

Aplikacje WWW - laboratorium

Serwlety

Celem ćwiczenia jest przygotowanie kilku prostych serwletów ilustrujących możliwości tej technologii.

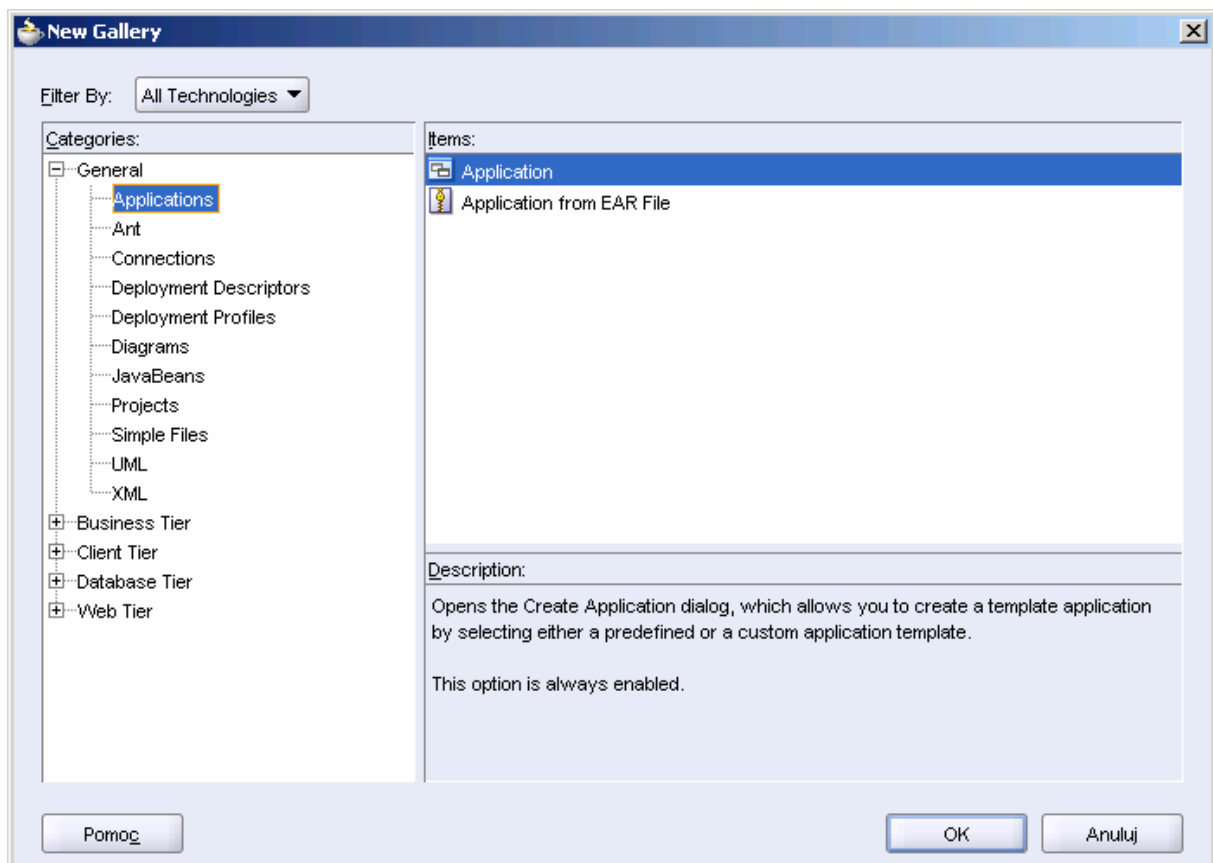
Poszczególne ćwiczenia prezentują sposób przygotowania środowiska, generowania i uruchamiania prostego serwletu oraz obsługę podstawowych metod i właściwości zdefiniowanych w serwetach.

Prezentowane ćwiczenia zostały wykonane w środowisku Oracle JDeveloper 10.1.3.0.4 (do pobrania ze strony: <http://www.oracle.com/>).

