

Pizza & Beer: Wprowadzenie do serwisów REST za pomocą WebHandlerów OE

**Spotkania techniczne dla partnerów aplikacyjnych
i klientów technologii Progress**

Agenda

SOA - Service Interface

Tworzenie własnych WebHandlerów

Konfiguracja WebHandlerów

Demo

SOA - Service Interface

Warstwa pośrednia (translation layer) pomiędzy żądaniem a zasadniczym serwisem biznesowym

- Przekierowywanie żądań (request)
- Komponowanie odpowiedzi (response)
- Obsługa błędów

Zapewnia uwierzytelnianie i autoryzację

Tłumaczenie formatów wejścia / wyjścia z/do modelu domeny

- JSON/XML/text na obiekty ProDataSet/Temp-Table
- Walidacja danych

SOA - Service Interface

REST (Mapped RPC)

- OE 11.2.0
- Elastyczny URI
- Narzędzia GUI do mapowania elementów HTTP do parametrów wejścia/wyjścia

REST (Data Object)

- OE 11.2.0
- Dawniej Mobile Services
- Adnotacje wybranych metod
- Ściśle narzucony model programowania oraz URI
- Tworzony jest Data Service Catalog

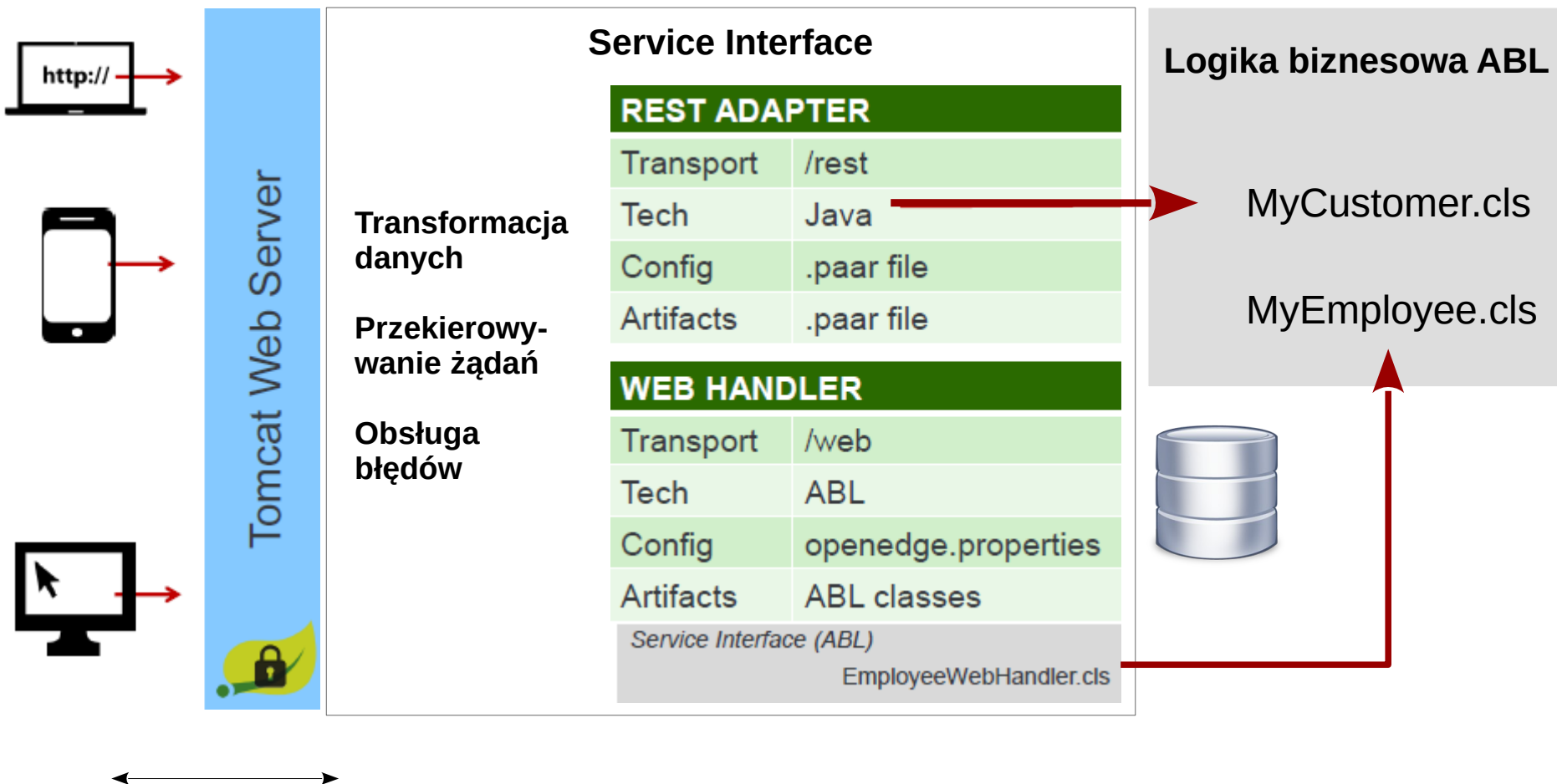
SOA - Service Interface

WebHandler

- OE 11.6.0
- WebHandler (klasa OO ABL) skojarzona jest z wzorcem URI
- Bardzo elastyczny
- Możliwość skorzystania z gotowych WebHandlerów lub napisania własnych
- Wszystkie czasowniki (verbs) HTTP
- Pełna obsługa żądania (request) HTTP
- Elastyczna kontrola odpowiedzi (response) HTTP

Architektura PASOE (11.6.0+)

