

Zaawansowane aplikacje internetowe

EJB 2

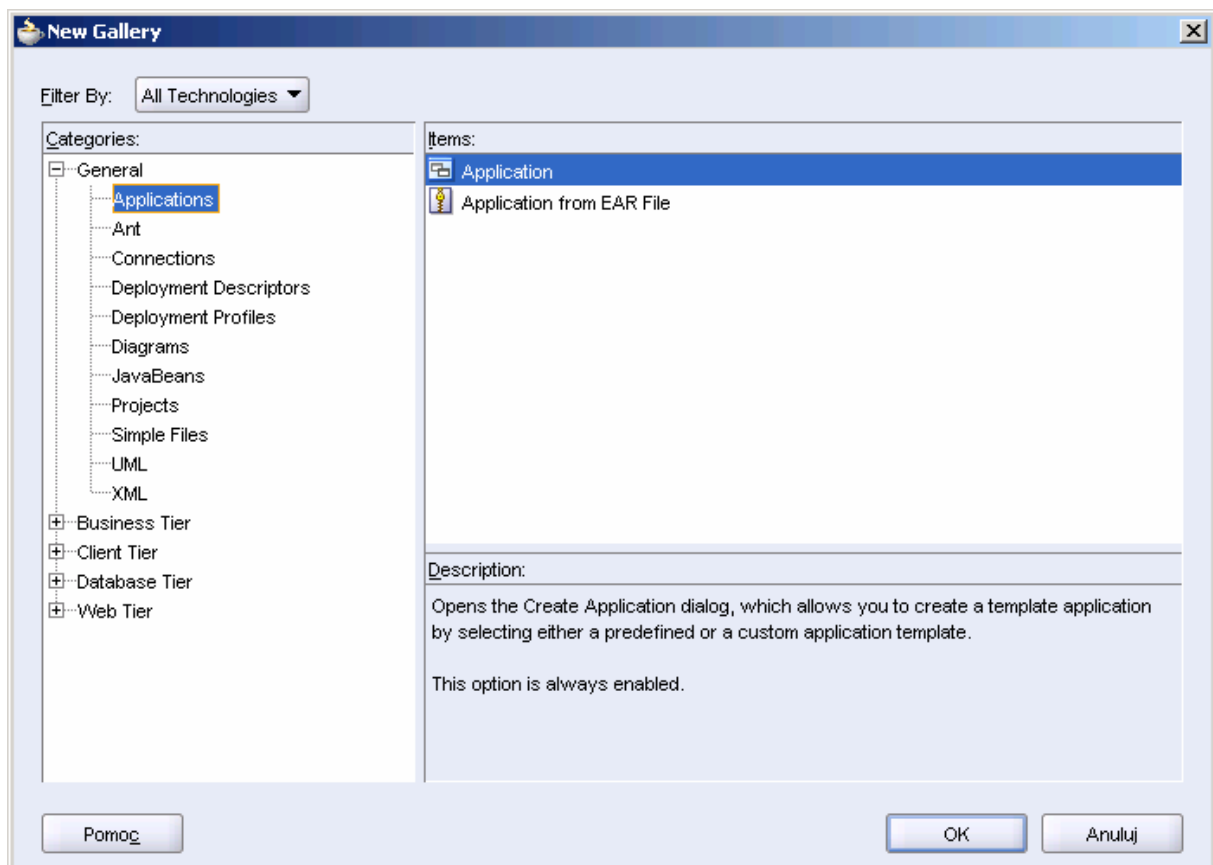
Celem tego laboratorium jest pokazanie, w jaki sposób aplikacje stworzone w różnych technologiach mogą korzystać z funkcjonalności udostępnianej przez komponenty EJB. W ramach ćwiczeń powstaną kolejno: serwlet oraz inny *bean* EJB, korzystające z funkcjonalności beana utworzonego w ćwiczeniu pierwszym.

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. A następnie zostanie przygotowany testowy *bean*.

- 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

Enter the name and location for the new application and specify the application template to use.

Application Name:
EJB2

Directory Name:
C:\Program Files\jdev\jdev\mywork\EJB2 Browse...

Application Package Prefix:

Application Template:
No Template [All Technologies] Manage Templates...

Template Description:
This application template consists of a single project configured to allow easy access to all JDeveloper technologies.

Pomoc OK Anuluj

- Kolejnym krokiem jest stworzenie projektu w ramach aplikacji – można pozostawić domyślną nazwę: Project1

Project Name:
Project1

Directory Name:
C:\Program Files\jdev\jdev\mywork\ejb2\Project1 Browse...

Pomoc OK Anuluj

- Następnie należy utworzyć testowego *beana* sesyjnego o nazwie *TestowyEJB* (dla potrzeb tego ćwiczenia nie ma znaczenia czy będzie on stanowy, czy też bezstanowy) udostępniającego następującą implementację metody *getHello*:

```
public String getHello(String kogo_witamy)
{
    return "Hello " + kogo_witamy;
}
```

- *Beana* należy uruchomić.
- Następnie należy wygenerować przykładową aplikację kliencką dla tego *beana* (szczegóły znajdują się w opisie do pierwszego zestawu ćwiczeń omawiających technologię EJB)
- W aplikacji testowej umieścić następujące wywołanie udostępnionej metody *beana*

```
System.out.println(testowyEJB.getHello("Student"));
```