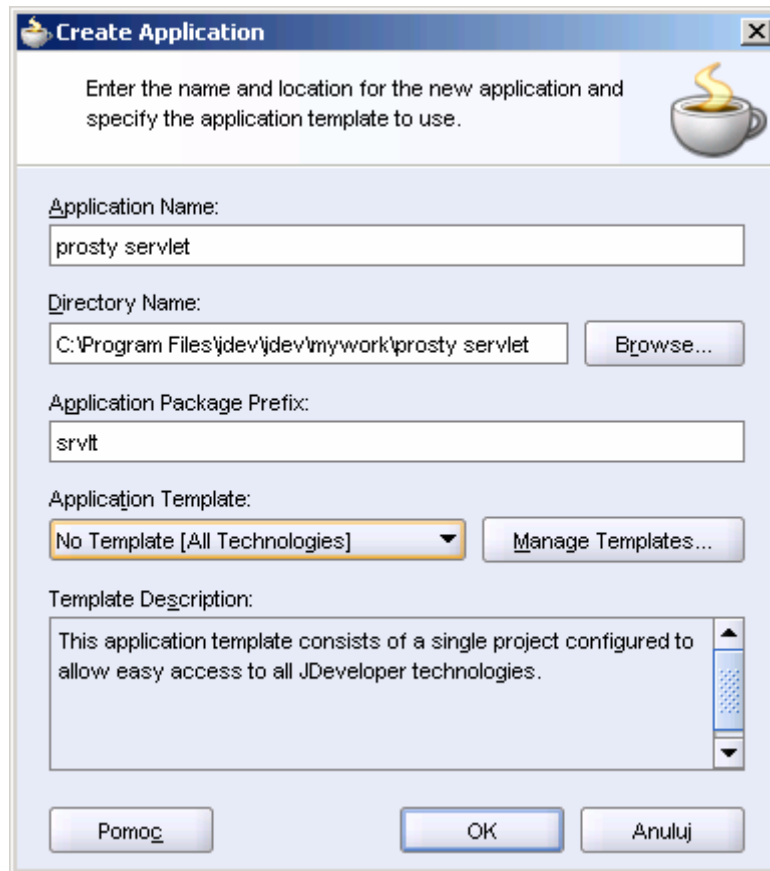
Ćwiczenie 1

W tym ćwiczeniu zostanie przygotowane środowisko pozwalające na wykonanie ćwiczeń – w repozytorium programu JDeveloper zostanie stworzona odpowiednia aplikacja i projekt.

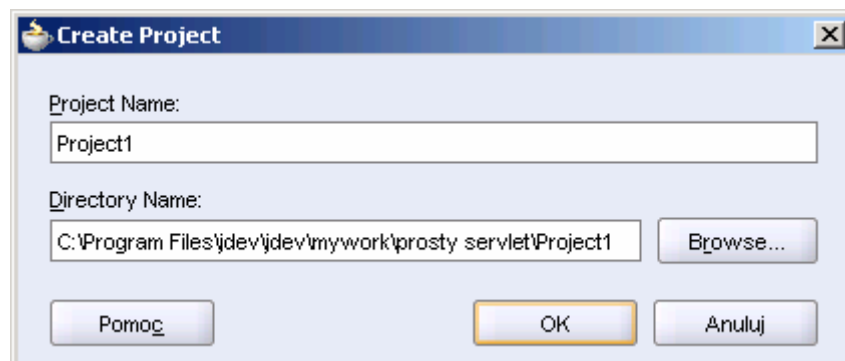
- Po uruchomieniu programu Oracle JDeveloper należy utworzyć nową aplikację. W tym celu należy wybrać opcję: File -> New
- W wyświetlonym oknie należy zaznaczyć chęć utworzenia nowej aplikacji (wygląd okna może się różnić od poniższego):



- Następny ekran pozwala na ustawienie właściwości tworzonej aplikacji. Należy wprowadzić: nazwę aplikacji, nazwę katalogu, w którym znajdą się jej pliki, prefiks dla pakietów tworzonych w ramach tej aplikacji i wybrać aplikację bez wzorca