Instancja PASOE



Tworzenie własnych WebHandlerów



Klasa OpenEdge.Web.WebHandler

Superklasa dla tworzenia WebHandlerów

Zawiera metody do obsługi HTTP verbs:

- HandleGet
- HandlePost
- HandleDelete
- HandlePatch
- HandlePut
- inne

The screenshot shows a dialog box for creating a new class. The 'WebHandler name' field contains 'myWebHandler'. The 'Modifiers' section has four unchecked checkboxes: 'Final', 'Abstract', 'Widget pool', and 'Serializable'. The 'Inherits' field contains 'OpenEdge.Web.WebHandler' and has a 'Browse...' button to its right. The 'Implements' field is empty and has 'Add...' and 'Remove' buttons to its right. Below the fields, the text 'Specify the code elements to be generated:' is followed by 'Method stubs:' and four checkboxes: 'GET' (checked), 'PUT', 'POST', and 'DELETE'.

Interfejs OpenEdge.Web.IWebRequest

Standardowy interfejs dla obsługi żądań PASOE/WEB

- PathInfo
- PathParameterNames
- TransportPath
- UriTemplate
- Wiele innych

Klasa OpenEdge.Web.WebResponse

**Reprezentacja obiektowa
odpowiedzi HTTP**

**Można jej użyć do
skomponowania odpowiedzi**

Metody nagłówka i cookie

WebResponse

«...»

+ StatusCode : INTEGER

+ ContentType : CHAR

+ Entity : Object

+ ContentLength : INTEGER

+ TransferEncoding : CHAR

«...»

+ SetHeader(pcName : CHAR, pcValue : CHAR)

