



PHP i5

#4

By SIRI

: webservices on i5





■ Agenda WS

- ✓ il protocollo soap
- ✓ struttura e interfaccia
- ✓ analisi di un ws da IDE Zend
- ✓ ws - esempi



Servizio all'utente

- Soap (**Simple Object Access Protocol**) è un protocollo leggero per lo scambio di informazioni in ambienti distribuiti

Tecnologia

- Dialogo per la comunicazione da server a server

Protocollo

- E' un protocollo basato sul flusso dati **XML**

Normativa

- Il consorzio **W3C** definisce le regole per la definizione della modalità di scambio informazioni all'interno dell'**XML**

Struttura

- La struttura XML consiste di tre parti

: Busta

- Definisce il contenuto del messaggio

: Regole

- L'insieme di regole per decodificare il messaggio e come processarlo

: Diaologo

- La convenzione delle chiamate e delle risposte

L'interfaccia di dialogo con i webservices è il WSDL

Web Service Description language

- Lo scopo di un **WSDL** è principalmente quello di definire l'interfaccia con tutti i servizi offerti dall'infrastruttura
- Il **WSDL** deve essere **Pubblico**
- Il **WSDL** è indipendente dalla tecnologia utilizzata per eseguire lo scambio di informazioni

SOAP: contenuto interfaccia

Questo particolare XML (WSDL) contiene tutte le informazioni circa il **web Service**

Contenuto

- Dove il servizio andrà a reperire le informazioni (**where**)
- Quali metodi verranno gestiti
- Tipo di dato (**data type**)
- Valori di ritorno (**return value**)

SOAP: interfaccia wsdl

```
<wsdl:definitions targetNamespace="https://adwords.google.com/api/adwords/v12">
  <wsdl:types>
    <schema elementFormDefault="qualified"
targetNamespace="https://adwords.google.com/api/adwords/v12">
      <complexType name="AdSchedule">
        <sequence>
          <element name="intervals" maxOccurs="unbounded"
type="impl:SchedulingInterval"/>
          <element name="status" type="impl:AdScheduleStatus"/>
        </sequence>
      </complexType>
      <simpleType name="AdScheduleStatus">
        <restriction base="xsd:string">
          <enumeration value="Disabled"/>
          <enumeration value="Enabled"/>
        </restriction>
      </simpleType>
      .....
      .....
```

where

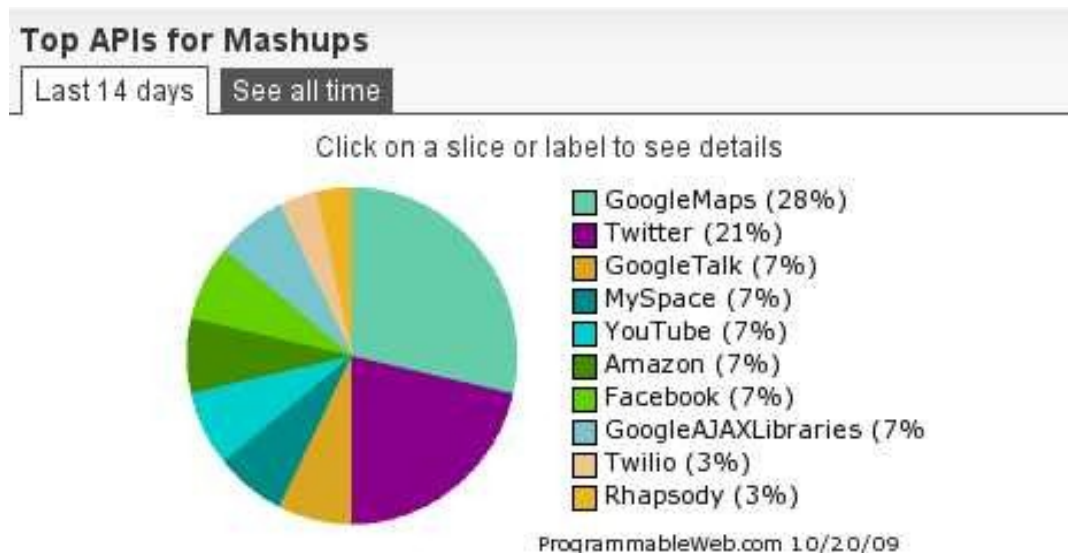
Data Type

Return value

SOAP: interfaccia - riferimenti

Riferimenti utili

- <http://www.programmableweb.com/>
- Panoramica complessiva sulla distribuzione dei webServices