- Kolejnym krokiem jest stworzenie projektu w ramach aplikacji

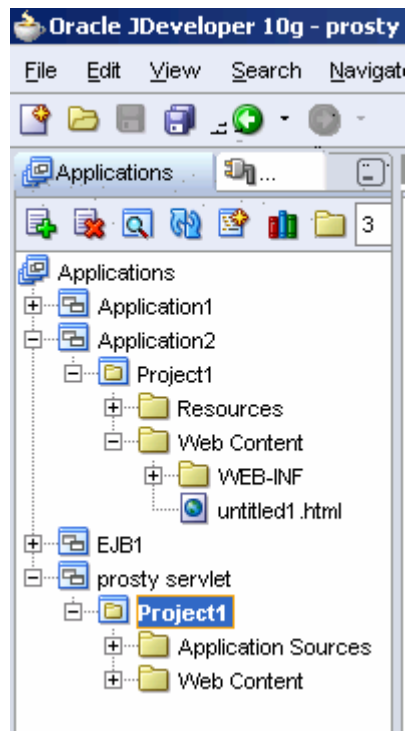


Ćwiczenie 2

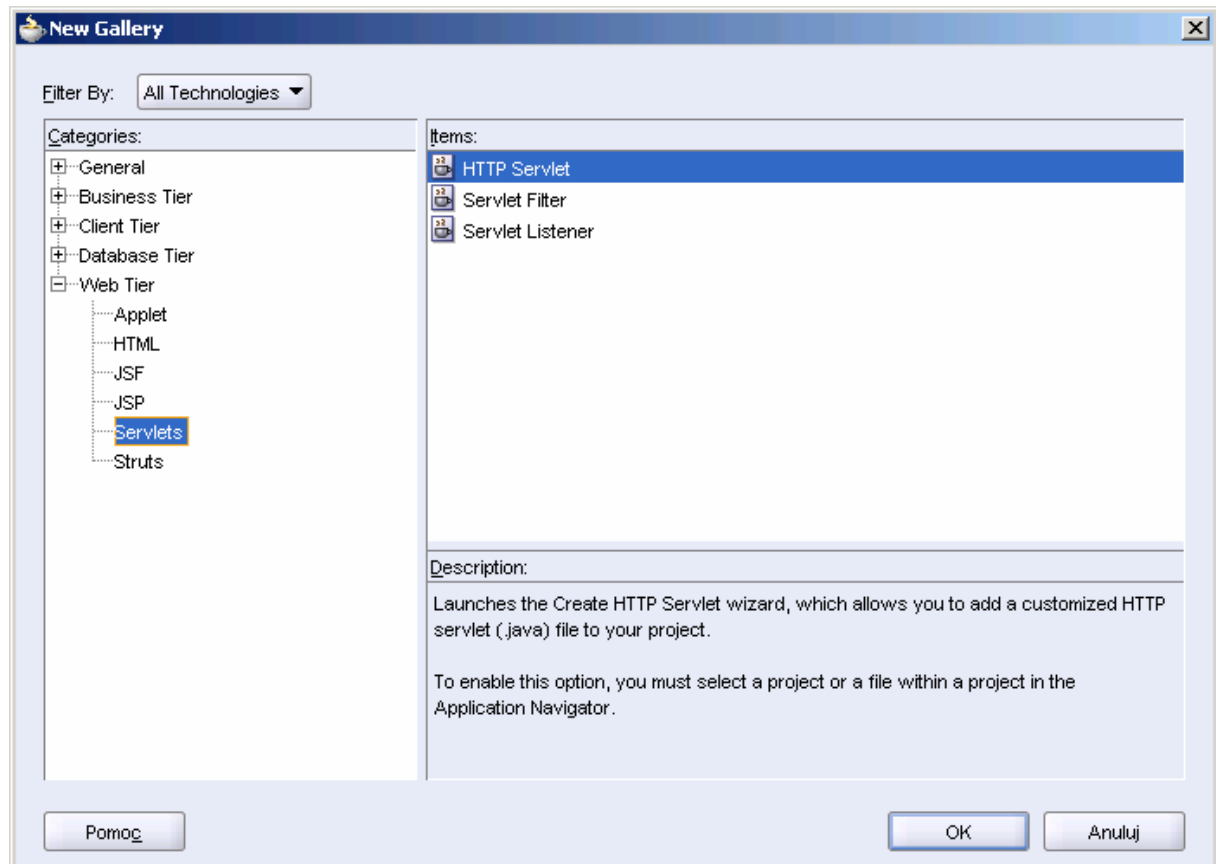
Ćwiczenie będzie polegało na wygenerowaniu szkieletu serwletu i uruchomieniu go.

W tym celu należy:

- W widocznym nawigatoryze aplikacji należy zaznaczyć utworzony w poprzednim ćwiczeniu projekt



- Następnie należy wybrać opcję: File -> New
- W wyświetlonym oknie należy wybrać chęć utworzenia nowego servletu (HTTP Servlet)



- W wyświetlonym kreatorze ustawiamy odpowiednie parametry

Create HTTP Servlet - Step 1 of 3: Servlet Information

Enter servlet details

Class:

Package: Browse...

Generate Content Type:

Generate Header Comments

Implement Methods

doGet() doPost() service()
 doPut() doDelete()

Pomoc < Wstecz **Dalej >** Zakończ Anuluj

Create HTTP Servlet - Step 2 of 3: Mapping Information

Enter servlet mapping.
While this is not required to create a servlet, it is required to run a servlet.