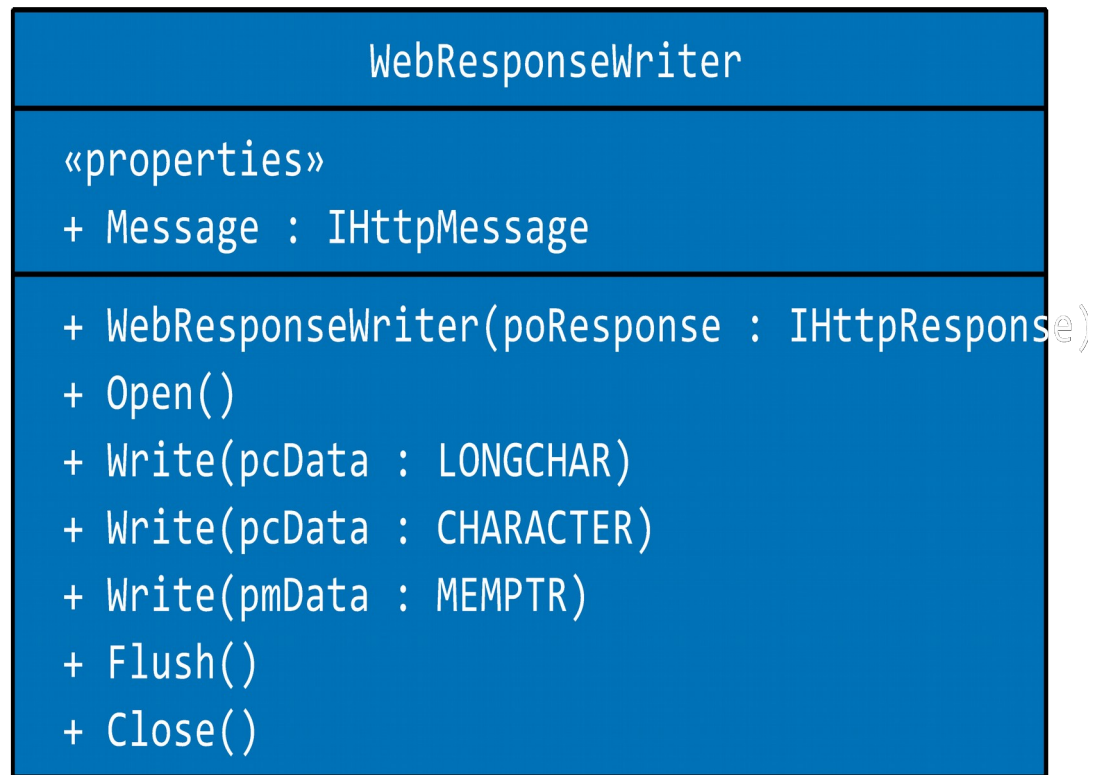
+ SetCookie(poCookie : Cookie)

Klasa OpenEdge.Web.WebResponseWriter

Zapisuje odpowiedź HTTP do strumienia Web Output Stream

Status/nagłówki zapisane przy pierwszym użyciu funkcji Write ()

Metody nagłówka i cookie



Klasa WebHandlera

Musi zawierać metody:

- `HandleNotAllowedMethod`
- `HandleNotImplemented`
- `HandleGet`
- Można dodać dalsze metody np: `HandlePost` itp.
- Plik `.cls` `..\[projekt]\PASOECContent\WEB-INF\openedge`

Przykład: HandleGet

```
METHOD OVERRIDE PROTECTED INTEGER HandleGet( INPUT poRequest AS  
OpenEdge.Web.IWebRequest ):
```

```
DEFINE VARIABLE oResponse AS OpenEdge.Net.HTTP.IHttpResponse NO-UNDO.  
DEFINE VARIABLE oWriter AS OpenEdge.Web.WebResponseWriter NO-UNDO.  
DEFINE VARIABLE oBody AS OpenEdge.Core.String NO-UNDO.
```

```
DEFINE VARIABLE jsonObj AS JsonObject NO-UNDO.  
DEFINE VARIABLE lcJSON AS LONGCHAR NO-UNDO.
```

ASSIGN

```
oResponse = NEW OpenEdge.Web.WebResponse()  
oResponse:StatusCode = INTEGER(StatusCodeEnum:OK).
```

```
jsonObj = NEW JsonObject().
```

```
IF ENTRY(2,poRequest:PathInfo,"/") = "Customer" THEN DO:  
DO:
```

```
DEFINE VARIABLE beCustomer AS customerBE NO-UNDO.  
DEFINE VARIABLE hTTCustomer AS HANDLE NO-UNDO.
```

```
beCustomer = NEW customerBE().  
beCustomer:ReadcustomerBE("",OUTPUT DATASET dsCustomer).  
hTTCustomer = TEMP-TABLE ttCustomer:HANDLE.  
hTTCustomer:WRITE-JSON ("JsonObject",jsonObj).  
lcJSON= jsonObj:GetJsonText().  
oBody = NEW OpenEdge.Core.String(lcJSON).
```

Przykład: HandleGet c.d.