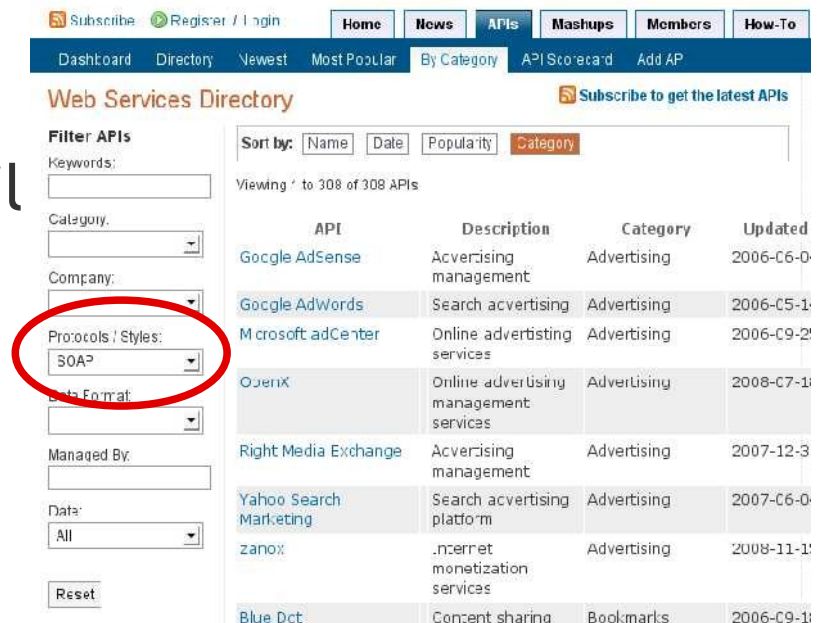


SOAP: interfaccia - riferimenti

Riferimenti utili

<http://www.programmableweb.com/apis/directory/1?protocol=SOAP>

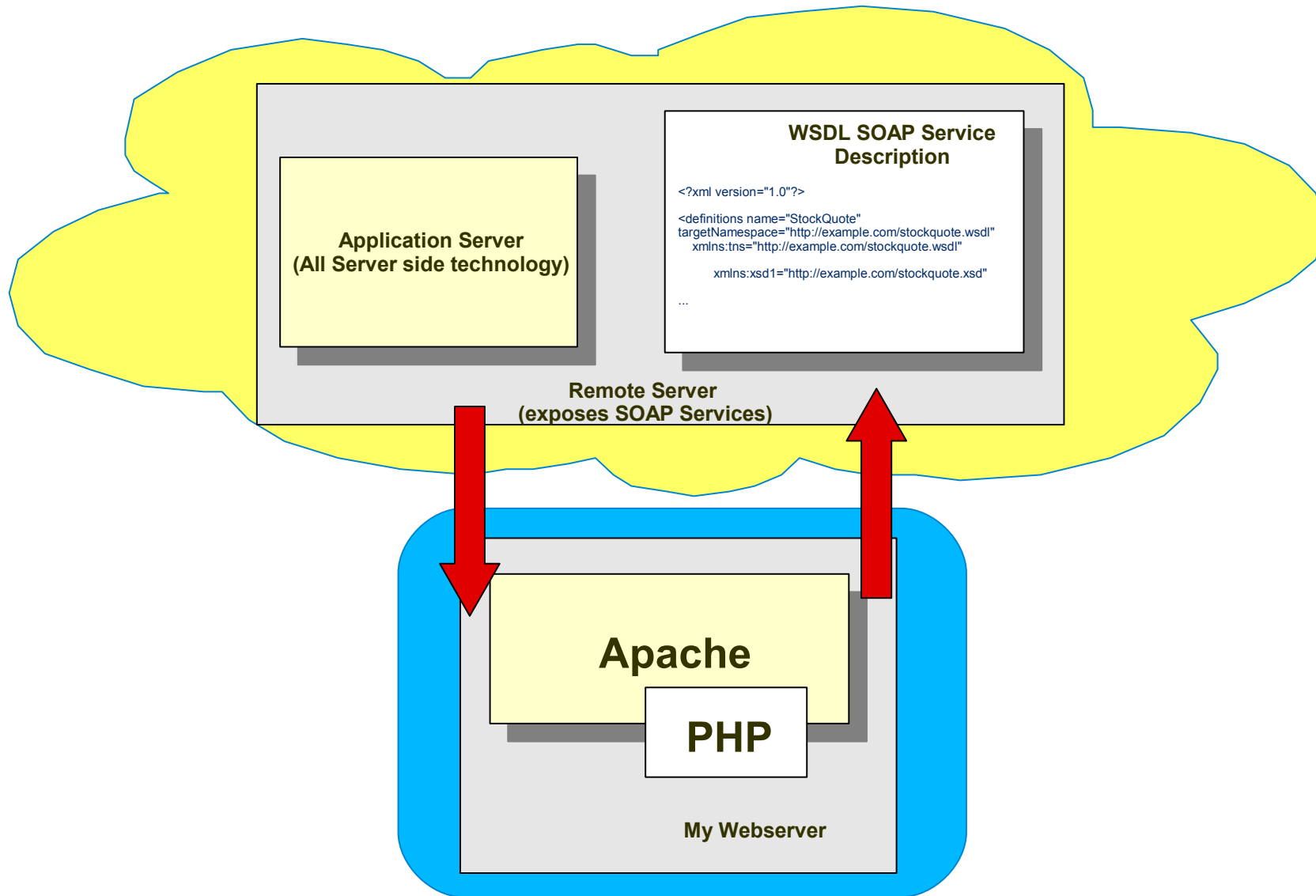
- Elenca più di 300 API relative al protocollo SOAP utilizzato
- Il “webservice” è la porta per il colloquio di software scritti in differenti linguaggi di programmazione per lo scambio di dati e di informazioni



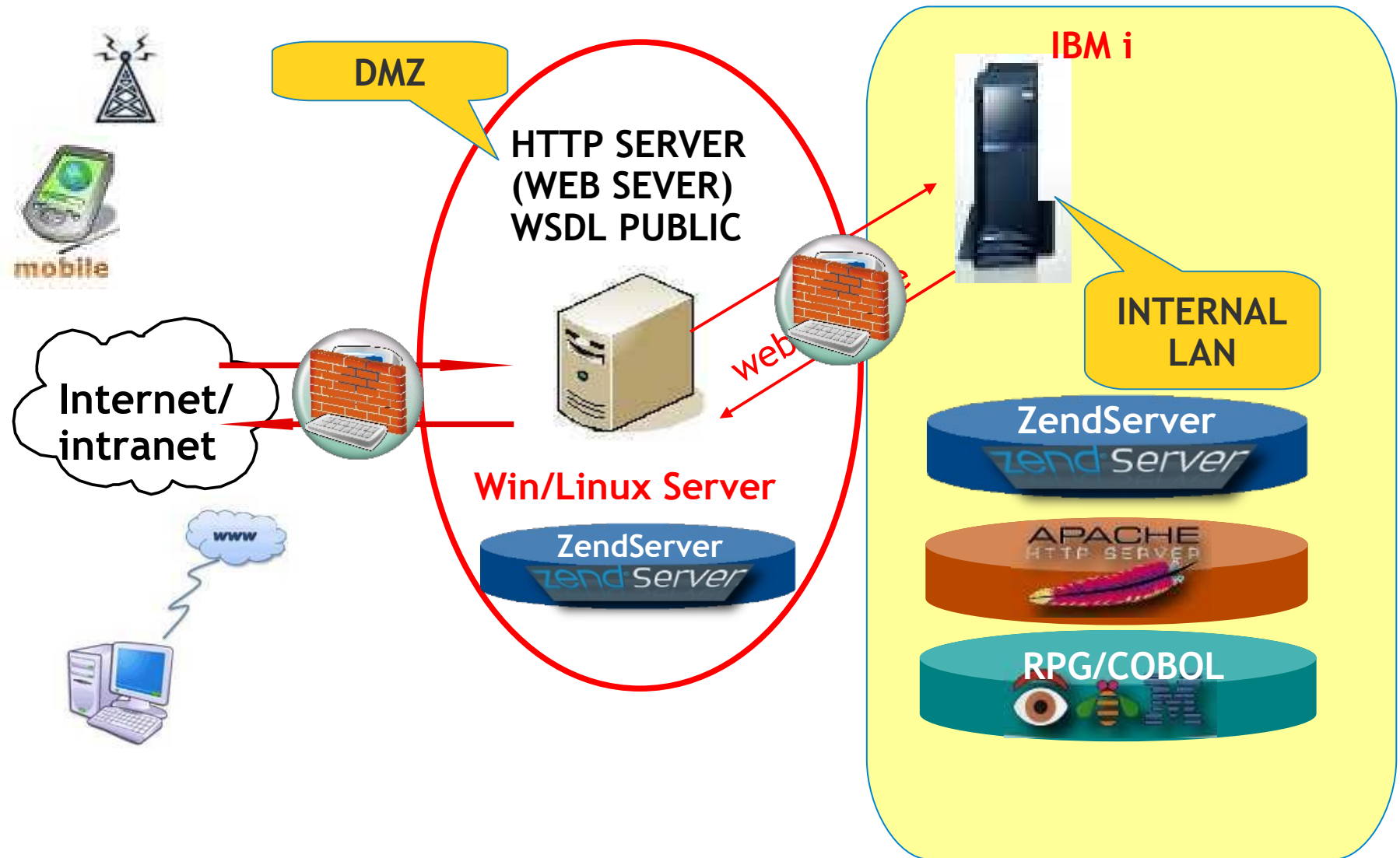
The screenshot shows the ProgrammableWeb API Directory interface. The 'Filter APIs' section on the left has 'SOAP' selected in the 'Protocols / Styles' dropdown menu, which is circled in red. The main content area displays a table of APIs with columns for API name, Description, Category, and Updated date. The table lists several APIs, including Google AdSense, Google AdWords, Microsoft adCenter, OpenX, Right Media Exchange, Yahoo Search Marketing, zanoX, and Blue Dot.