Specify a name and mapping for the servlet.

Mapping Details

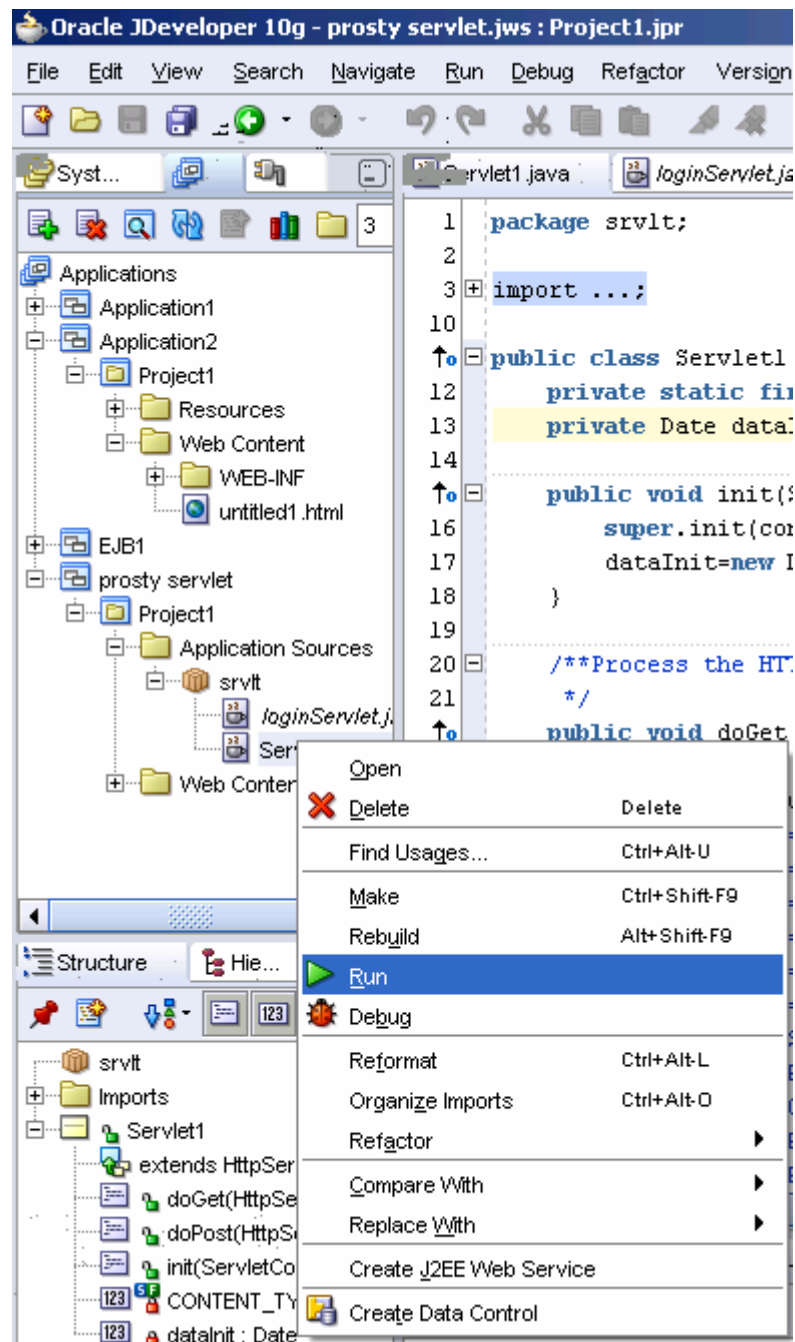
Name:

URL Pattern:

Pomoc < Wstecz **Dalej >** **Zakończ** Anuluj

- Po kliknięciu przycisku Zakończ zostanie wygenerowany kod serwletu.
- Należy obejrzeć ten kod, zidentyfikować jego podstawowe elementy (omówione na wykładzie metody, ich parametry, ...)

