
SOAPované distribuované systémy

Róbert Novotný

(UINF/KOPR, 27. november 2014)

20. storočie v Remote Procedure Calls

- milión rozličných protokolov
- binárne + proprietárne
 - RMI [Java]
 - DCOM [Microsoft]
 - CORBA [OMG]

RPC



Čo to je SOAP?

- *„SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment.“*
 - *Simple Object Access Protocol 1.1*

SOAP == webservisy

ľudová
slovesnosť

SOAP

envelope

- obsah správy
- adresáti
- povinnosť

encoding rules

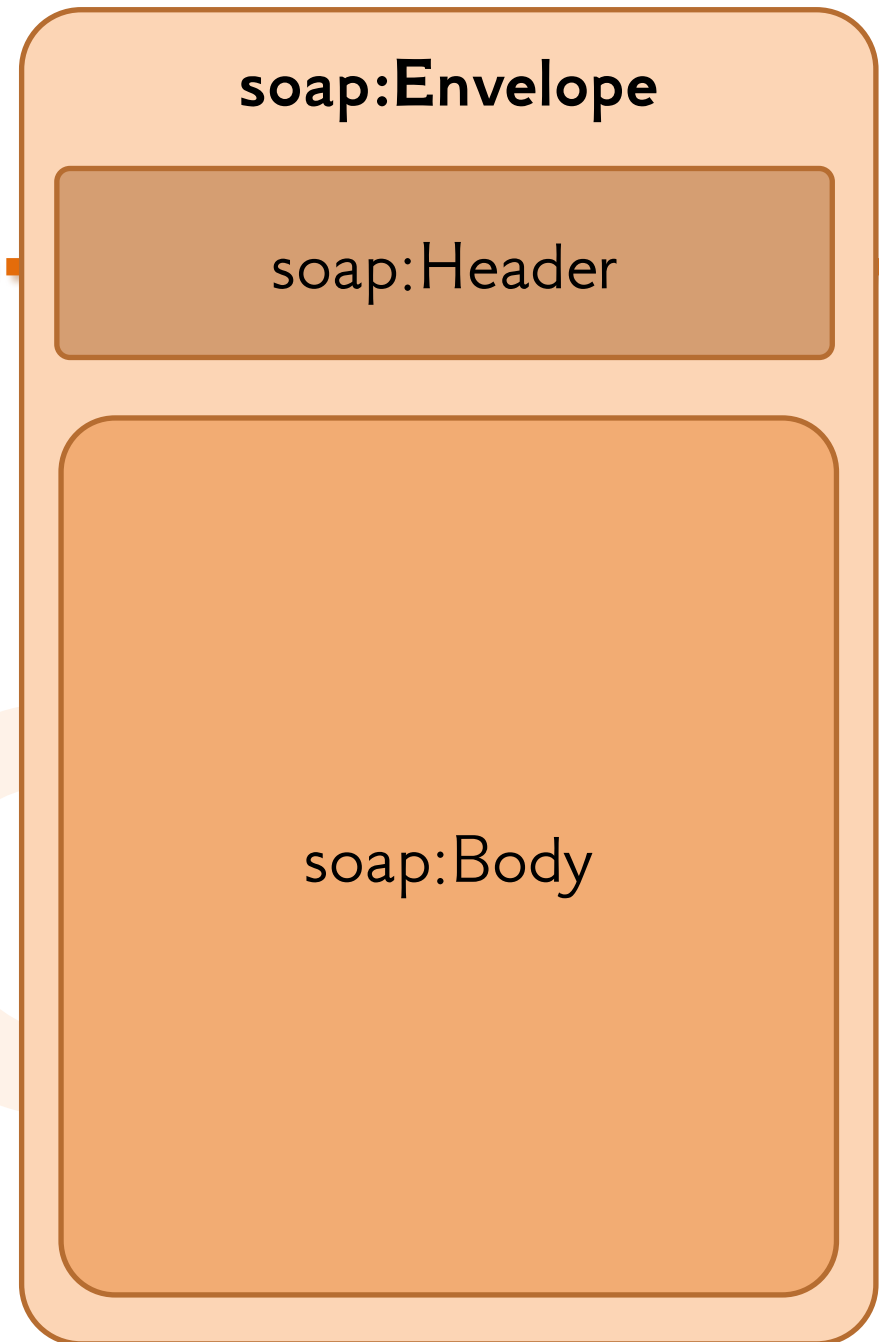
- serializácia objektov
- formát dát v obsahu