API	Description	Category	Updated
Google AdSense	Advertising management	Advertising	2006-06-0
Google AdWords	Search advertising	Advertising	2006-05-1
Microsoft adCenter	Online advertising services	Advertising	2006-09-2
OpenX	Online advertising management services	Advertising	2008-07-1
Right Media Exchange	Advertising management	Advertising	2007-12-3
Yahoo Search Marketing	Search advertising platform	Advertising	2007-06-0
zanoX	Internet monetization services	Advertising	2008-11-1
Blue Dot	Content sharing	Bookmarks	2006-09-1

SOAP: interfaccia con apache & php



SOAP: interfaccia in IBMi con apache & php





- SOAP

- ✓ Estensioni necessarie
- ✓ Uso delle classi



SOAP: prerequisiti

Zend Core™ for i5/OS
Control Center Configuration Documentation About Logout
System Overview | PHPinfo | Benchmark | Support | Updates

phpinfo

PHP Version 5.2.9 

System	OS400 ZEND826 4 5 0010000BE27C
Build Date	Mar 22 2009 18:43:53
Configure Command	'./configure' '--prefix=/usr/local/Zend/Core' '--with-config-file-path=/etc' '--enable-force-cgi-redirect' '--enable-fastcgi' '--disable-debug' '--enable-inline-optimization' '--enable-memory-limit' '--disable-all' '--enable-ctype' '--enable-dom' '--enable-libxml' '--with-libxml-dir=/usr/local/Zend/Core' '--with-openssl=/usr' '--with-pcre-regex' '--enable-session' '--enable-simplexml' '--enable-spl' '--enable-wddx' '--enable-xml' '--with-zlib=/usr' '--with-pear' '--with-apxs2=/usr/local/Zend/apache2/bin/apxs' '--with-layout=GNU' '--enable-zmail' '--enable-json' '--enable-filter' '--enable-hash' '--enable-reflection'
	Apache 2.0 Handler

Deve essere abilitata l'estensione SOAP nella configurazione del PHP

soap

Soap Client	enabled
Soap Server	enabled

Directive	Local Value	Master Value
soap.wsdl_cache	1	1
soap.wsdl_cache_dir	/tmp	/tmp
soap.wsdl_cache_enabled	1	1
soap.wsdl_cache_limit	5	5
soap.wsdl_cache_ttl	86400	86400

SOAP: uso delle classi

Esempio di istanza di classe del servizio
Server

```
?php
class WSMethods {
    function myMethod($param) {
        return $param;
    }
}

$soap = new SoapServer ( 'file.wsdl' );
$soap->setClass ( 'WSMethods' );
```

SOAP: uso delle classi

Esempio di istanza di classe del servizio
Client

```
<?php

// Instantiate (using WSDL)
$soap = new SoapClient ( 'file.wsdl' );

//Then call methods
$soap->myMethod ( 'Hello!' );
// or
$soap->__soapCall ( 'myMethod', array ( 'Hello!' ) );
```

Handling - Gestione dei possibili errori

```
<?php

//Using exceptions
try {
    $soap = new SoapClient ( 'file.wsdl' );
    $res = $soap->myOtherMethod ( 'xyz' );
} catch ( SoapFault $ex ) {
}

//Using a function
if ( is_soap_fault ( $res ) ) {
}
```



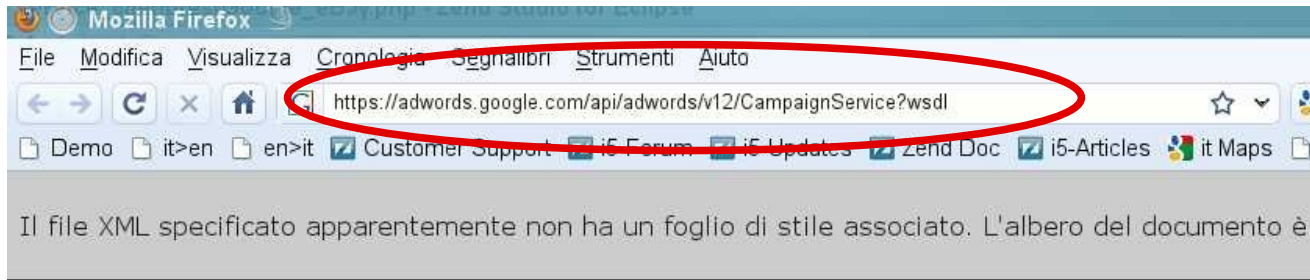

- **Analisi di WSDL**

- ✓ componenti di un wsdl



WSDL: analisi

un “WSDL” **pubblico** visto dal browser



```
-<wsdl:definitions targetNamespace="https://adwords.google.com/api/adwords/v12">
  -<wsdl:types>
    -<schema elementFormDefault="qualified" targetNamespace="https://adwords.google.com/a
      -<complexType name="AdSchedule">
        -<sequence>
          <element name="intervals" maxOccurs="unbounded" type="impl:SchedulingInterval"/>
          <element name="status" type="impl:AdScheduleStatus"/>
        </sequence>
      </complexType>
    -<simpleType name="AdScheduleStatus">
      -<restriction base="xsd:string">
        <enumeration value="Disabled"/>
        <enumeration value="Enabled"/>
      </restriction>
    </simpleType>
```

Struttura del WSDL
da analizzare

WSDL: analisi

un "WSDL" **pubblico** visto dall'IDE ZendStudio

The screenshot shows the Zend Studio IDE interface. The main editor displays a PHP script that creates a SoapClient instance for the URL `https://adwords.google.com/api/adwords/v12/CampaignService?wsdl`. The IDE's Outline view on the left shows the project structure, with the Soap Clients section expanded to show the `CampaignService?wsdl` client and its methods. A red circle highlights this section. The right-hand pane shows the details of the selected method, including its location, class, and return type. A yellow callout points to the URL in the code, and another yellow callout points to the method details pane. A third yellow callout points to the Outline view.

