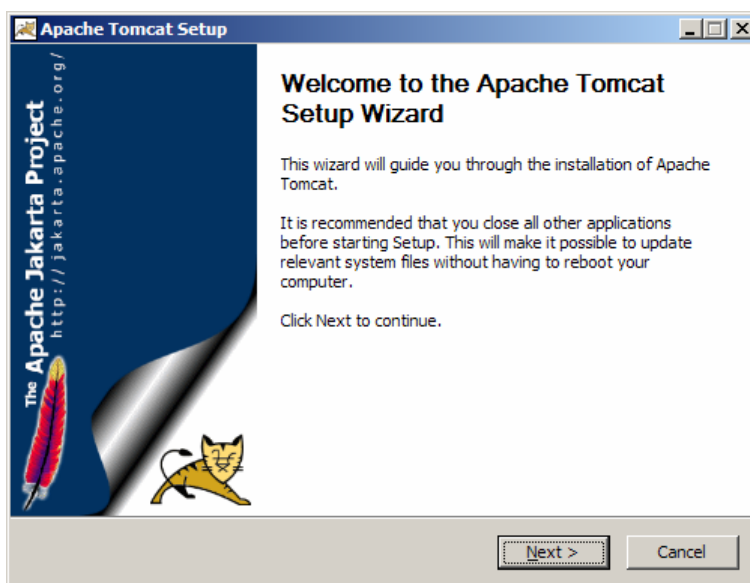


# Aplikacje internetowe - laboratorium

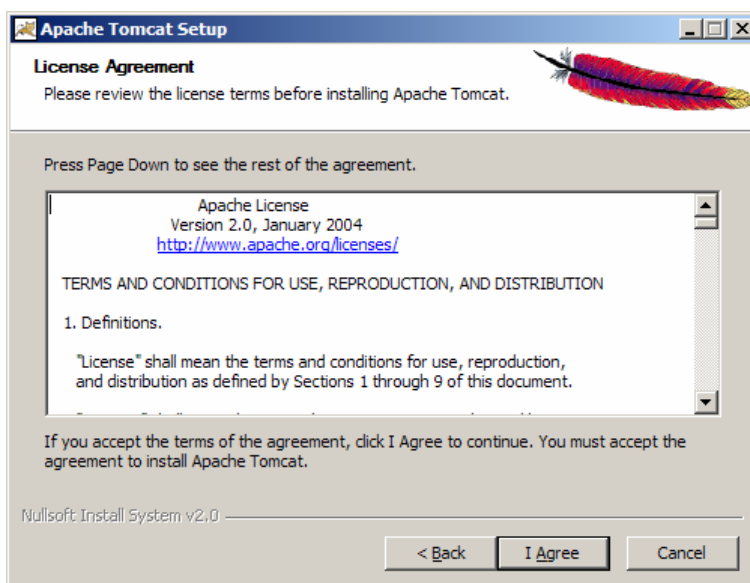
## Administracja serwerem aplikacji.

Celem ćwiczenia jest zainstalowanie i administracja prostym serwerem aplikacji. Ćwiczenie zostanie wykonane przy użyciu popularnego darmowego serwera aplikacji Apache Tomcat w wersji 5.5.17 (do pobrania ze strony <http://tomcat.apache.org>) oraz (opcjonalnie) sterownika JDBC do bazy danych Oracle.

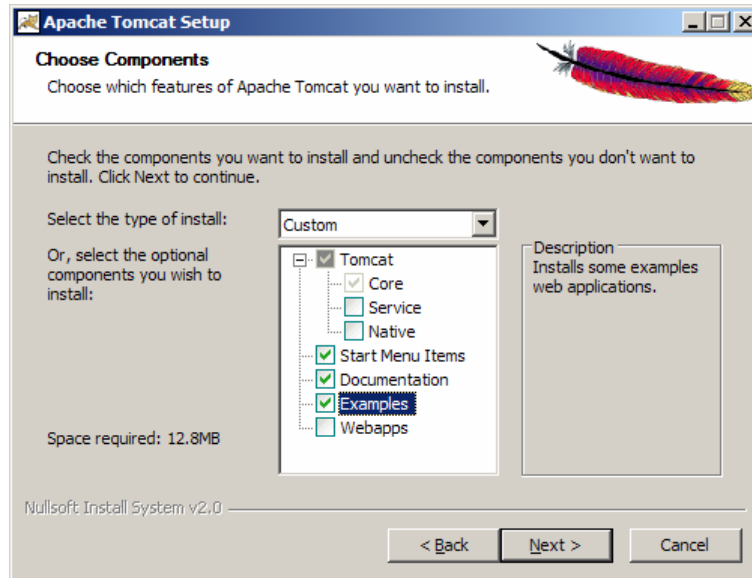
1. Ściągnij plik `apache-tomcat-5.5.17.exe` ze strony <http://tomcat.apache.org>
2. Uruchom plik `apache-tomcat-5.5.17.exe` w celu rozpoczęcia instalacji serwera aplikacji. Kliknij przycisk **Next >**.



