

Zaawansowane aplikacje internetowe

AJAX 1

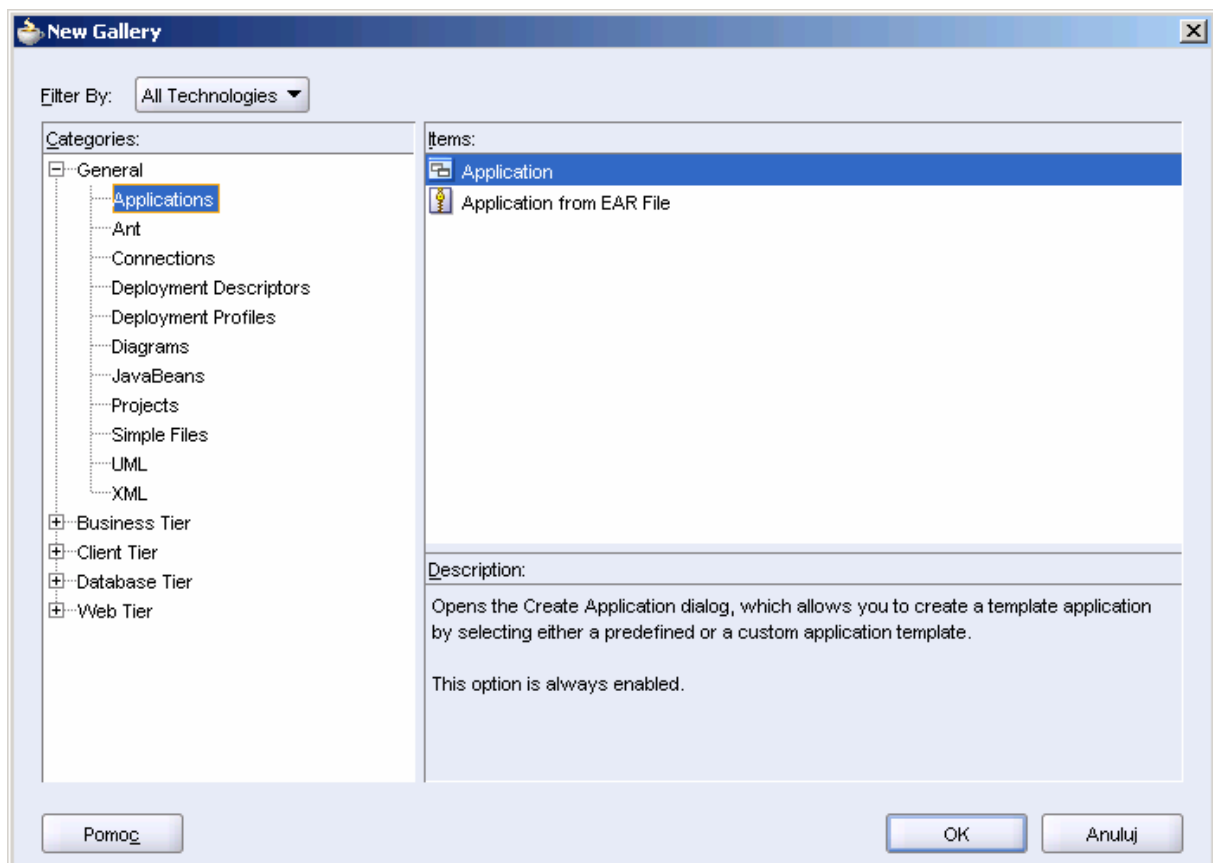
Celem tego laboratorium jest pokazanie możliwości technologii AJAX. W ramach ćwiczeń zostanie zbudowana prosta aplikacja, przechwytyjąca kliknięcia użytkownika w umieszczony na stronie przycisk, następnie wysyłający żądanie do serwletu, zwracającego aktualny czas systemowy. W dalszej części program zostanie zmodyfikowany tak, że będzie modyfikował swoje działanie w zależności od wartości wprowadzonych do formularza.