Instance dell'url WSDL

Tutte le informazioni dell'oggetto SOAP disponibili

Analisi metodi WSDL remoto



- Wsdl - esempi

✓ lato server



SOAP IBMi – lato server

```
....ILERPG
C  *entry  plist
C          parm          code          10
C          parm          name          10

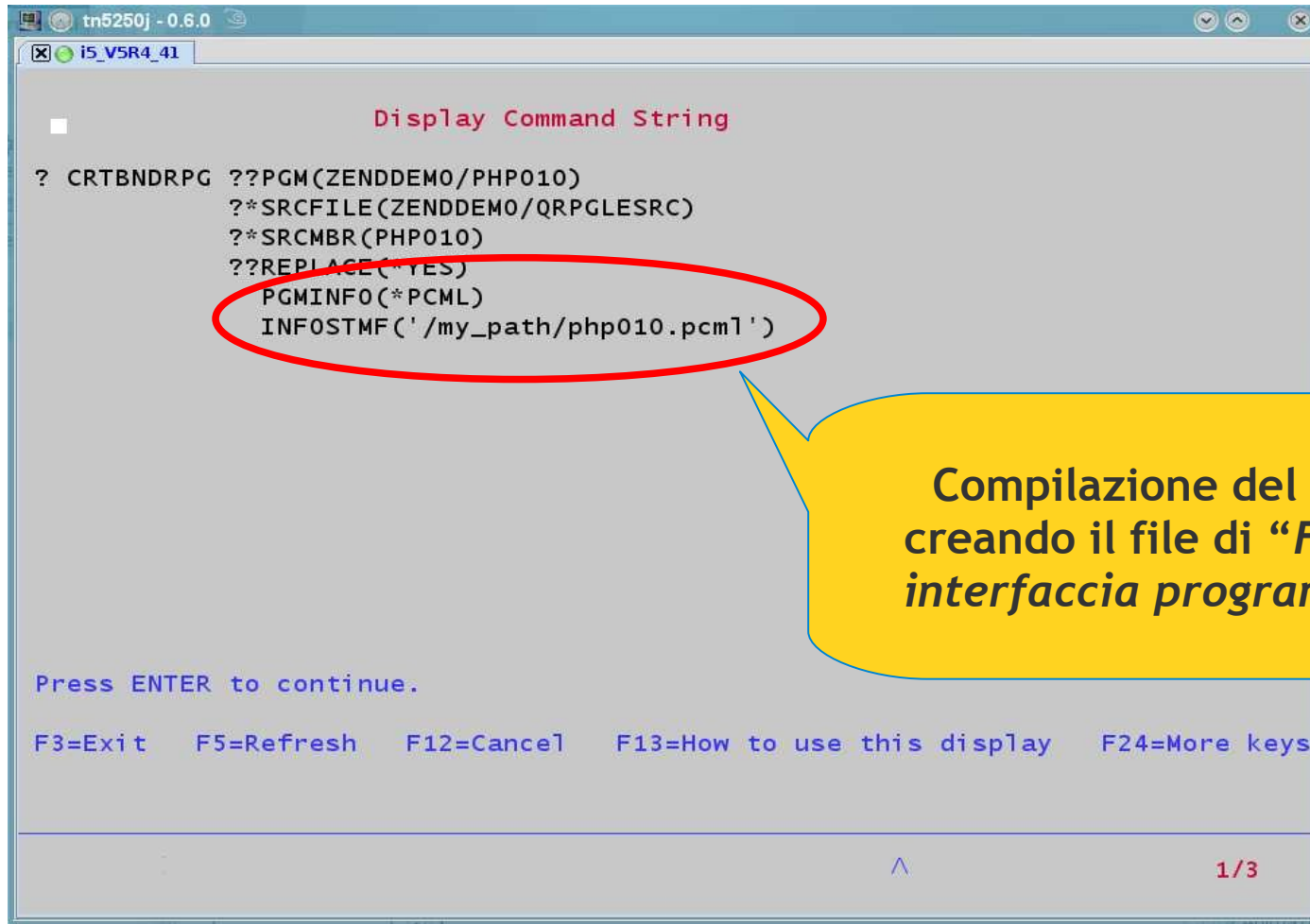
c          select
c          when code='1'
c          eval          name='IBM'
c          when code='2'
c          eval          name='Siri'
c          other
c          eval          name='no code'
c          endsl

c          eval          *inlr=*on
c          return

/eject
```

Definizione programma
RPG

SOAP IBMi – lato server



```
tn5250j - 0.6.0
I5_V5R4_41

Display Command String

? CRTBNDRPG ??PGM(ZENDDemo/PHP010)
  ??SRCFILE(ZENDDemo/QRPGLESRC)
  ??SRCMBR(PHP010)
  ??REPLACE(*YES)
  PGMINFO(*PCML)
  INFOSTMF('/my_path/php010.pcm1')

Press ENTER to continue.

F3=Exit  F5=Refresh  F12=Cancel  F13=How to use this display  F24=More keys

^ 1/3
```

Compilazione del pgm
creando il file di “*Flusso
interfaccia programma*”

SOAP IBMi – lato server

```
<pcml version="4.0">
  <!-- RPG program: PHP010 -->
  <!-- created: 2009-07-16-15.40.50 -->
  <!-- source: ZENDDEMO/QRPGLESRC(PHP010) -->
  <program name="PHP010" path="/QSYS.LIB/ZENDDEMO.LIB/PHP010.PGM">
    <data name="CODE" type="char" length="1" usage="inputoutput" />
    <data name="NAME" type="char" length="10"usage="inputoutput" />
  </program>
</pcml>
```

L'XML risultante dalla
compilazione del pgm RPG

SOAP IBMi – lato server

- Script **PHP** che prende in carico le richieste **SOAP**

```
<?php
class program_service {}...

ini_set ( "soap.wsdl_cache_enabled", "0" ); // disabling WSDL cache

/* istanzia variabile di classe SoapServer */
$server = new SoapServer ( "soap_service.wsdl" );
//Imposta la classe che gestira' le richieste SOAP.
$server->setClass ( "program_service" );
// Processa le richiesta SOAP
$server->handle ();
```