- Następnie należy uruchomić wygenerowany serwlet. W tym celu z menu kontekstowego (lub głównego) należy wybrać opcje Run



Ćwiczenie 3

W tym ćwiczeniu należy przeanalizować zachowanie pól serwletu, którym wartości są nadawane w różnych momentach działania serwletu (w metodach *init()* i *doGet()*)

- Do wygenerowanego w zadaniu 2 kodu należy dodać zlecenie importu biblioteki `java.util.Date`:

```
import java.util.Date;
```

- Następnie zadeklarować pole `dataInit`:

```
private Date dataInit;
```

- W metodzie `init` dopisać:

```
dataInit=new Date();
```

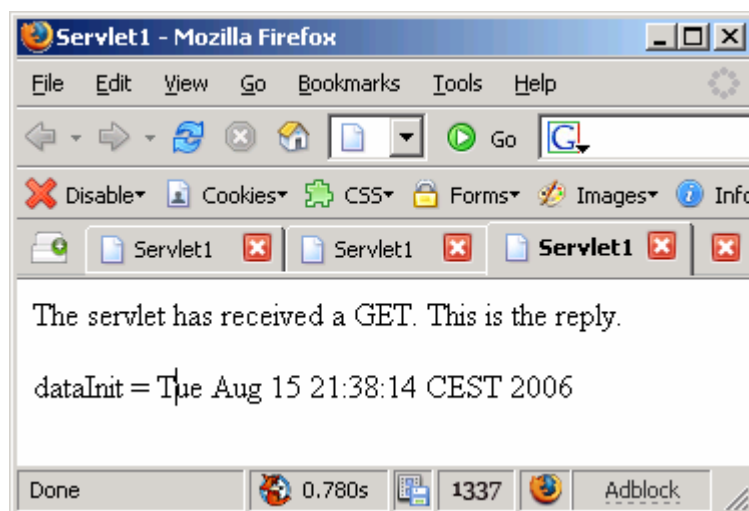
- W metodzie `doGet` dopisać:

```
out.println("<p>dataInit = "+dataInit+"</p>");
```

- Uruchomić serwlet

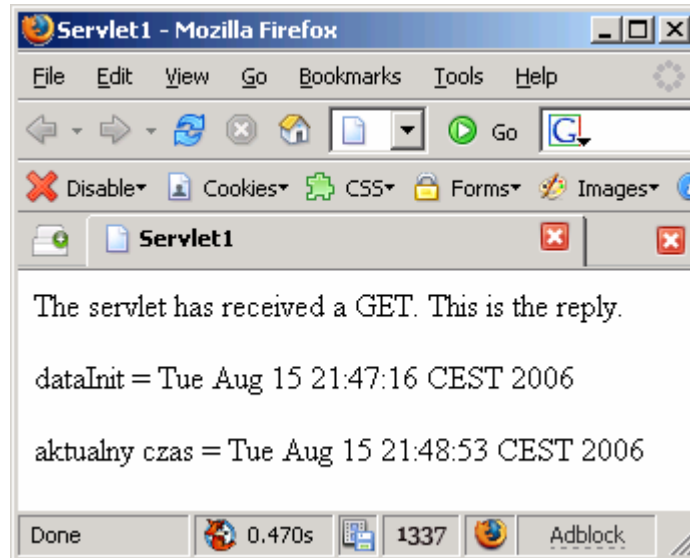


- Następnie w przeglądarce kilkakrotnie odświeżyć stronę. Czy wyświetlany czas zmienia się? Dlaczego tak się dzieje?

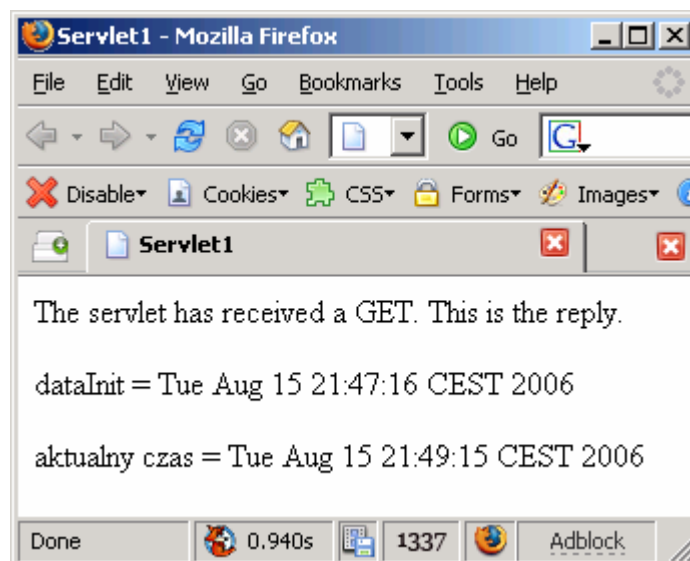


- Następnie w przeglądarce kilkakrotnie odświeżyć stronę. Czy wyświetlana wartość zmienia się? Dlaczego tak się dzieje?
- W metodzie doGet dopisać:

```
out.println("<p>aktualny czas = "+new Date()+"</p>");
```
- Uruchomić serwlet



- Następnie w przeglądarce kilkakrotnie odświeżyć stronę. Czy wyświetlany czas zmienia się? Dlaczego tak się dzieje?