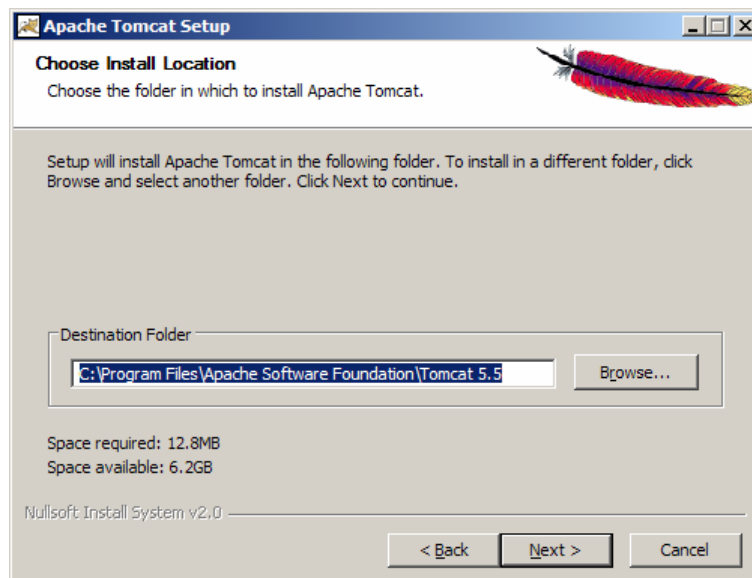
3. Zapoznaj się z licencją Apache Licence na której rozpowszechniane jest oprogramowanie Apache Tomcat. Kliknij przycisk **I Agree**.



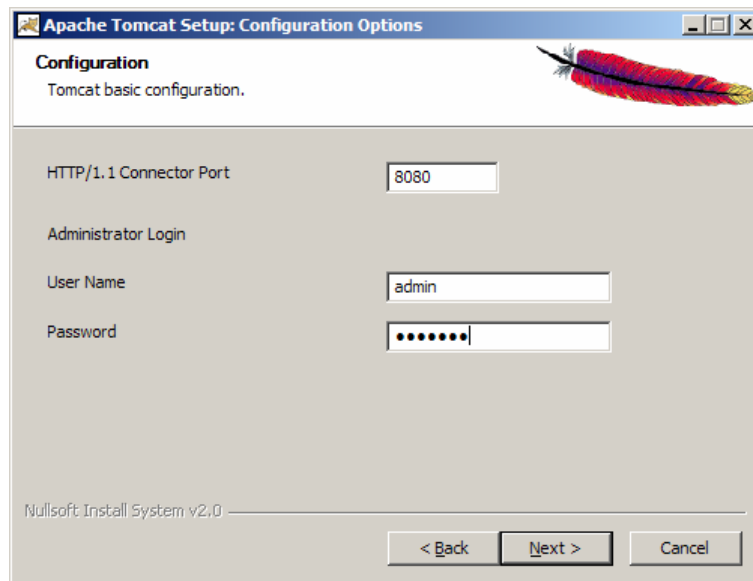
4. Wybierz rodzaj instalacji **Normal**. Rozwiń gałąź **Tomcat**. Masz możliwość zainstalować opcję **Service** która powoduje automatyczne uruchamianie serwera aplikacji w momencie uruchomienia systemu operacyjnego oraz opcję **Native** służącą do poprawy wydajności działania serwera aplikacji w środowisku produkcyjnym. Zaznacz chęć doinstalowania przykładów aplikacji internetowych. Kliknij przycisk **Next >**.



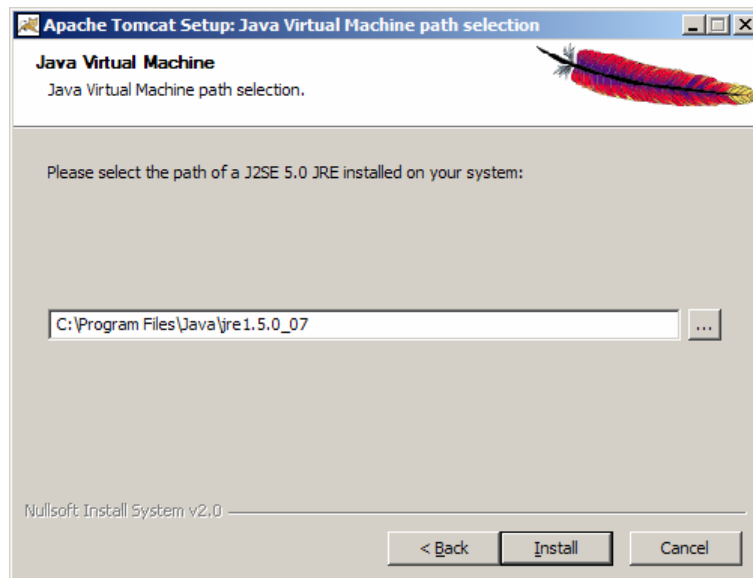
5. Wskaż katalog instalacyjny dla serwera aplikacji (możesz pozostawić domyślną lokalizację). Kliknij przycisk **Next >**.



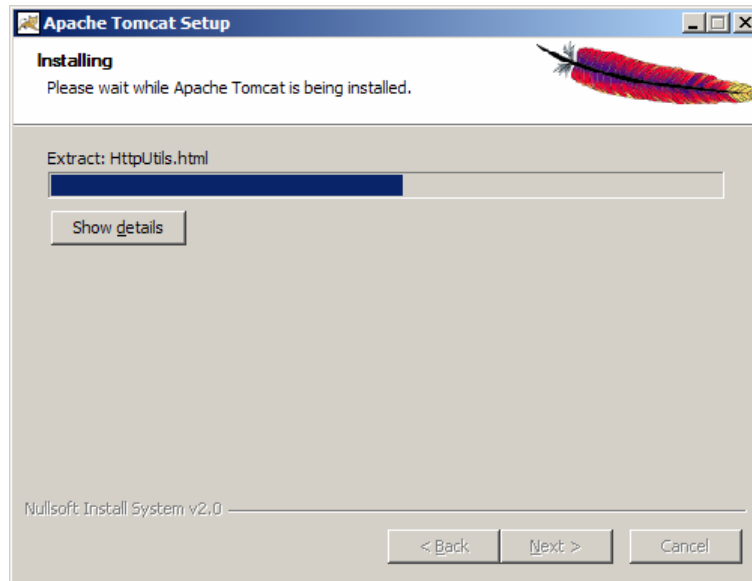
6. Wskaż port TCP/IP na którym ma być uruchomiony serwer aplikacji oraz nazwę i hasło użytkownika, który będzie miał prawo administrowania serwerem aplikacji. Kliknij przycisk **Next >**.