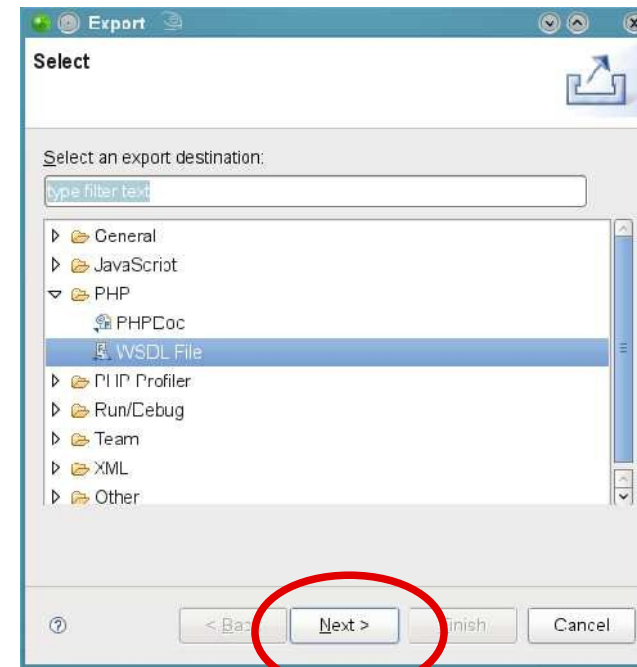
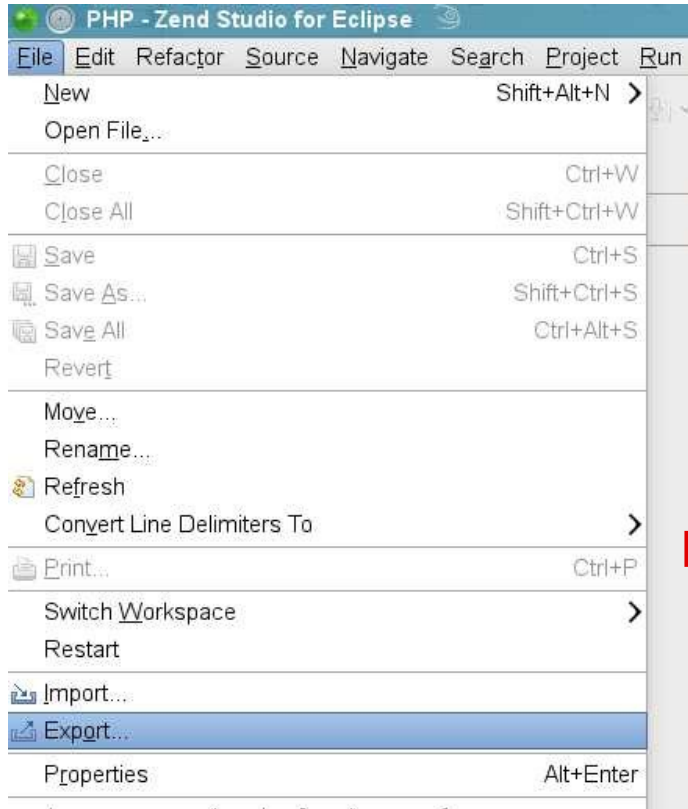

SOAP IBMi – lato server

- Dettaglio della classe che gestisce il servizio

```
public function get_type($code)
{
    require_once('i5_connect_bkg.php');
    $description=file_get_contents("php010.pcml");
    /* Apertura programma con parametri */
    $pgm = i5_program_prepare_PCML($description);
    /* Impostazione parametri di input output*/
    $parmln = array(
        "CODE"=>$code,
    );
    $parmOut = array(
        "NAME"=>"name_out",
    );
    /* Richiamo programma */
    $ret = i5_program_call($pgm, $parmln, $parmOut);
    // return
    return $name_out;
}
```