RPC representation

- URI cieľa
- špecifikácia vzdialenej metódy

SOAP

- správa je vždy XML
- ľubovoľný transport
 - binding
 - default: HTTP



POST http://www.websvcicex.net/stockquote.asmx
SOAPAction: "http://www.webserviceX.NET/GetQuote"
Host: www.websvcicex.net
Content-Type: text/xml;charset=UTF-8

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <GetQuote xmlns="http://www.webserviceX.NET/">
      <symbol>AAPL</symbol>
    </GetQuote>
  </soap:Body>
</soap:Envelope>
```

Request

HTTP/1.1 200 OK

Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

Response

<soap:Envelope

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

<soap:Body>

<GetQuoteResult xmlns="http://www.webserviceX.NET/">

<Price>34.5</Price>

</GetQuoteResult>

</soap:Body>

</soap:Envelope>

Ako postaviť endpoint?

1. vezmem HTTP server
2. prijmem XML v požiadavke + naparsujem
3. spočítam výsledok
4. zlepím XML, pošlem do odpovede
5. ???

6. PROFIT!

Instantný SOAP service v Jave

- použijem JAX-WS 2.0
- priama podpora v JDK6!
- jedna trieda
- jeden spúšťač!

SOAP

Vytvorte "Hello World" endpoint v
JAX-WS 2.0

Použite **soapUI** / **curl** /
HttpRequester ako klienta.

JAX-WS: webservises v Java

- **operácie** = metódy obslužných tried
 - metódy ozdobíme @WebMethod / @WebParam
 - triedy ozdobíme @WebService
- **dáta** = triedy pre dáta
 - voliteľne ozdobíme JAXB anotáciami
 - serializácia medzi XML a objektami
- spustíme v SOAP serveri

@WebService

```
public class ChocolateService {  
    private List<Chocolate> chocolates  
        = new CopyOnWriteArrayList<Chocolate>();  
    public ChocolateService() {  
        chocolates.add(new Chocolate("Lindt Excellence 70%", 70));  
    }  
  
    public List<Chocolate> list() {  
        return chocolates;  
    }  
  
    public void add(Chocolate chocolate) {  
        chocolates.add(chocolate);  
    }  
}
```

všetky *public*
metódy sa
zverejnia