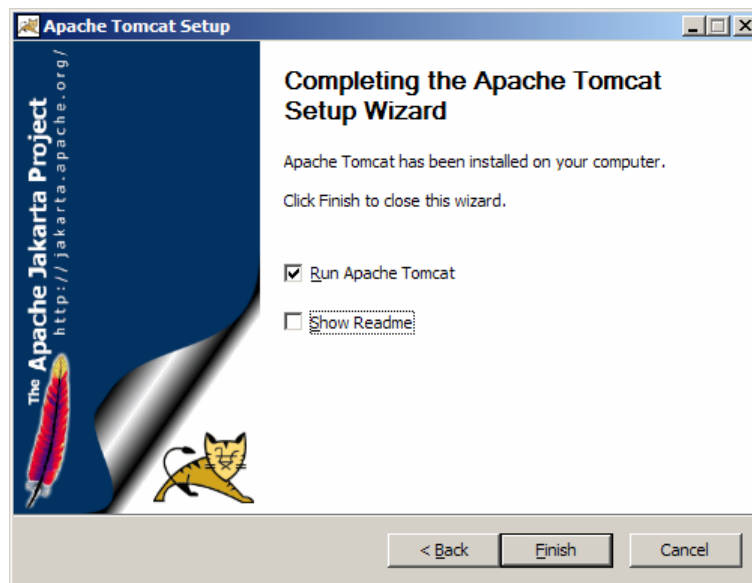
7. Wskaż lokalizację środowiska Java Runtime Environment (koniecznie w wersji 5.0). Kliknij przycisk **Install**.



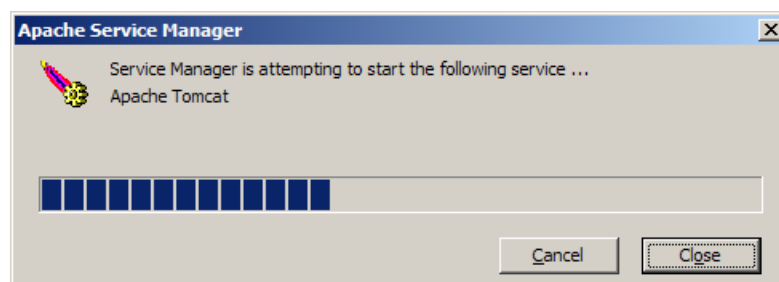
8. Poczekaj na zakończenie procesu instalacji.



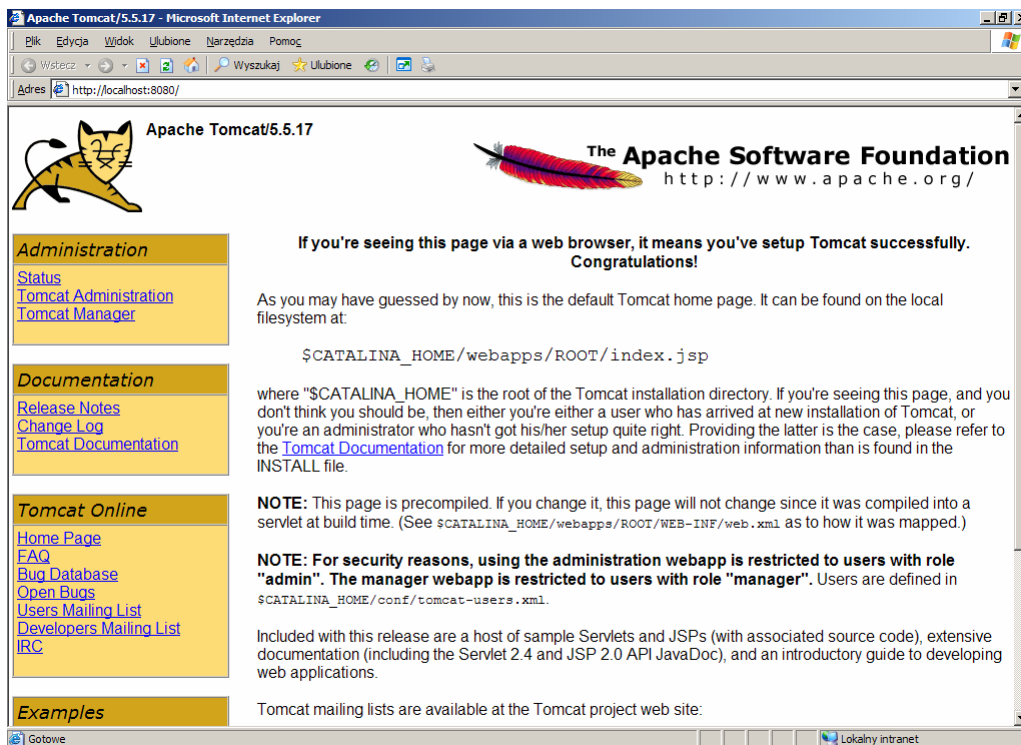
9. Upewnij się, że opcja Run Apache Tomcat jest zaznaczona. Kliknij przycisk **Finish**.