SOAP IBMi – lato server

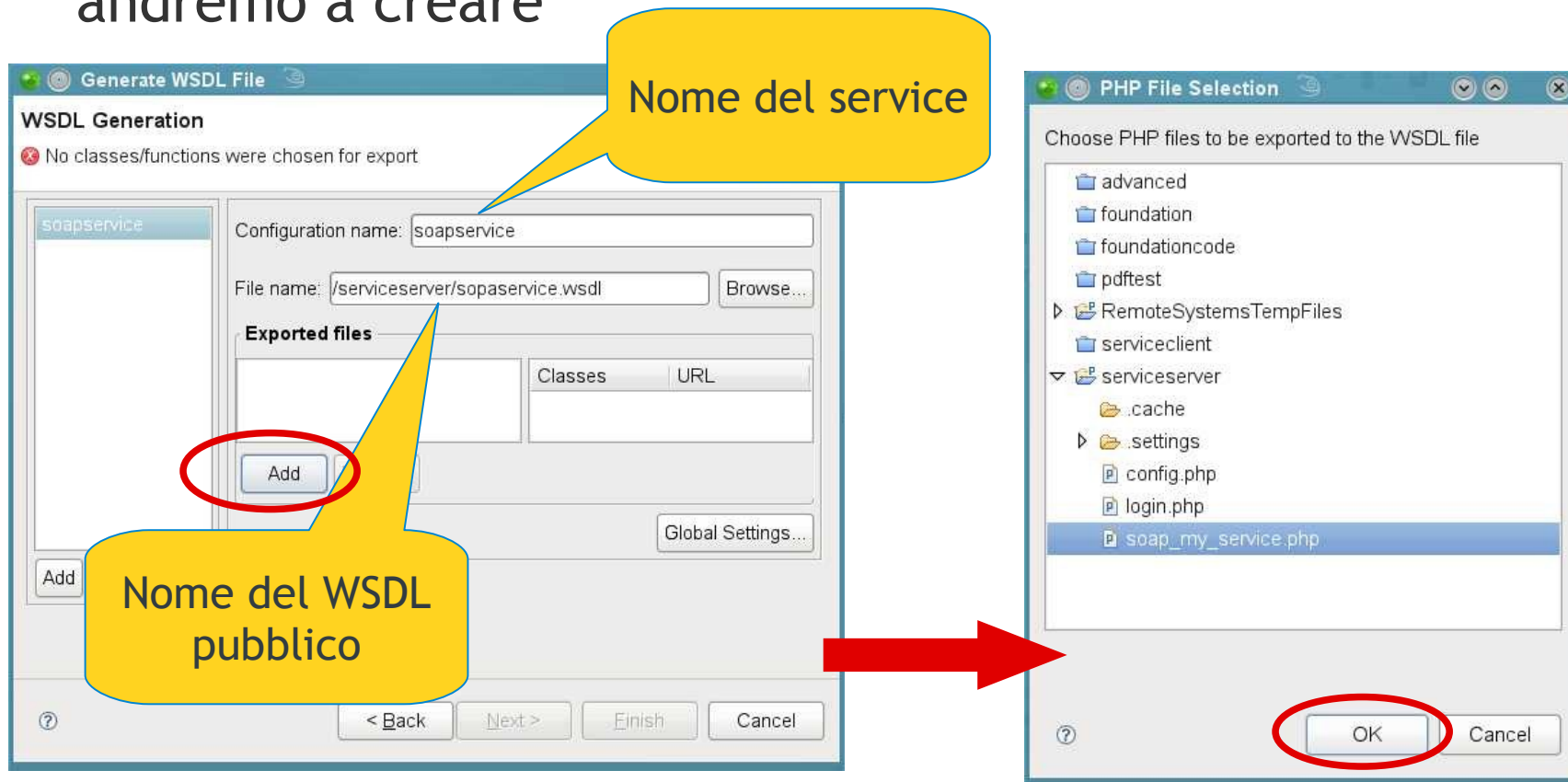
- Generazione del WSDL



file->Export->php->wsdl file

SOAP IBMi – lato server

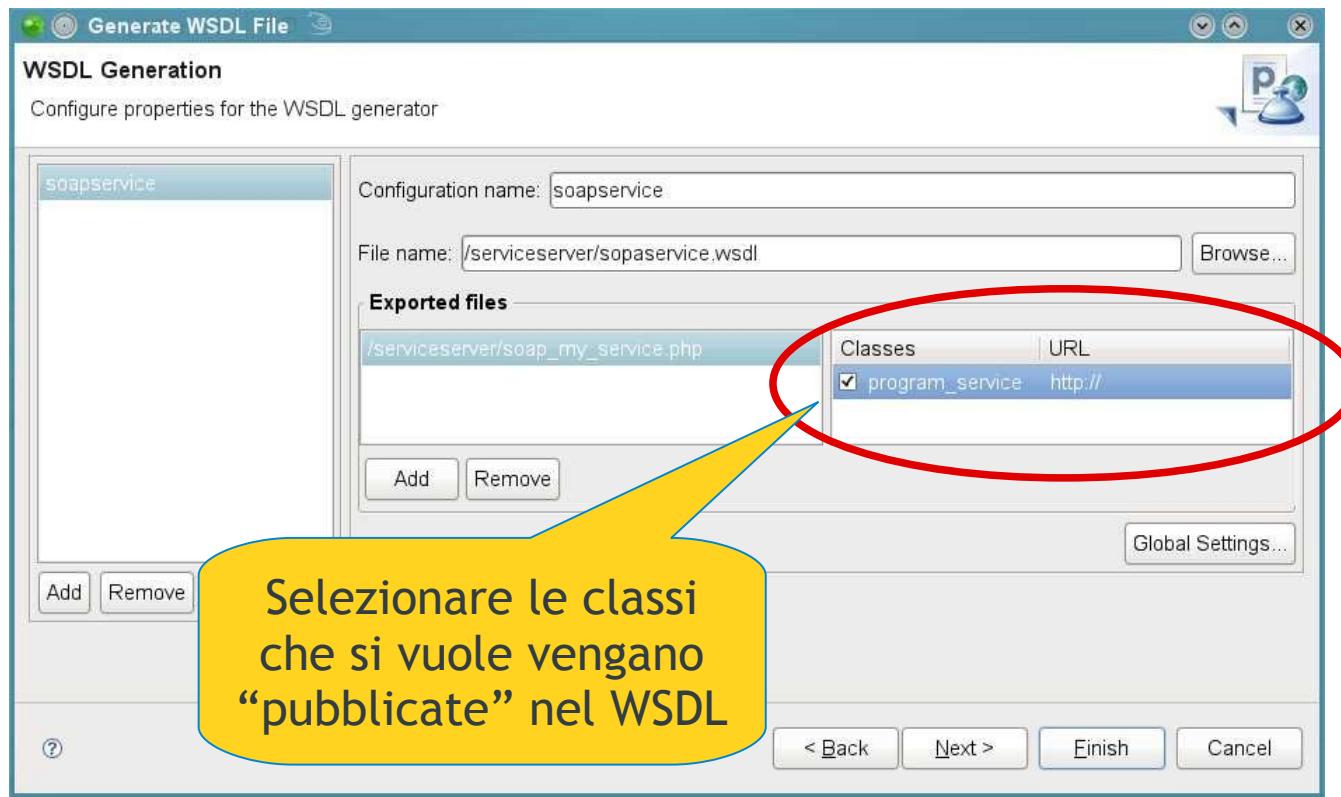
- Immettere il nome del webservice e della pagina WSDL che andremo a creare



- aggiungere la/le pagine **PHP** che faranno parte del **WSDL**

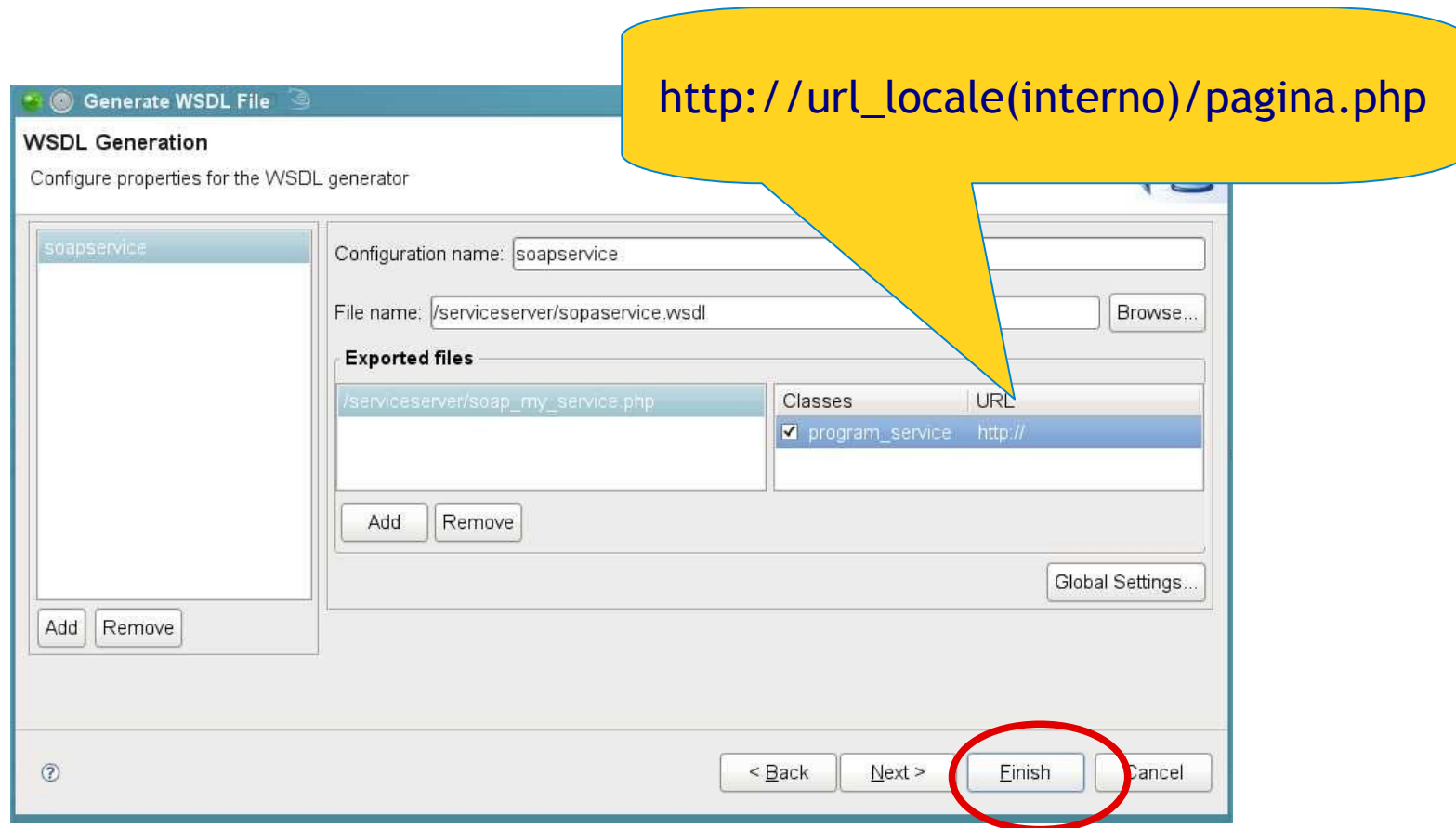
SOAP IBMi – lato server

- **ZendStudio** individua in modo autonomo la/le classi che fanno parte dello script **PHP** e chiede di selezionare quali faranno parte del **webservice**