```
Endpoint.publish("http://localhost:10000/ws/chocolates",  
    new ChocolateService());
```

zverejní SOAP
službu cez
HTTP

pristupujeme
cez POST

navštívme
adresu v
browseri

DEMO

[JAX-WS 2.0]

SOAP



Ako sa pripojiť?

- priamo posielat' správy cez HTTP
 - odkiaľ poznáme tvar správ?
 - odkiaľ zoznam poskytovaných metód?
 -
- potrebujeme dokumentáciu!



WSDL

Web
Services
Description
Language

- metadátoový popis webovej služby
 - operácie?
 - dátové typy?
 - štýly?
 - bindings?
- XML formát pre čítanie strojom
- z neho možno generovať **klientov!**

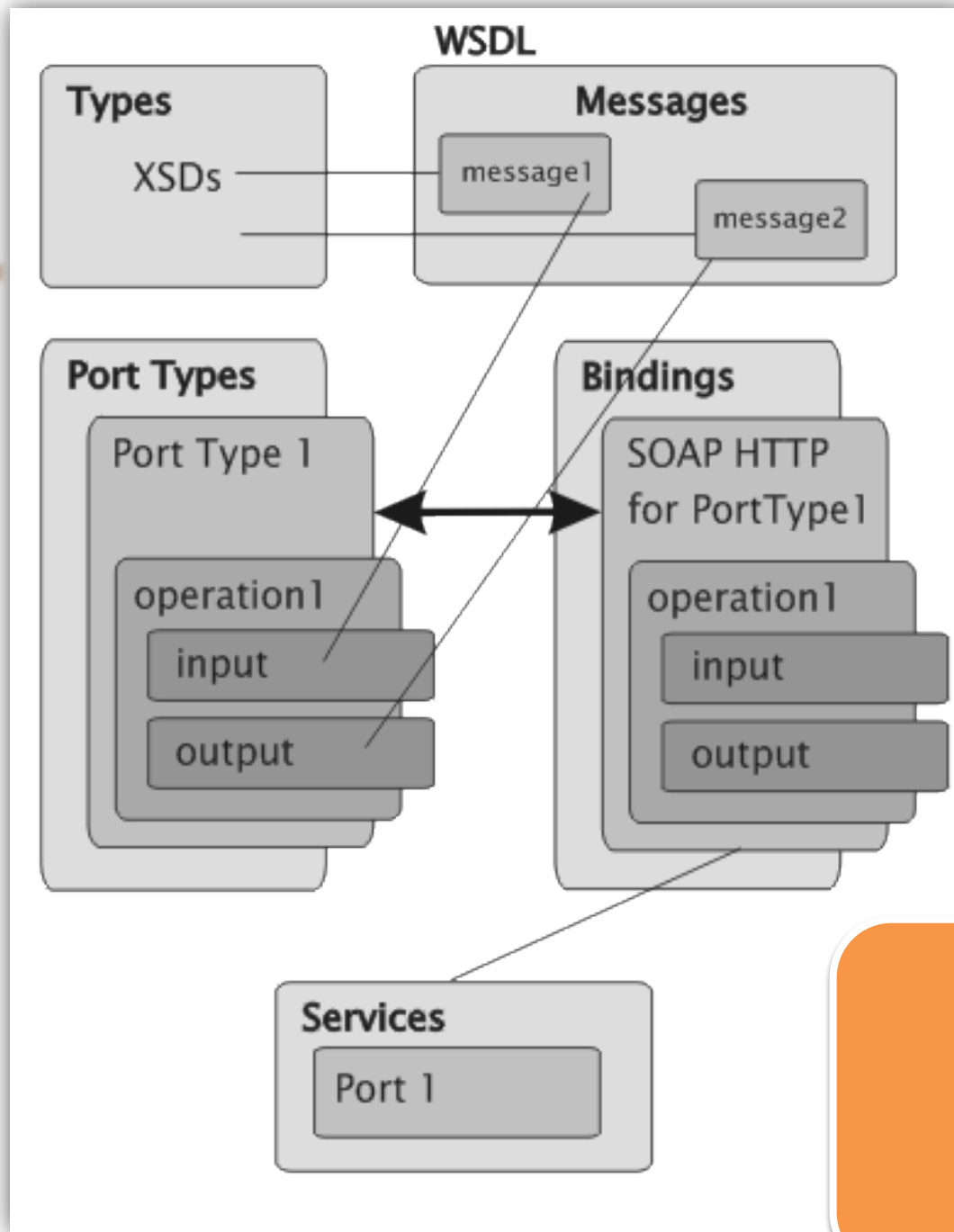


WSDL: automatické generovanie

- JAX-WS 2.0 server automaticky vygeneruje WSDL

<http://localhost:10000/ws/chocolates?wsdl>

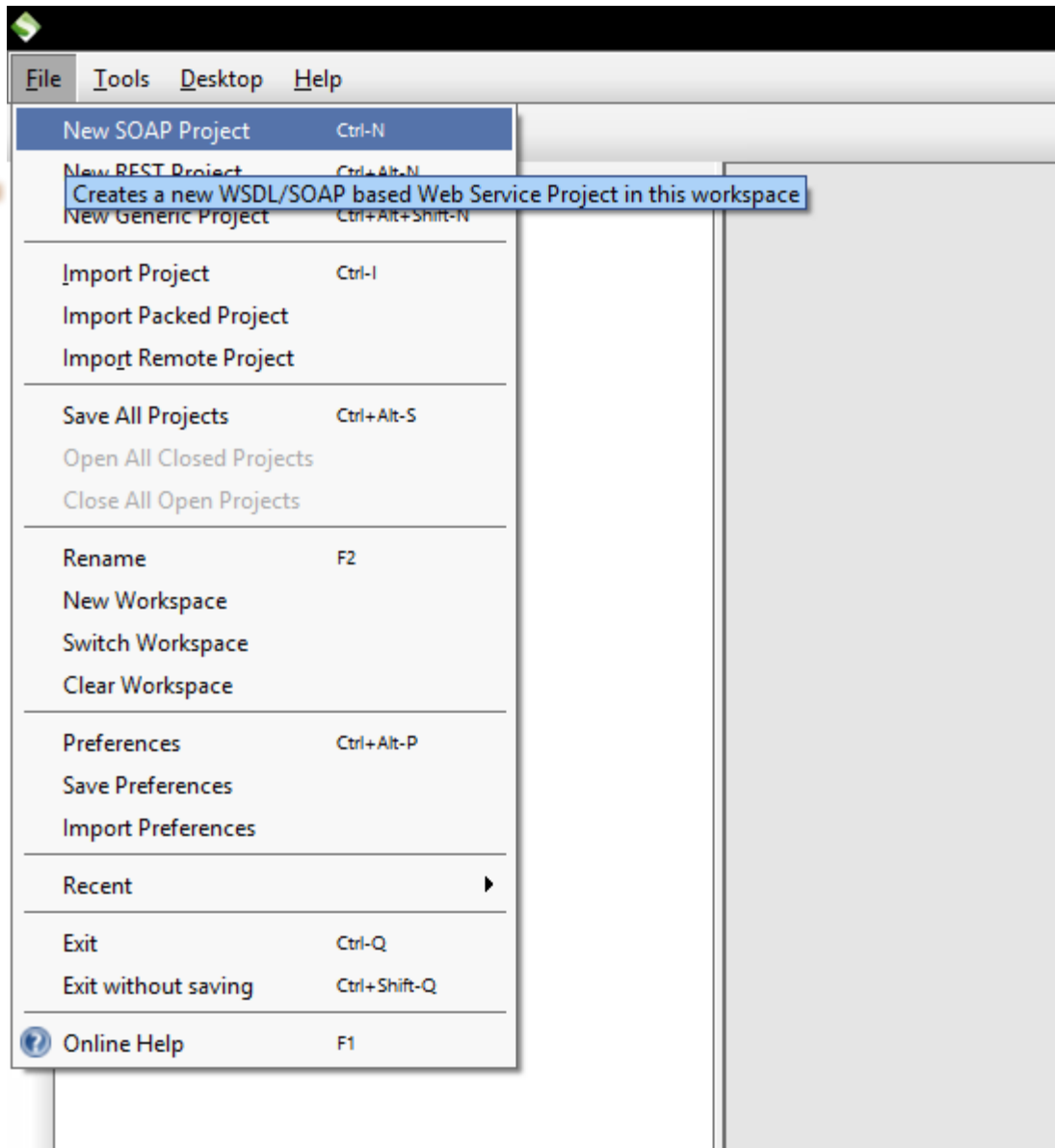
- určené pre stroje, nie pre ľudí!
- ale neskôr budeme vytvárať ručne