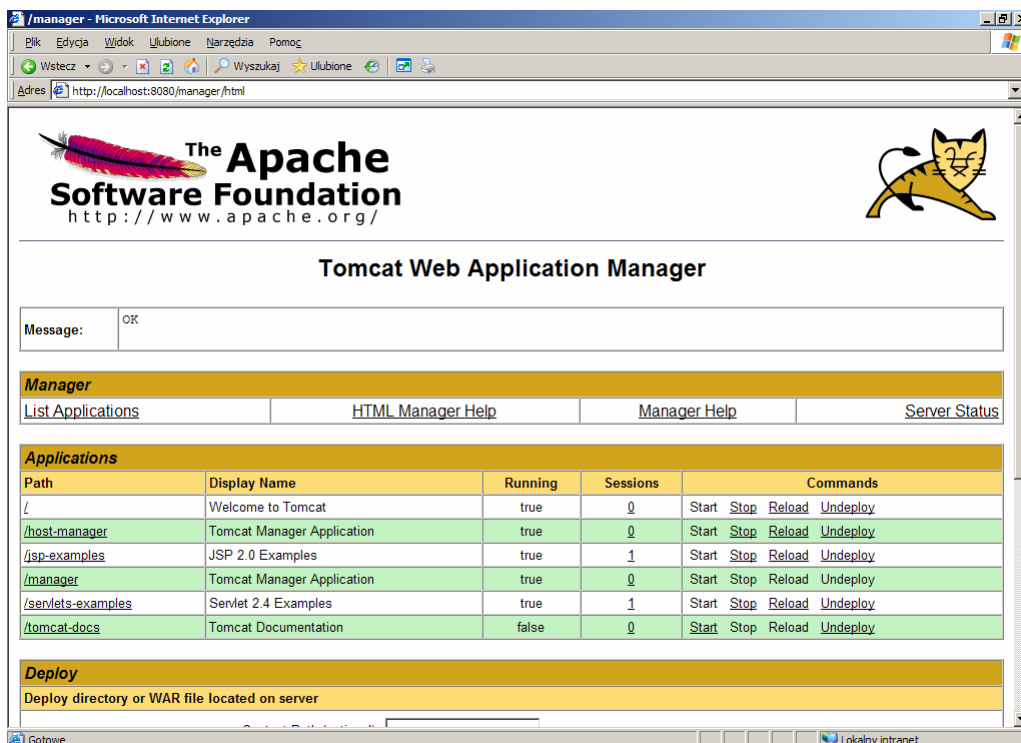
10. Poczekaj na uruchomienie usługi. Upewnij się, że w pasku systemowym jest obecna ikona Apache Tomcat.



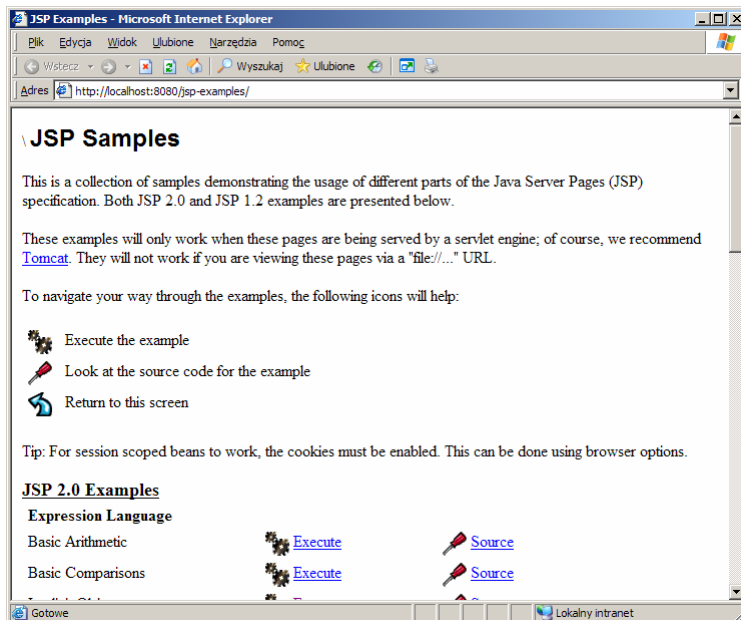
11. Uruchom przeglądarkę internetową i połącz się z adresem <http://localhost:8080>



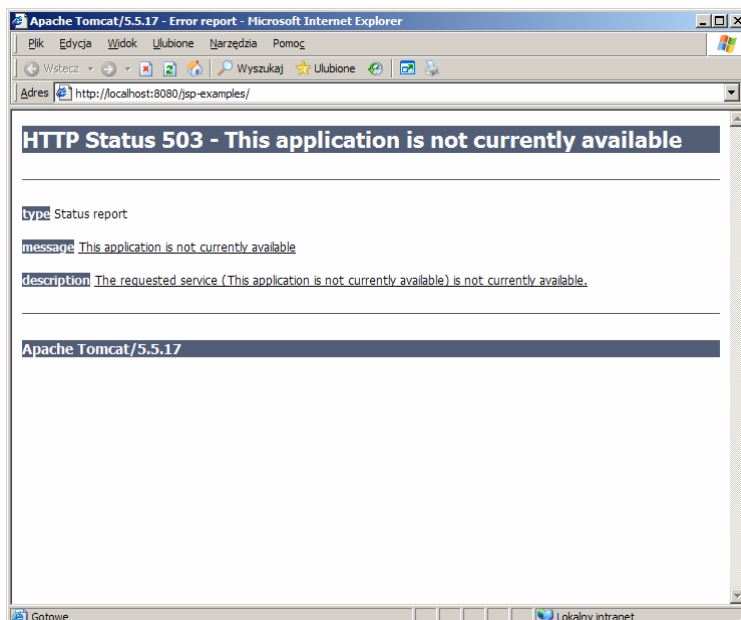
12. Kliknij na odnośnik [Tomcat Manager](#) w lewej górnej części okna. Wprowadź nazwę i hasło użytkownika, którego stworzyła(e)s w trakcie instalacji serwera aplikacji.