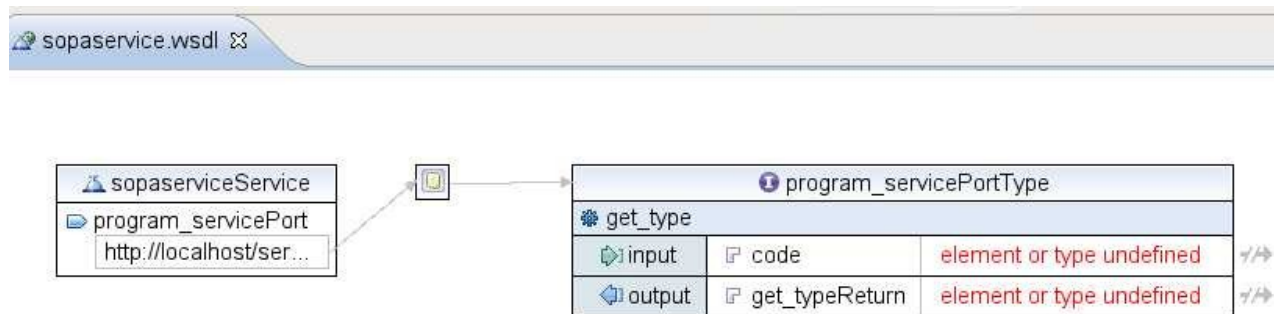
SOAP IBMi – lato server

- va indicato “manualmente” l'URL (interno) a cui il WSDL deve appoggiarsi per attivare il servizio



SOAP IBMi – lato server

- al termine viene creato il file **WSDL** e **ZendStudio** provvede a visualizzare in modo grafico il tipo di servizio
- Il grafico può essere utilizzato per generare della documentazione



SOAP IBMi – lato server

- Il file **WSDL** può anche essere editato per le ulteriori verifiche

```
<?xml version='1.0' encoding='UTF-8'?>
<!-- WSDL file generated by Zend Studio. -->

<definitions name="soap_service" targetNamespace="urn:soap_service" xmlns:typens="urn:soap_service"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="get_type">
    <part name="code" type="xsd:string"/>
  </message>
  <message name="get_typeResponse">
    <part name="get_typeReturn" type="xsd:string"/>
  </message>
  <portType name="i5_program_servicePortType">
    <operation name="get_type">
      <documentation>
        Soap service example reperimento valore da ILE-RPG SYTEM-I
      </documentation>
      <input message="typens:get_type"/>
      <output message="typens:get_typeResponse"/>
    </operation>
  </portType>
  <binding name="i5_program_serviceBinding" type="typens:i5_program_servicePortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="get_type">
      <soap:operation soapAction="urn:i5_program_serviceAction"/>
    </operation>
  </binding>
</definitions>
```