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

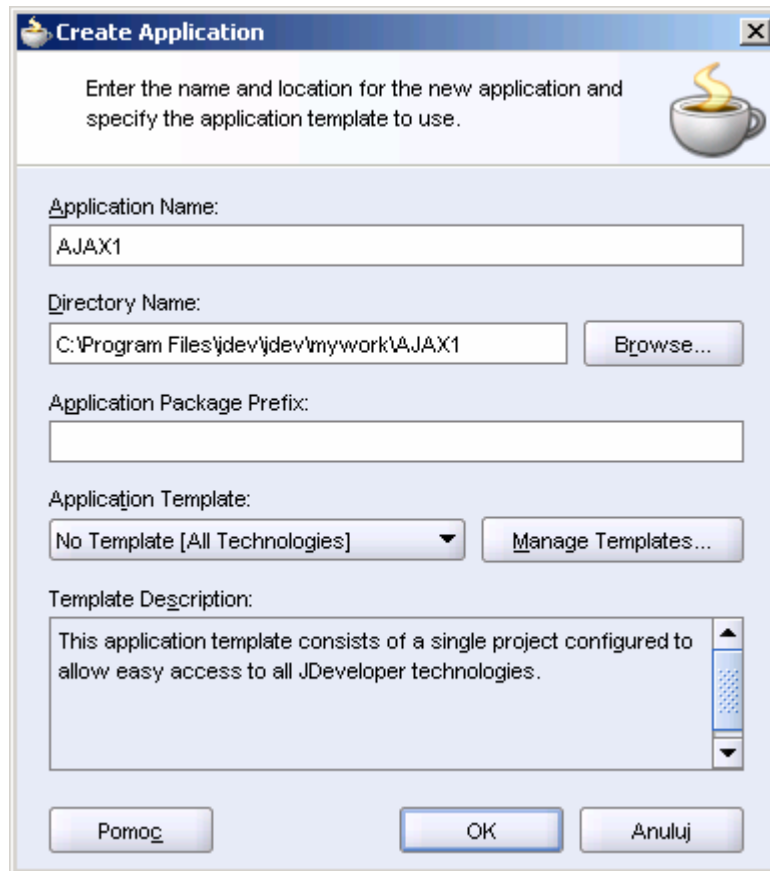
Ćwiczenie 1

W tym ćwiczeniu zostanie przygotowane środowisko pozwalające na wykonanie ćwiczeń a następnie zostanie utworzony przykładowy dokument wspierający technologię AJAX.

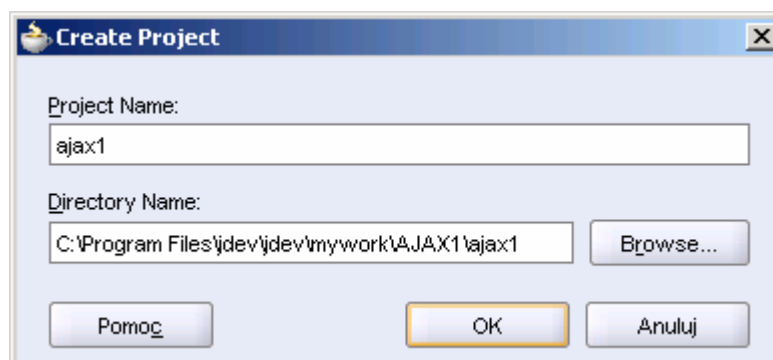
- Po uruchomieniu programu Oracle JDeveloper należy utworzyć nową aplikację. W tym celu należy wybrać opcję: File -> New
- W wyświetlonym oknie należy zaznaczyć chęć utworzenia nowej aplikacji (wygląd okna może się różnić od poniższego):