13. Okno przedstawia listę wszystkich aplikacji, które są aktualnie zainstalowane w serwerze aplikacji. Kliknij prawym przyciskiem myszy na odnośniku [/jsp-examples](#) i wybierz opcję **Otwórz w nowym oknie**. Możesz poświęcić chwilę na obejrzenie prostych przykładów zastosowania języka JSP EL, plików znaczników, dokumentów JSP XML, i innych.



14. Wróć do okna zarządcy serwera aplikacji i wybierz komendę **Stop** odnoszącą się do aplikacji [/jsp-examples](#). W okienku pop-up naciśnij przycisk **OK**. Wróć do okna z przykładami JSP i odśwież zawartość okna przeglądarki.



15. Zainstalujmy teraz przykładową aplikację internetową przygotowaną w postaci gotowego archiwum WAR (Web Archive). Przejdź do okna zarządcy serwera aplikacji. Sekcja Deploy pozwala na zainstalowanie aplikacji spakowanej w postaci pliku \*.war umieszczonej po stronie serwera lub umieszczonej na komputerze lokalnym. Znajdź pole **Select WAR file to upload** i kliknij przycisk **Przeglądaj...** Przejdź następnie do katalogu %CATALINA\_HOME%\webapps\tomcat-docs\appdev\sample (zmienna %CATALINA\_HOME% wskazuje na katalog instalacji serwera Tomcat) i zaznacz plik sample.war. Kliknij przycisk **Otwórz**. Kliknij przycisk **Deploy**.

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/jsp-examples	JSP 2.0 Examples	false	0	Start Stop Reload Undeploy
/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
<u>/sample</u>	Hello, World Application	true	0	Start Stop Reload Undeploy
/servlets-examples	Servlet 2.4 Examples	true	0	Start Stop Reload Undeploy
/tomcat-docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy

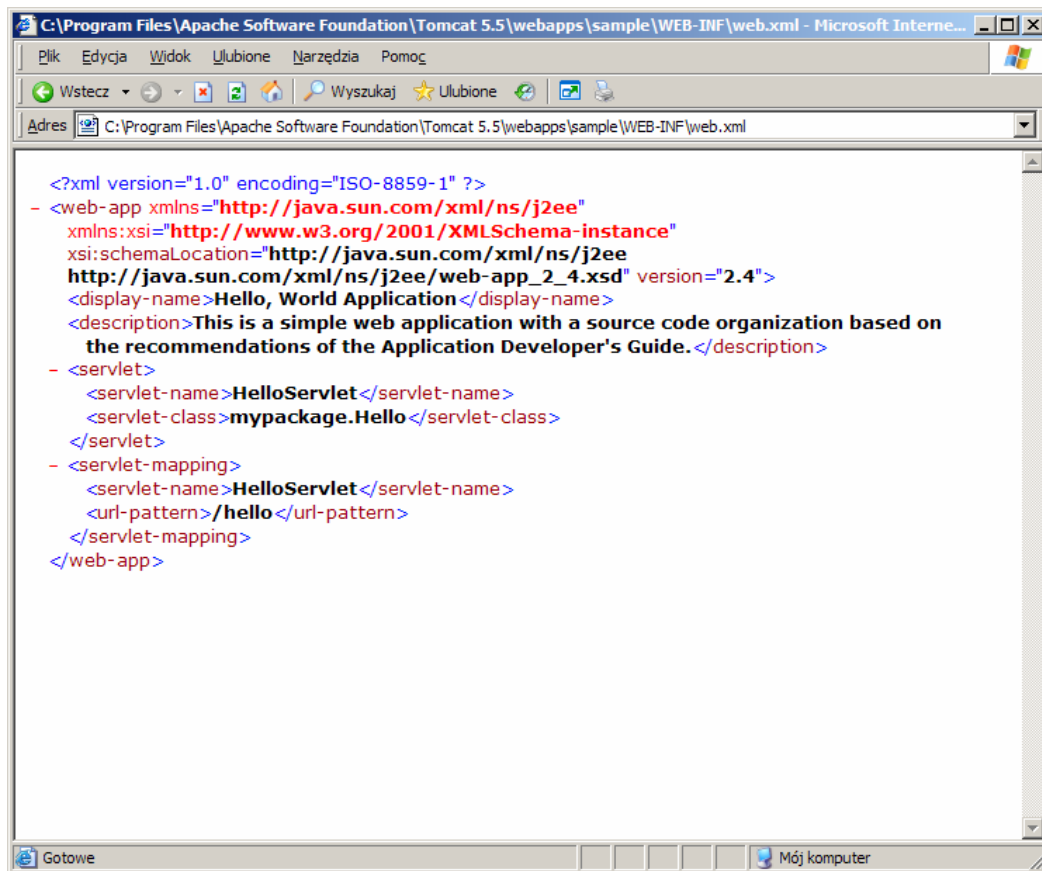
16. Kliknij na odnośnik /sample w bloku Applications i obejrzyj przykładową aplikację

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

To prove that they work, you can execute either of the following links:

- To a [JSP page](#).
- To a [servlet](#).