.....

```
IF ENTRY(2,poRequest:PathInfo,"/") = "Orders" THEN
```

```
DO:
```

```
    DEFINE VARIABLE beOrder AS orderBE NO-UNDO.
```

```
    DEFINE VARIABLE htTOrder AS HANDLE NO-UNDO.
```

```
    beOrder = NEW orderBE().
```

```
    beOrder:ReadOrderBE("",OUTPUT DATASET dsOrder).
```

```
    htTOrder = TEMP-TABLE ttOrder:HANDLE.
```

```
    htTOrder:WRITE-JSON ("JsonObject",jsonObj).
```

```
    lcJSON= jsonObj:GetJsonText().
```

```
    oBody = NEW OpenEdge.Core.String(lcJSON).
```

.....

Konfiguracja WebHandlerów

Który handler jest użyty

handlerN=ooabl.typ.nazwa : <relatywny-uri>

- Relatywny w stosunku do /web

- RegEx 1* ["/" [token|text]]

- Musi być wiodący znak "/"

 - Następnie Text (np. Orders)

 - lub Token {np. OrderNum}

- Wzorce są dopasowywane w porządku numerycznym oraz wg najlepszego dopasowania

- Jeden handler może być użyty dla różnych ścieżek URI

- Każda WebApp ma wzorzec domyślny dla no-match-found

 - OenEdge.Web.Default lub CompatibilityHandler

Konfiguracja WebHandlerów

`\oepas1\conf\openedge.properties`

```
# Transport properties for the WEB protocol
[oepas1.webh2.WEB]
  adapterEnabled=1
  defaultCookieDomain=
  defaultCookiePath=
  defaultHandler=OpenEdge.Web.CompatibilityHandler
handler1=webh2Handler: /webh2
handler2=webh2Handler: /Customer
handler3=webh2Handler: /Orders
  srvrAppMode=development
  srvrDebug=0
  wsRoot=/webh2/static/webspeed
```




Więcej Informacji

- OpenEdge ABL API Reference:

<https://documentation.progress.com/output/oehttpclient/oe117/>

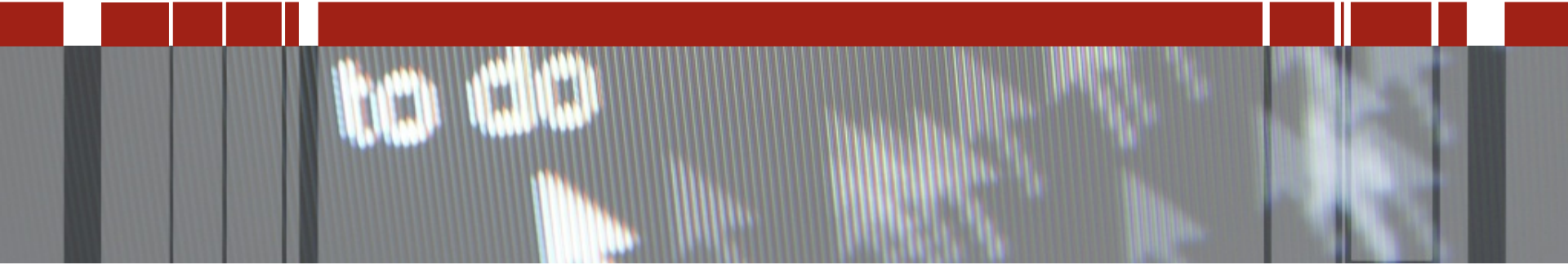
- Dokumentacja OpenEdge do najnowszej wersji

- PASOE Web Handlers – A Deep Dive – prezentacja Petera Judge'a

- Blog OpenEdge

<http://blog-progress.pl/pasoe-i-serwisy-webhandler-cz-i/>

<http://blog-progress.pl/pasoe-i-serwisy-webhandler-cz-ii/>



to do

Dziękujemy za uwagę