štruktúra
WSDL



Vytvorte klienta pre službu v JAX-
WS 2.0 pomocou SoapUi



File Tools Desktop Help

New SOAP Project Ctrl-N

New REST Project Ctrl+Alt-N

Creates a new WSDL/SOAP based Web Service Project in this workspace

New Generic Project Ctrl+Alt+Shift-N

Import Project Ctrl-I

Import Packed Project

Import Remote Project

Save All Projects Ctrl+Alt-S

Open All Closed Projects

Close All Open Projects

Rename F2

New Workspace

Switch Workspace

Clear Workspace

Preferences Ctrl+Alt-P

Save Preferences

Import Preferences

Recent ▶

Exit Ctrl-Q

Exit without saving Ctrl+Shift-Q

Online Help F1

New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name: chocolates

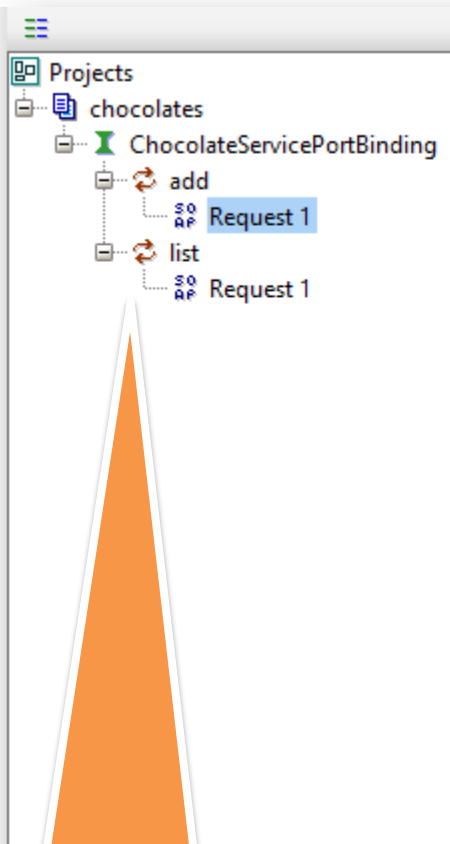
Initial WSDL: http://localhost:10000/ws/chocolates?wsdl

Create Requests: Create sample requests for all operations?