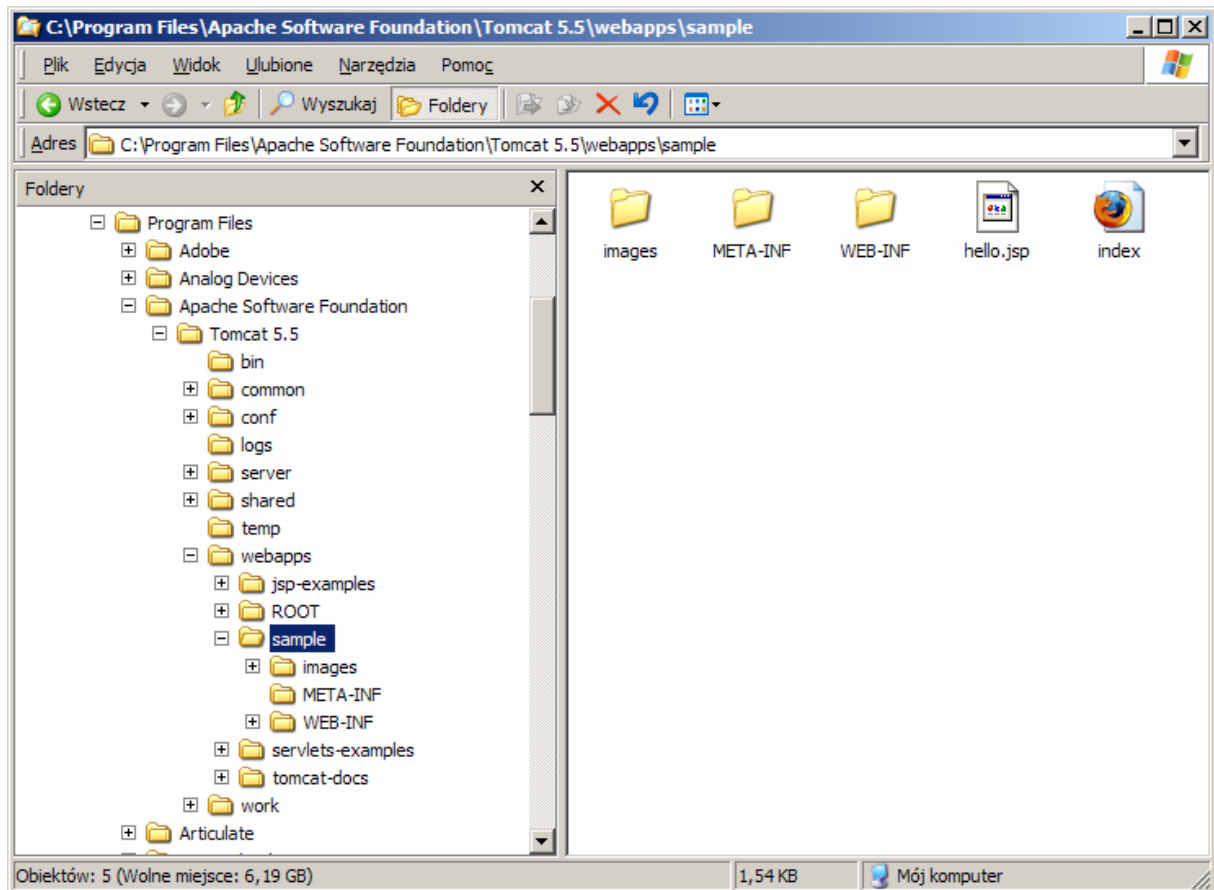
17. Zobaczmy teraz, jak aplikacja `/sample` jest fizycznie zlokalizowana na dysku. Przejdź do katalogu `%CATALINA_HOME%\webapps`. To jest katalog zawierający wszystkie aplikacje zainstalowane w serwerze aplikacji. Archiwum `sample.war` zostało załadowane właśnie tutaj. Pierwsze odwołanie do adresu URL <http://localhost:8080/sample> spowodowało automatyczne rozpakowanie tego pliku. Przejdź do katalogu `sample`. Katalog `images` zawiera wykorzystany na jednej ze stron obraz `tomcat.gif`. Katalog `META-INF` zawiera pusty plik `MANIFEST.MF`. W tym katalogu, opcjonalnie, mógłby się znaleźć deskryptor kontekstu aplikacji (czyli plik konfiguracyjny aplikacji internetowej). Katalog `WEB-INF` zawiera deskryptor wdrożenia aplikacji internetowej `web.xml`. Otwórz i zapoznaj się z zawartością tego pliku.



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">
  <display-name>Hello, World Application</display-name>
  <description>This is a simple web application with a source code organization based on
  the recommendations of the Application Developer's Guide.</description>
  - <servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>mypackage.Hello</servlet-class>
  </servlet>
  - <servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>
</web-app>
```



18. W pliku zdefiniowano jeden serwlet o nazwie `HelloServlet` zaimplementowany w postaci klasy `mypackage.Hello` i dostępny pod adresem `/hello` w ramach kontekstu aplikacji `/sample`. Otwórz adres <http://localhost:8080/sample/hello> i sprawdź, czy serwlet tam działa. W katalogu `WEB-INF` mieszczą się także źródła i postać skompilowana aplikacji internetowej. W podkatalogu `classes/mypackage` mieści się kod serwletu w pliku `Hello.class`, a źródło serwletu mieści się w podkatalogu `src/mypackage` w pliku `Hello.java`. Pliki `*.html` i `*.jsp` są umieszczone bezpośrednio w głównym katalogu `sample`.



19. Automatyczne rozpakowanie i instalacja aplikacji były możliwe ze względu na właściwą konfigurację serwera aplikacji. Podstawowym plikiem konfiguracyjnym jest plik `%CATALINA_HOME%\conf\server.xml`. Otwórz ten plik i znajdź w nim opcję `autoDeploy`. Poniższy fragment definiuje główny katalog w którym są umieszczane aplikacje (attribut `appBase`), automatyczne rozpakowywanie archiwów `*.war` (attribut `unpackWARs`), oraz automatyczną instalację rozpakowanych plików (attribut `autoDeploy`).