Ćwiczenie 4

Celem tego ćwiczenia jest obserwacja środowiska pracy serwletu ustawianego przez serwer HTML

- Do kodu serwletu z poprzedniego zadania należy do metody doGet dodać polecenia wyświetlające metadane żądania HTTP

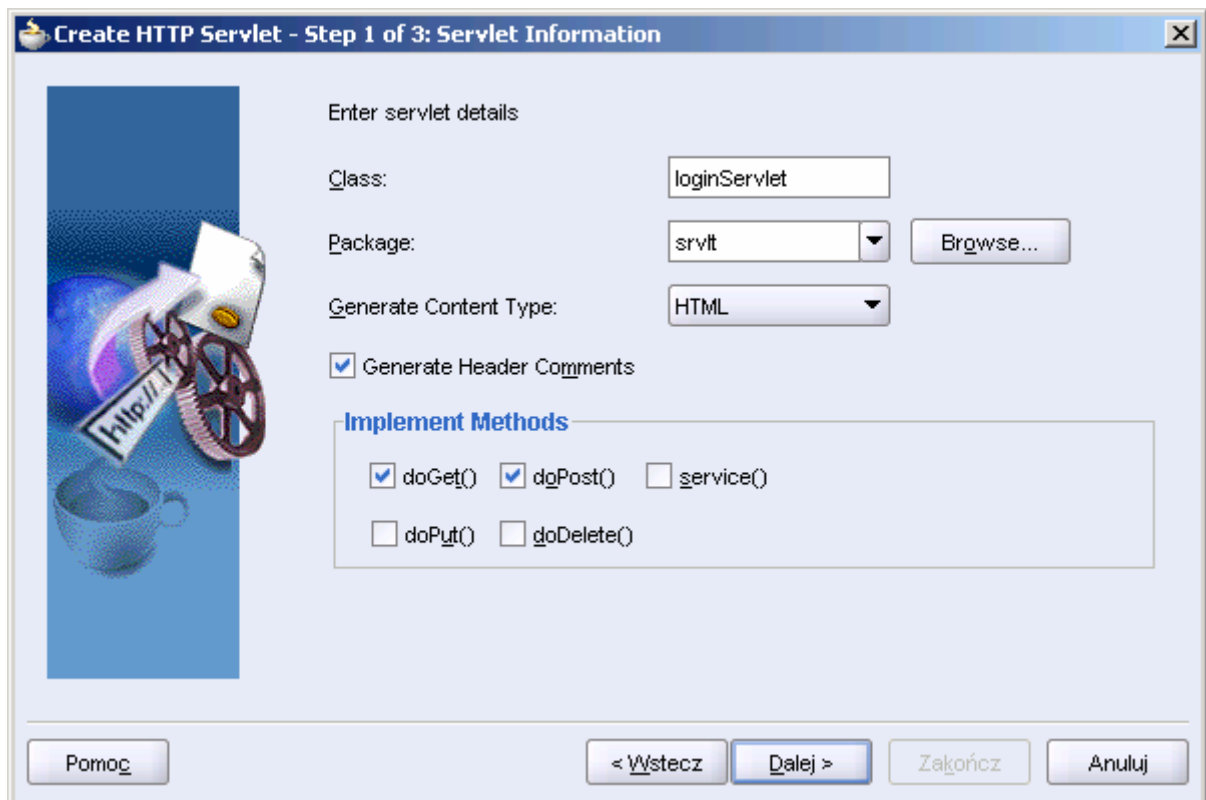
```
out.println("SERVER_NAME="+request.getServerName()+"<BR>");
out.println("REQUEST_METHOD="+request.getMethod()+"<BR>");
out.println("QUERY_STRING="+request.getQueryString()+"<BR>");
out.println("REMOTE_HOST="+request.getRemoteHost()+"<BR>");
out.println("REMOTE_ADDR="+request.getRemoteAddr());
```