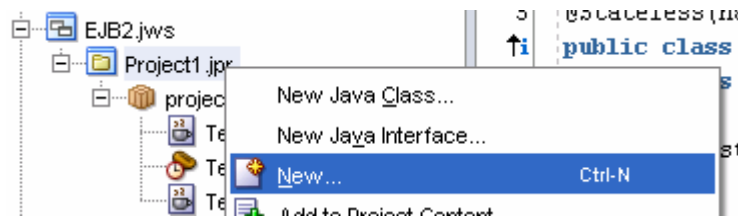
- Sprawdzić działanie aplikacji klienckiej

Ćwiczenie 2

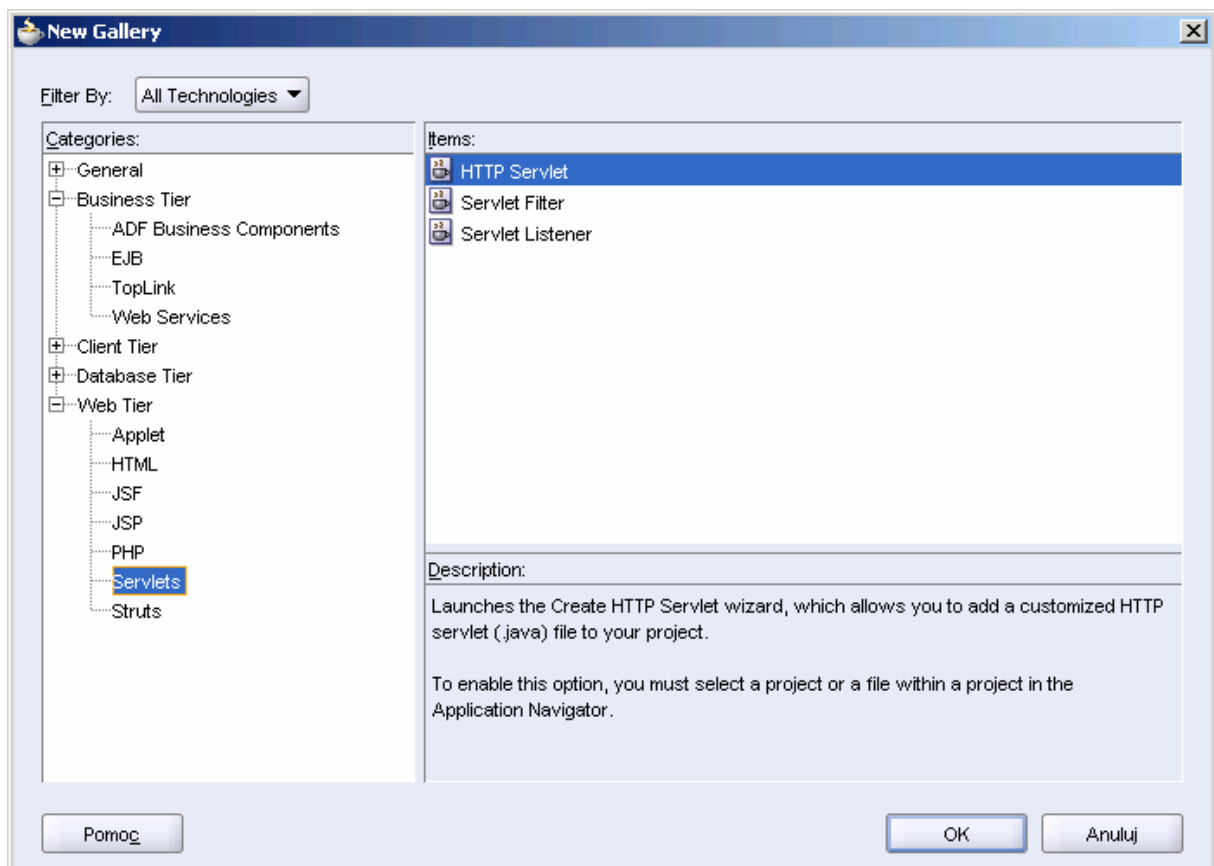
Ćwiczenie będzie polegało na stworzeniu serwletu, uruchamiającego metodę utworzonego wcześniej *beana*.

W tym celu należy:

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



- Następnie należy wybrać opcję: File -> New, lub z menu kontekstowego wybrać opcję New
- W wyświetlonym oknie należy wybrać chęć utworzenia serwletu



- Należy określić nazwę i typ dla tworzonego serwletu (można pozostawić proponowaną przez system). Implementowana będzie tylko metoda doGet()

Enter servlet details

Class:

Package:

Generate Content Type:

Generate Header Comments

Implement Methods

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

Pomoc < Wstecz Dalej > Zakończ Anuluj

- Do utworzonej klasy serwletu należy zaimportować pakiet

```
import javax.naming.InitialContext;
```

- Następnie wewnątrz metody doGet() utworzonego serwletu dodać poniższy kod

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException
{
    response.setContentType(CONTENT_TYPE);
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Servlet1</title></head>");
    out.println("<body>");
    out.println("<p>The servlet has received a GET. This is
                the reply.</p>");

    try
    {
        final Context context = new InitialContext();
        TestowyEJB testowyEJB =
            (TestowyEJB)context.lookup("TestowyEJB");
        // Call any of the Remote methods below to access the
        // EJB
        out.println(testowyEJB.getHello( "Student" ));
    }
    catch (Exception ex)
    {
        ex.printStackTrace();
    }
    out.println("</body></html>");
    out.close();
}
```

- Na koniec należy uruchomić serwlet

Ćwiczenie 3

- W tym ćwiczeniu należy utworzyć *beana* o nazwie klientEJB, który podobnie jak serwlet z poprzedniego ćwiczenia wywoła metodę beana o nazwie testowyEJB.
- Następnie stworzyć przykładową aplikację dla tego beana i sprawdzić jej działanie.