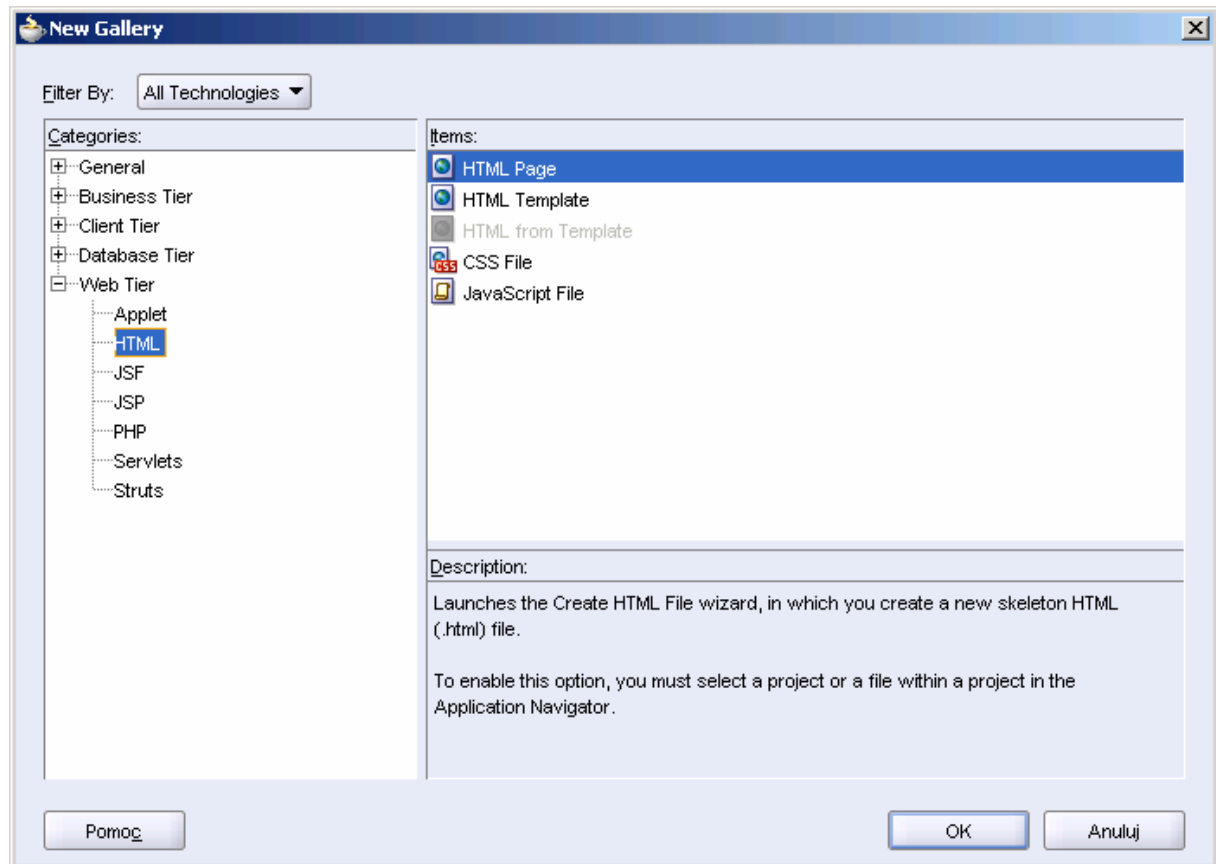
- Następny ekran pozwala na ustawienie właściwości tworzonej aplikacji. Należy wprowadzić: nazwę aplikacji, nazwę katalogu, w którym znajdą się jej pliki, prefiks dla pakietów tworzonych w ramach tej aplikacji i wybrać aplikację bez wzorca



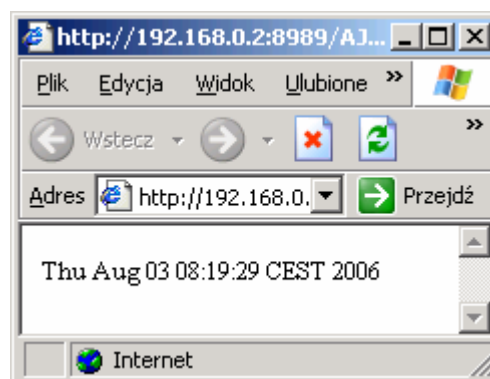
- Kolejnym krokiem jest stworzenie projektu w ramach aplikacji – można pozostawić domyślną nazwę: Project1



- Następnie należy utworzyć stronę w języku HTML (plik o nazwie *index.html*), przez którą użytkownik będzie się kontaktował z serwerem



- W podobny sposób należy też utworzyć plik, w którym zostanie zawarty kod w języku JavaScript, dołączany do nagłówek dokumentów HTML, a zawierający kod do obsługi technologii AJAX (plik: *ajax.js*)
- Stwórz prosty serwlet zwracający bieżącą datę



- Korzystając z komponentu *script* dostępnego w palecie komponentów, należy dołączyć utworzony plik ze skryptem (na razie pusty) do sekcji HEAD dokumentu HTML.