```
<Host name="localhost" appBase="webapps"  
  unpackWARs="true" autoDeploy="true"  
  xmlValidation="false" xmlNamespaceAware="false">
```

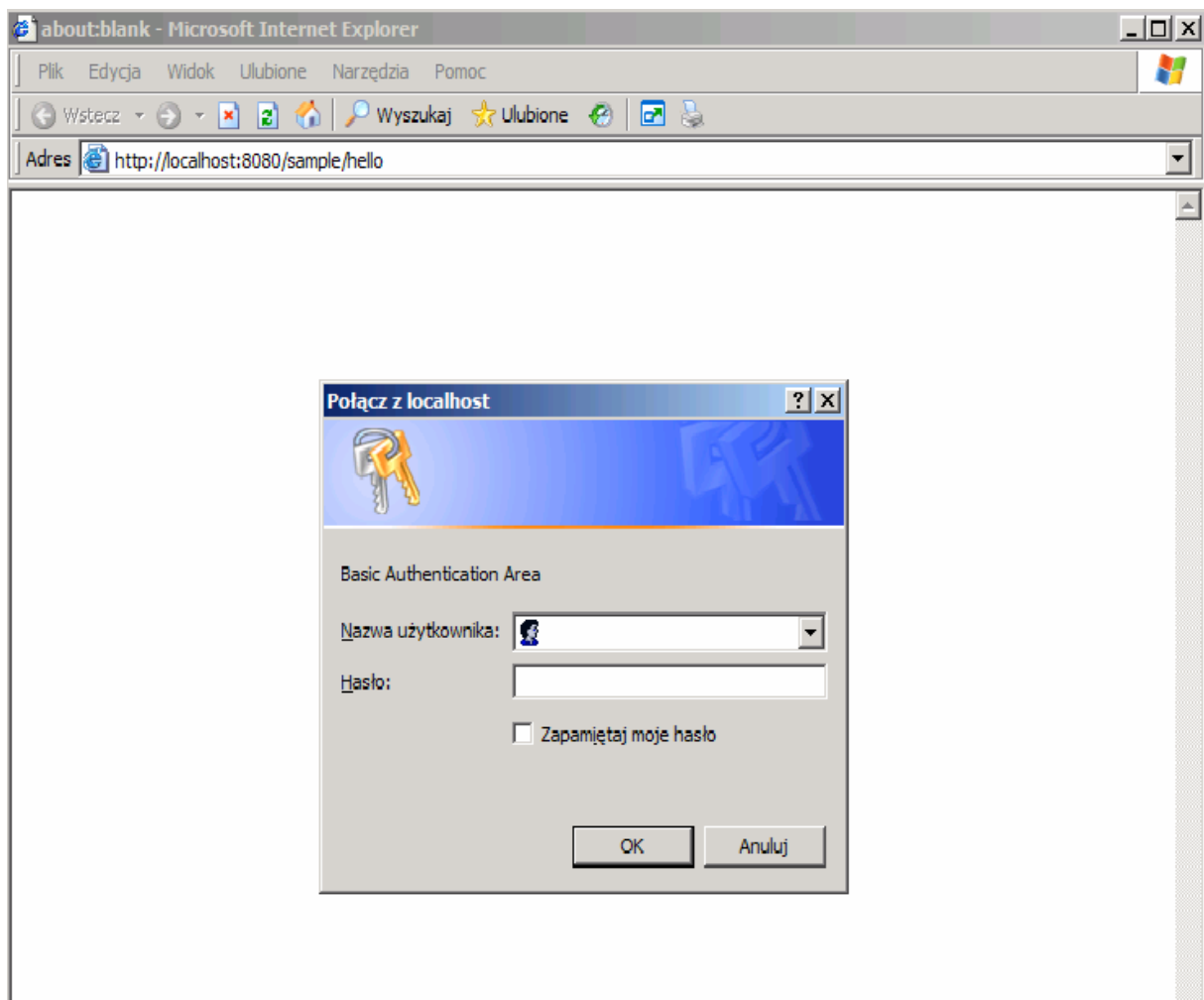
20. Wróć do katalogu %CATALINA\_HOME%\webapps\sample\WEB-INF i dodaj do pliku web.xml poniższy kod

```
<security-constraint>
  <display-name>Security Constraint</display-name>
  <web-resource-collection>
    <web-resource-name>Protected Area</web-resource-name>
    <url-pattern>/hello</url-pattern>
    <http-method>GET</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>helloRole</role-name>
  </auth-constraint>
</security-constraint>