- Uruchomić serwlet i przeanalizować wyniki

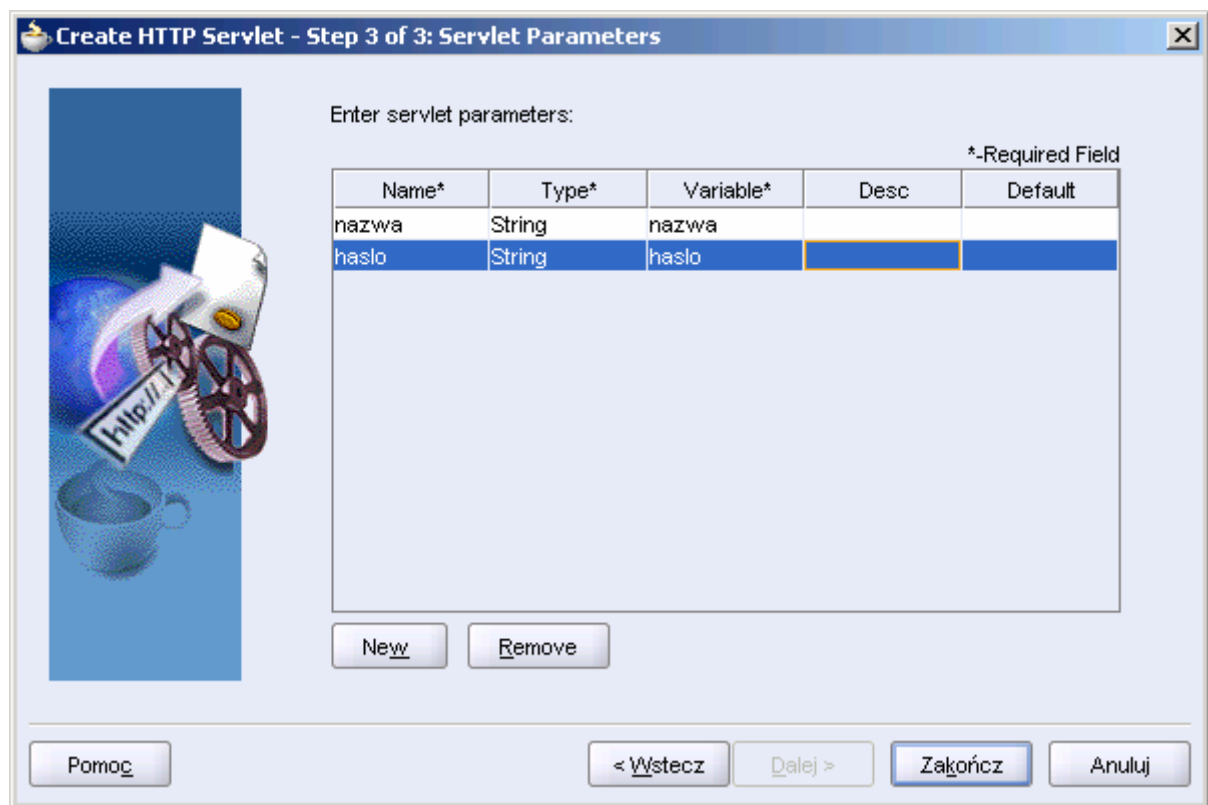
Ćwiczenie 5

W tej części ćwiczenia zostanie przygotowany serwlet, który wywołany przy pomocy metody GET wyświetli formularz logowania a wywołany przy pomocy metody POST sprawdzi dane przesłane przy pomocy tego formularza.

- Należy stworzyć nowy serwlet o nazwie: loginServlet, implementujący metody: doGet() i doPost()