- Poniższy kod należy umieścić w pliku *ajax.js* - jest to przykład fragment wspólnego dla wszystkich dokumentów w ramach projektu, korzystających z technologii AJA

```

var xmlHttpRequestHandler = new Object();
xmlHttpRequestHandler.createXmlHttpRequest = function(){

    var XmlHttpRequestObject;
    if (typeof XMLHttpRequest != "undefined"){
        XmlHttpRequestObject = new XMLHttpRequest();
    }
    else if (window.ActiveXObject){
        // look up the highest possible MSXML version
        var tryPossibleVersions=["MSXML2.XMLHttp.5.0",
                                "MSXML2.XMLHttp.4.0",
                                "MSXML2.XMLHttp.3.0",
                                "MSXML2.XMLHttp",
                                "Microsoft.XMLHttp"];

        for (i=0; i< tryPossibleVersions.length; i++){
            try{
                XmlHttpRequestObject = new ActiveXObject(tryPossibleVersions[i]);
                break;
            }
            catch (xmlHttpRequestObjectError){
                //ignore
            }
        }
    }
    return XmlHttpRequestObject;
}

```

- W sekcji HEAD dokumentu HTML należy też umieścić kod specyficzny dla danego dokumentu

```

<script type="text/javascript">
function doTheAjaxThing(){
    var PAGE_SUCCESS = 200;
    var requestObject = xmlhttpRequestHandler.createXmlHttpRequest();
    requestObject.open("Get","servlet1",false);
    requestObject.send(null);

    if (requestObject.status==PAGE_SUCCESS){
        var div_handle = document.getElementById("message");
        //check if DIV element is found
        if(div_handle){
            div_handle.innerHTML+=' '+requestObject.responseText;
        }
    }
    else{
        alert ("Request failed");
    }
}
</script>

```

- W sekcji BODY dokumentu HTML należy umieścić następujący kod

```

<input type="button" value="Pobierz czas" onclick="doTheAjaxThing();" />
<div id="message">
</div>

```

- Po uruchomieniu dokumentu i naciśnięciu widocznego w przeglądarce przycisku – można już obserwować działanie programu.

Ćwiczenie 2

W tym ćwiczeniu będzie można zaobserwować wsparcie technologii AJAX dla dynamicznej obsługi formularzy.

- Należy utworzyć nowy dokument HTML o nazwie index2.html
- W sekcji HEAD należy dołączyć plik ajax.js
- Należy stworzyć serwlet o nazwie *servlet2*, który dla akceptuje parametr o nazwie imię i zwraca tekst: Witaj Przekazane_imie
- Należy zmodyfikować (w stosunku do poprzedniego ćwiczenia) początek funkcji doTheAjaxThing, aby doprowadzić ją do poniższej postaci

```

function doTheAjaxThing(){
    var requestObject;
    var PAGE_SUCCESS = 200;
    var param="imie="+ (document.form1.imie.value.length >0
        ?document.form1.imie.value:"nobody");

    requestObject = xmlhttpRequestHandler.createXmlHttpRequest();
    requestObject.open("Get","servlet2?" +param,false);
    requestObject.send(null);
}

```

- Należy zmodyfikować (w stosunku do poprzedniego ćwiczenia) zawartość sekcji BODY dokumentu HTML, aby doprowadzić ją do poniższej postaci

```

<form name="form1" action="">
    <input type="button" value="Wyswietl powitanie"
        onclick="doTheAjaxThing();" />
    <input type="text" id="imie" name="imie" />
</form>
<div id="message" class="message"></div>

```

- Po uruchomieniu dokumentu i naciśnięciu widocznego w przeglądarce przycisku – można już obserwować działanie programu.

Ćwiczenie 3

Należy (**samodzielnie**) przygotować formularz, pozwalający na wpisanie dwóch liczb, oraz wybór jednego spośród 4 działań matematycznych. Po stronie przeglądarki, jeszcze przed wysłaniem żądania wykonania obliczeń należy sprawdzić, czy w przypadku operacji dzielenia dzielna nie jest przypadkiem zerem i w takiej sytuacji wyświetlić odpowiedni komunikat, kończąc działanie skryptu. W pozostałych sytuacjach należy przesłać dane na serwer i wyświetlić odpowiedź.

Logikę realizującą przetwarzanie po stronie serwera należy przygotować w formie skryptu PHP lub JSP.

Wynik działania „serwerowej” części aplikacji powinien być ciągiem znaków o następującej treści:

pierwsza_liczba operator druga_liczba = wynik