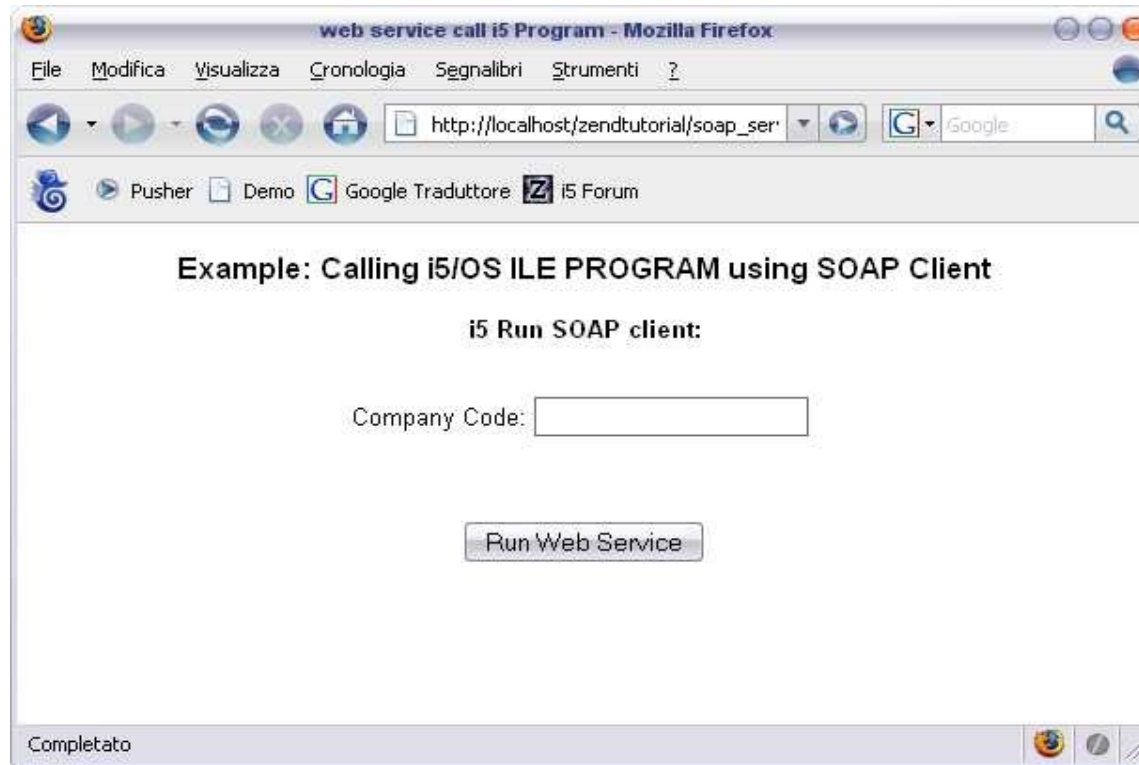
- Wsdl - esempi

✓ lato client



SOAP IBMi – lato client

- impostazione del lato client: creazione form **WEB**



SOAP IBMi – lato client

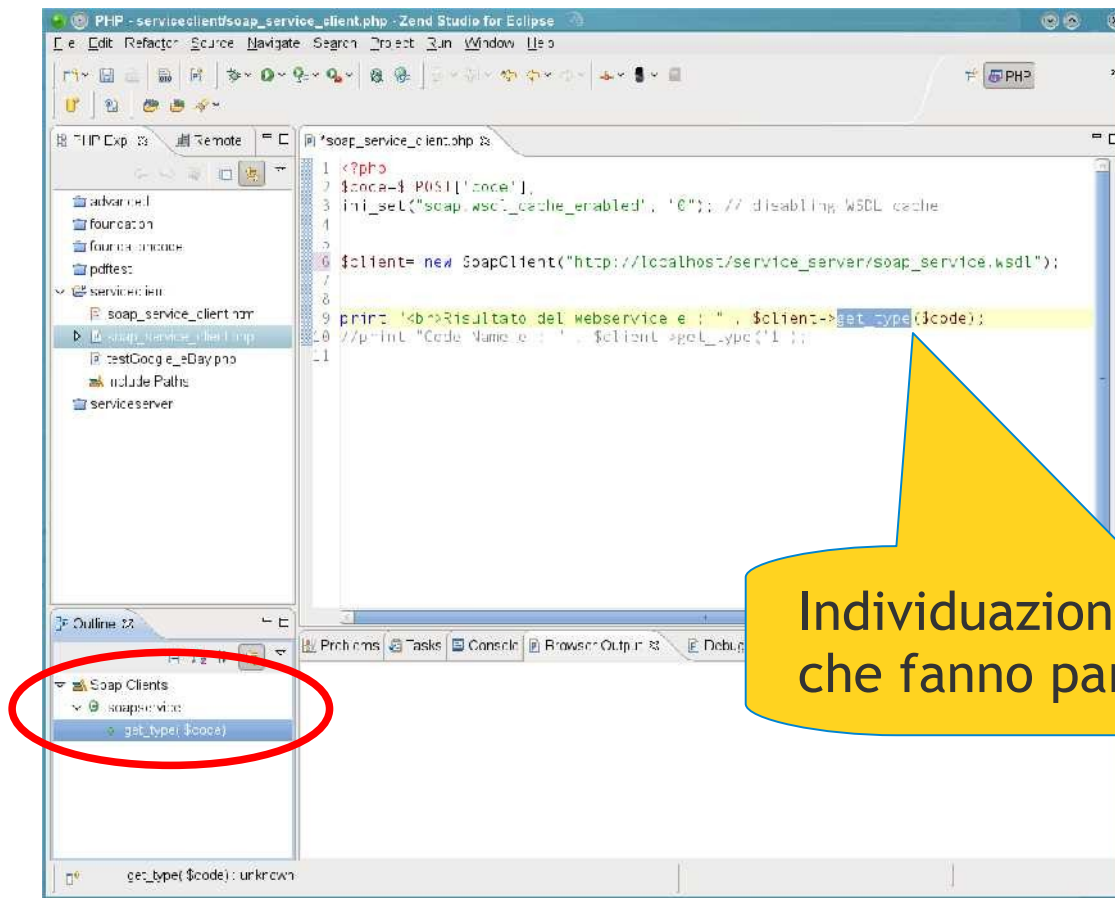
- impostazione del lato client: esecuzione del PHP

```
1 <html>
2 <head>
3 <title>web service call i5 Program</title>
4 <link rel="stylesheet" type="text/css" href="php.css">
5 </head>
6
7 <body>
8 <center>
9 <h3>Example: Calling i5/OS ILE PROGRAM using SOAP Client</h3>
10
11
12
13 <p><b>i5 Run SOAP client:</b></p>
14
15 <form action="soap_service_client.php" method="POST">
16
17 <br>
18 Company Code: <input type="text" name="code" /> <br>
19 <br>
20 <br><br>
21 <input type="submit" value="Run Web Service" />
22 </form>
23 </body>
24 </html>
```

predisposizione del php per richiamo del web-service

SOAP IBMi – lato client

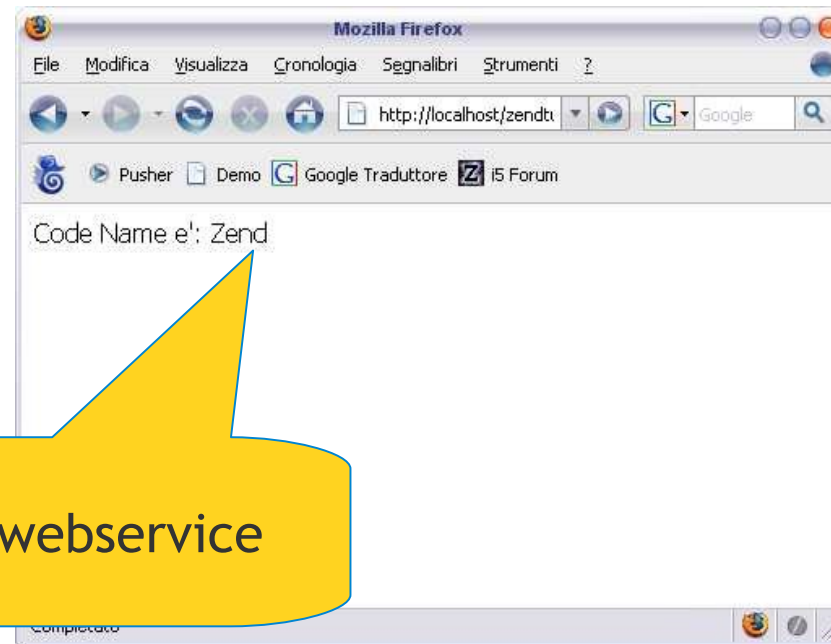
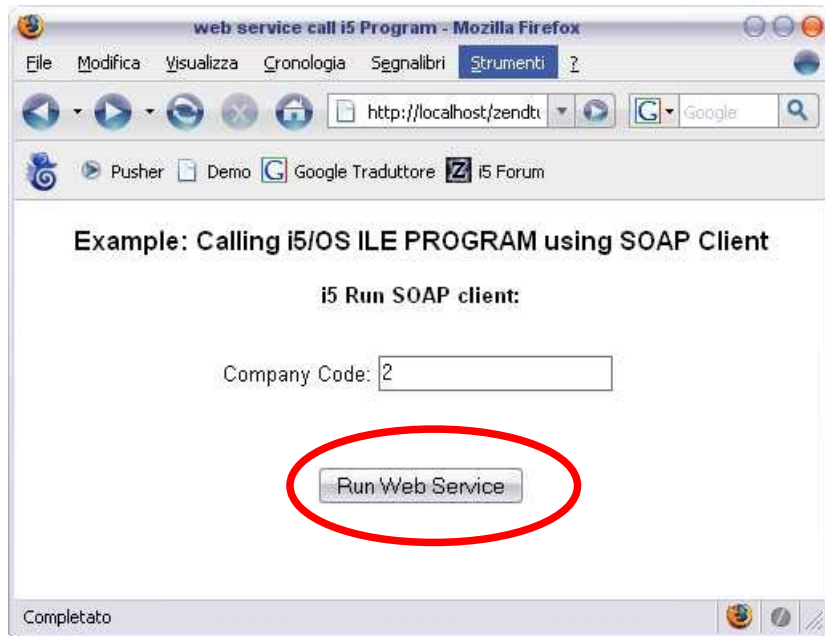
- Individuazione del servizio offerto dal server remoto tramite l'IDE ZendStudio



Individuazione della/le classi che fanno parte del webservice

SOAP IBMi – lato client

- test sul risultato finale



Risultato del webservice



QUESTION TIME ?



Nome _____

Cognome _____

Data _____





ARRIVEDERCI



TITOLO