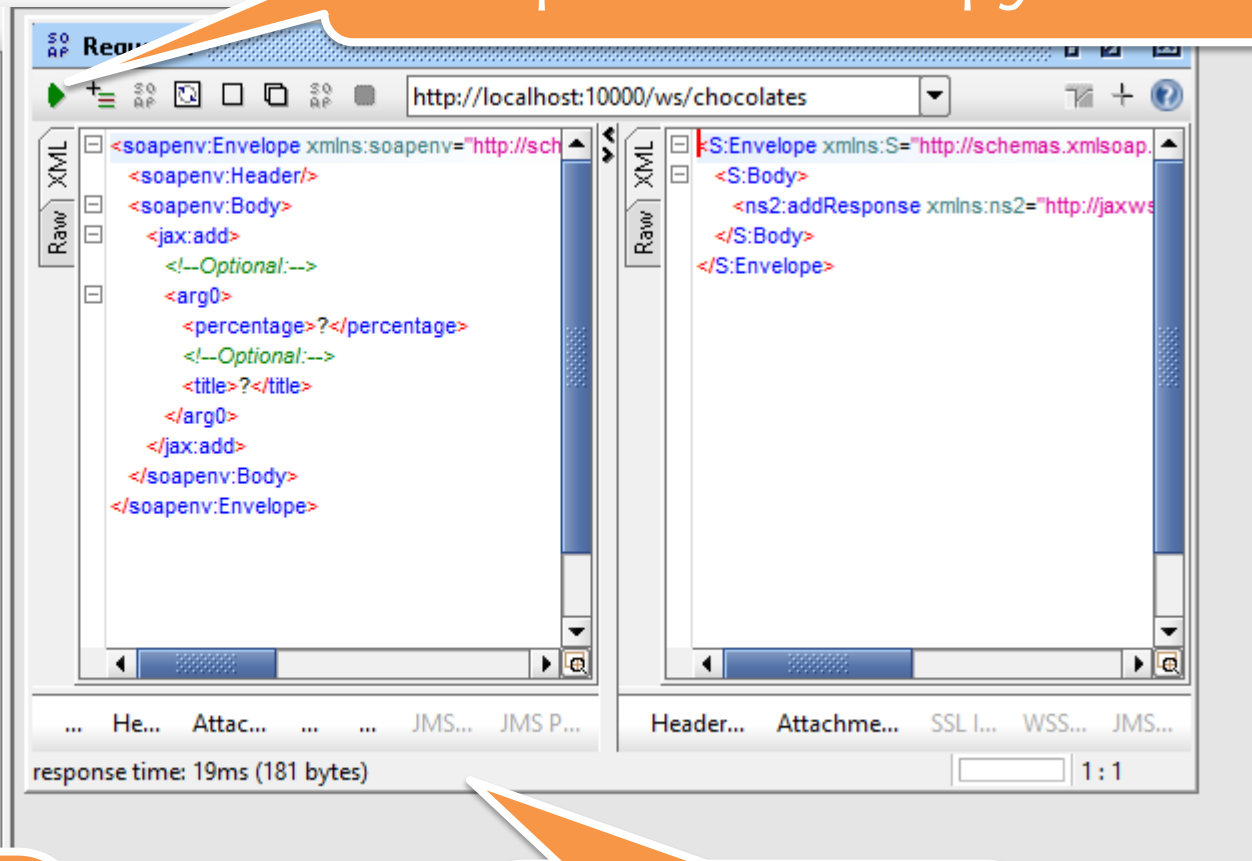
Create TestSuite: Creates a TestSuite for the imported WSDL

Relative Paths: Stores all file paths in project relatively to project file (requires save)

spustenie dopytu

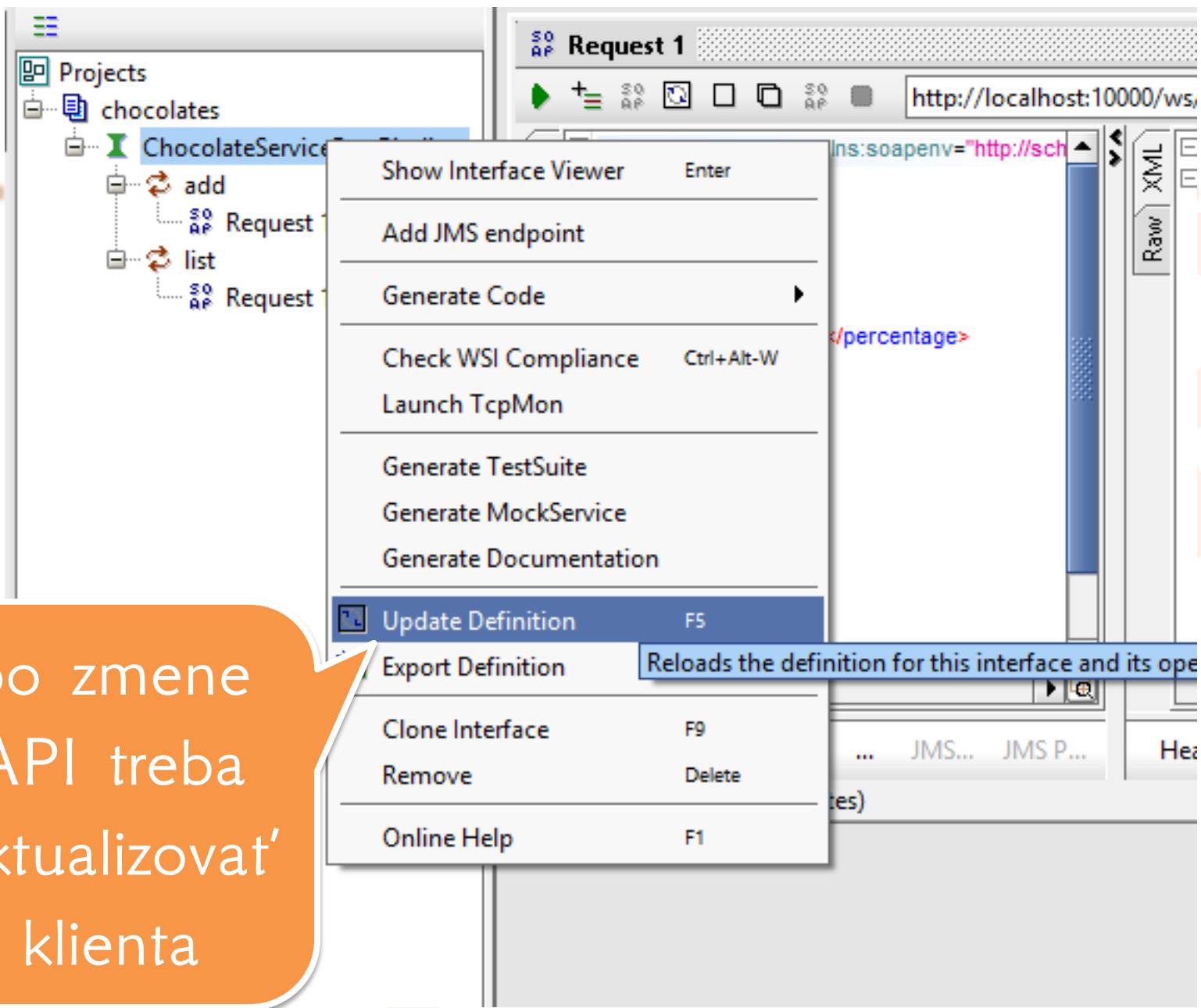


operácie a
ukážkové
dopyty



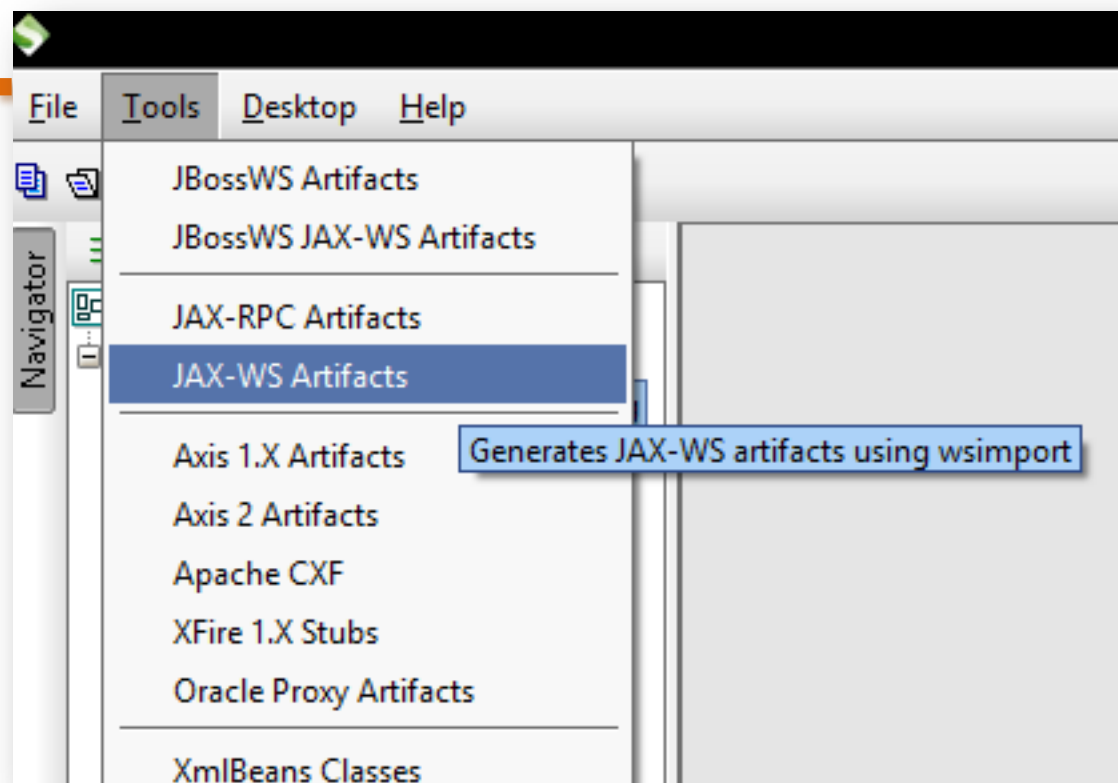
dopyt +
odpoved' zo
servera

po zmene
API treba
aktualizovat'
klienta





Vygenerujte JAX WS 2.0 klienta