- Serwlet będzie obsługiwał parametry o nazwach: nazwa i hasło



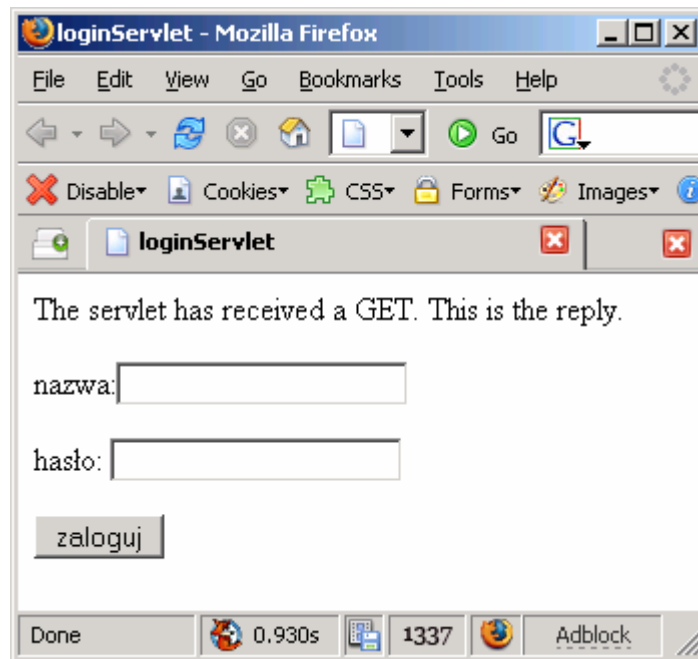
- W metodzie doGet dopisać:

```

out.println("<form name=\"login_form\" action=\"
loginervlet\" method=\"post\">");
out.println("    <p>");
out.println("        nazwa:<input type=\"text\" name=\"nazwa\"
        maxlength=\"20\" size=\"20\"/>");
out.println("    </p>");
out.println("    <p>");
out.println("        haslo:");
out.println("        <input type=\"password\" name=\"haslo\"
        maxlength=\"20\" size=\"20\"/>");
out.println("    </p>");
out.println("    <p>");
out.println("        <input type=\"submit\" name=\"zaloguj\"
        value=\"zaloguj\"/>");
out.println("    </p>");
out.println("</form>");

```

- Następnie należy uruchomić serwlet i sprawdzić poprawność wyświetlania formularza. Można spróbować wprowadzić dane i sprawdzić jego działanie.



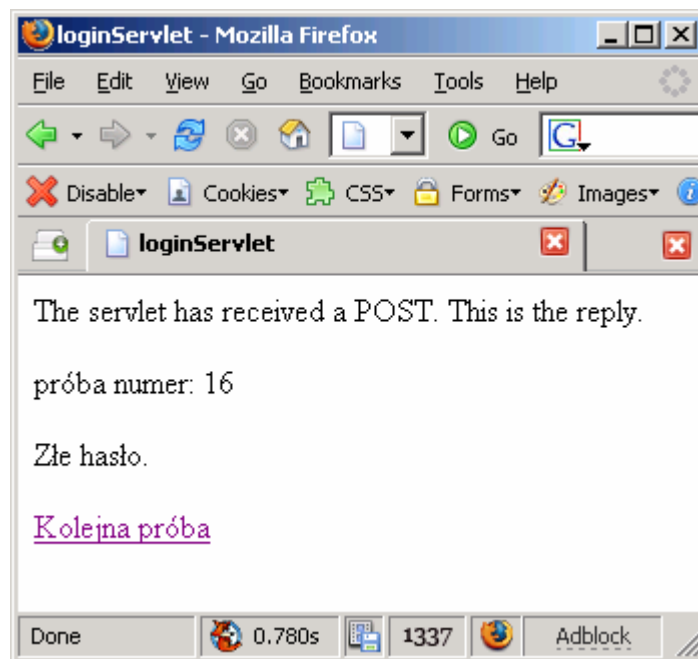
- W kolejnym kroku należy zadeklarować pole licznik:

```
int licznik=0;
```

- W metodzie doPost dopisać:

```
out.println("<p>próba numer: "+ (++licznik) + "</p>");  
if (nazwa.equals(haslo)) {  
    out.println("<p>Złe hasło.</p>");  
    out.println("<p><a href=\"loginServlet\">Kolejna  
próba</a></p>");  
} else {  
    out.println("<p>Hasło poprawne.</p>");  
}
```

- Następnie należy uruchomić serwlet i przeanalizować jego działanie



- Następnie należy uruchomić drugie okno przeglądarki i w polu adresu wprowadzić adres serwletu. Pracując jednocześnie z dwoma oknami – obserwować działanie mechanizmu obsługi przesyłanych danych oraz licznika. Czy działanie licznika jest poprawne i akceptowalne? Dlaczego?

Ćwiczenie 6

Obsługa Cookies i sesji (do samodzielnego wykonania).

- Należy tak zmodyfikować kod z poprzedniego zadania, aby stan licznika był przechowywany w zmiennych Cookies i zmiennych sesyjnych.
- Następnie należy uruchomić przeglądarkę i przeanalizować zachowanie się obu liczników w sytuacjach, kiedy użytkownik „bez przerwy klika” po stronach serwisu, kiedy użytkownik robi sobie dłuższe przerwy oraz kiedy użytkownik zamknie przeglądarkę i uruchomi ją ponownie (lub ma uruchomione dwa okna przeglądarki). Jakie są różnice w zachowaniu tych liczników? Z czego one wynikają?