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

21. Przejdź do okna zarządcy serwera aplikacji i przeładuj aplikację /sample (odnośnik Reload)

22. Spróbuj otworzyć w przeglądarce adres <http://localhost:8080/sample/hello>

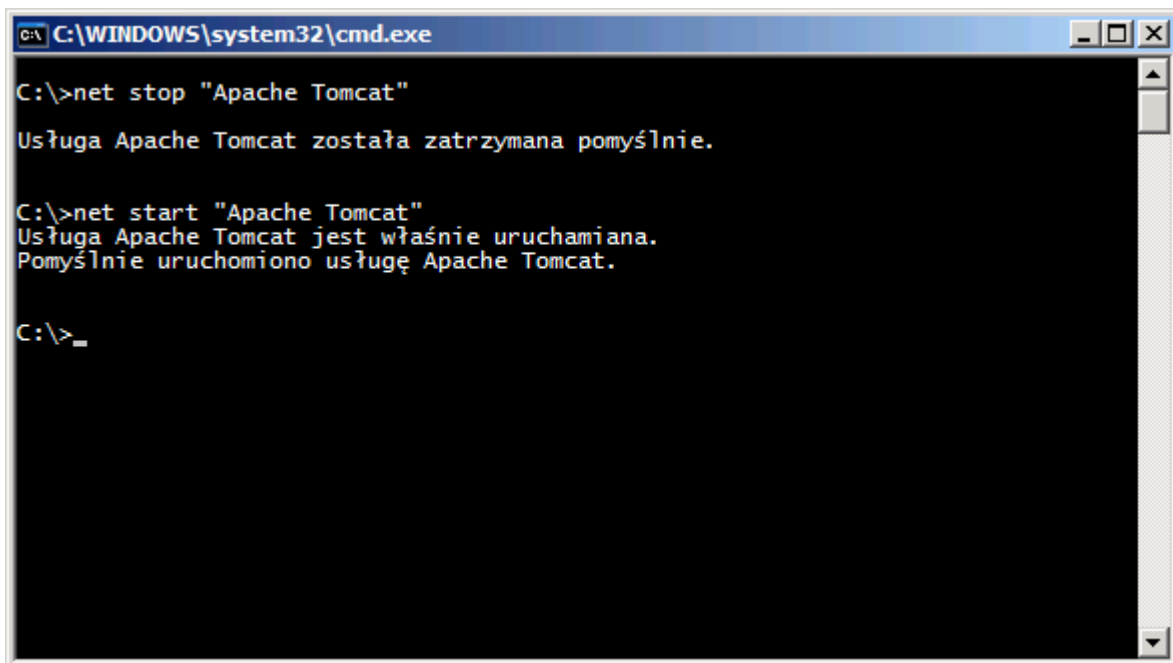


23. Kod wprowadzony do deskryptora wdrożenia `web.xml` definiuje ograniczenie bezpieczeństwa nazwane **Security Constraint** i obejmujące sobą adres URL `/hello` w kontekście aplikacji. Ograniczenie dotyczywołania tego adresu za pomocą metody GET. Jedynie użytkownicy posiadający rolę `helloRole` otrzymują dostęp do chronionego zasobu. Sposób logowania został ustawiony na **Basic**, co oznacza wykorzystanie wewnętrznego mechanizmu przeglądarki.

24. Przejdź do katalogu `%CATALINA_HOME%\conf\` i otwórz w dowolnym edytorze plik `tomcat-users.xml`. Dodaj do pliku poniższe linie

```
<role rolename="helloRole"/>
<user username="scott" password="tiger" roles="helloRole"/>
```

25. Restartuj serwer aplikacji. Otwórz konsolę MS-DOS i wydaj poniższe komendy



```
C:\WINDOWS\system32\cmd.exe
C:\>net stop "Apache Tomcat"
Usługa Apache Tomcat została zatrzymana pomyślnie.

C:\>net start "Apache Tomcat"
Usługa Apache Tomcat jest właśnie uruchamiana.
Pomyślnie uruchomiono usługę Apache Tomcat.

C:\>_
```

26. Ponownie otwórz w przeglądarce adres <http://localhost:8080/sample/hello> i wpisz nazwę i hasło utworzonego przez siebie użytkownika.

27. Serwer aplikacji Tomcat umożliwia także konfigurowanie połączeń do zewnętrznych zasobów, np. do baz danych. W kolejnym kroku zdefiniujemy połączenie z bazą danych na poziomie serwera aplikacji i wykorzystamy je do stworzenia testowej strony JSP. Przejdź do katalogu `%CATALINA_HOME%\conf\` i dodaj do pliku `server.xml`, tuż przed zamykającym znacznikiem `</Host>`, poniższy kod (oczywiście wpisz właściwe parametry połączenia z serwerem bazy danych i właściwe konto w bazie danych)

```
<Context path="/sample">
  <Resource name="jdbc/oracle" auth="Container"
    type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    url="jdbc:oracle:thin:@host:port:sid"
    username="scott" password="tiger"
    maxActive="20" maxIdle="10" maxWait="-1"/>
</Context>
```

28. Do prawidłowej pracy aplikacji konieczna jest dostępność sterownika JDBC dla bazy danych Oracle. Umieść sterownik w katalogu %CATALINA\_HOME%\common\lib (jest to katalog bibliotek wykorzystywanych przez serwer aplikacji wewnętrznie)
29. Aplikacja testowa będzie wykorzystywać bibliotekę znaczników JSTL. Umieść biblioteki JSTL w katalogu %CATALINA\_HOME%\shared\lib (jest to katalog bibliotek wykorzystywanych wspólnie przez aplikacje)
30. Restartuj serwer aplikacji
31. Przejdź do katalogu %CATALINA\_HOME%\webapps\sample i utwórz w nim nowy plik test.jsp. Umieść w pliku test.jsp poniższy kod

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<sql:query var="rs" dataSource="jdbc/oracle">
  select * from emp
</sql:query>

<html>
  <head>
    <title>DB Test</title>
  </head>
  <body>

    <h2>Results</h2>

    <c:forEach var="row" items="{rs.rows}">
      ${row.empno} ${row.ename} ${row.job}<br/>
    </c:forEach>

  </body>
</html>
```

32. Otwórz w przeglądarce adres <http://localhost:8080/sample/test.jsp>