pomocou SoapUI.



Generates JAX-WS artifacts using wsimport

JAX-WS Artifacts [X]

JAX-WS Artifacts [Settings]

Specify arguments for JWSDP/JAX-WS wsimport

Basic Custom Args

WSDL:

Target Directory:

Package:

Source Directory:

HTTP Proxy:


Catalog:

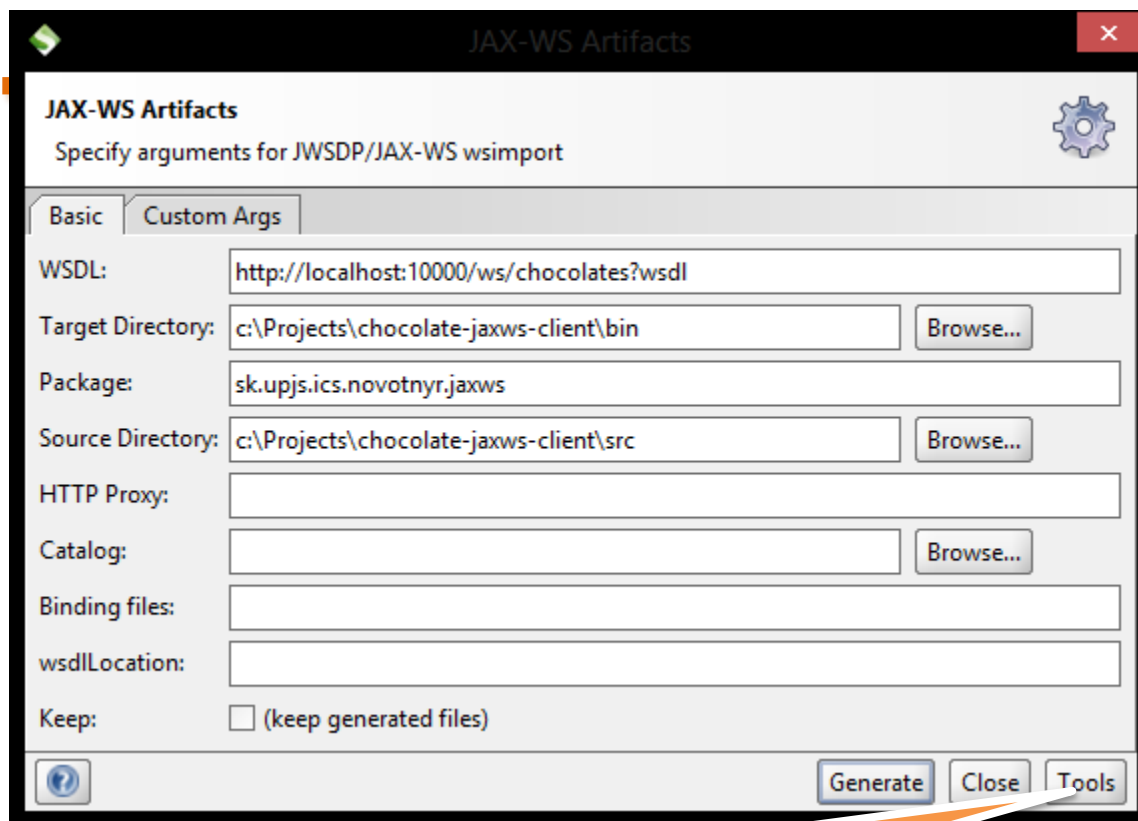
Binding files:

wsdlLocation:

Keep: (keep generated files)

Error [X]

 wsimport directory must be set in global preferences



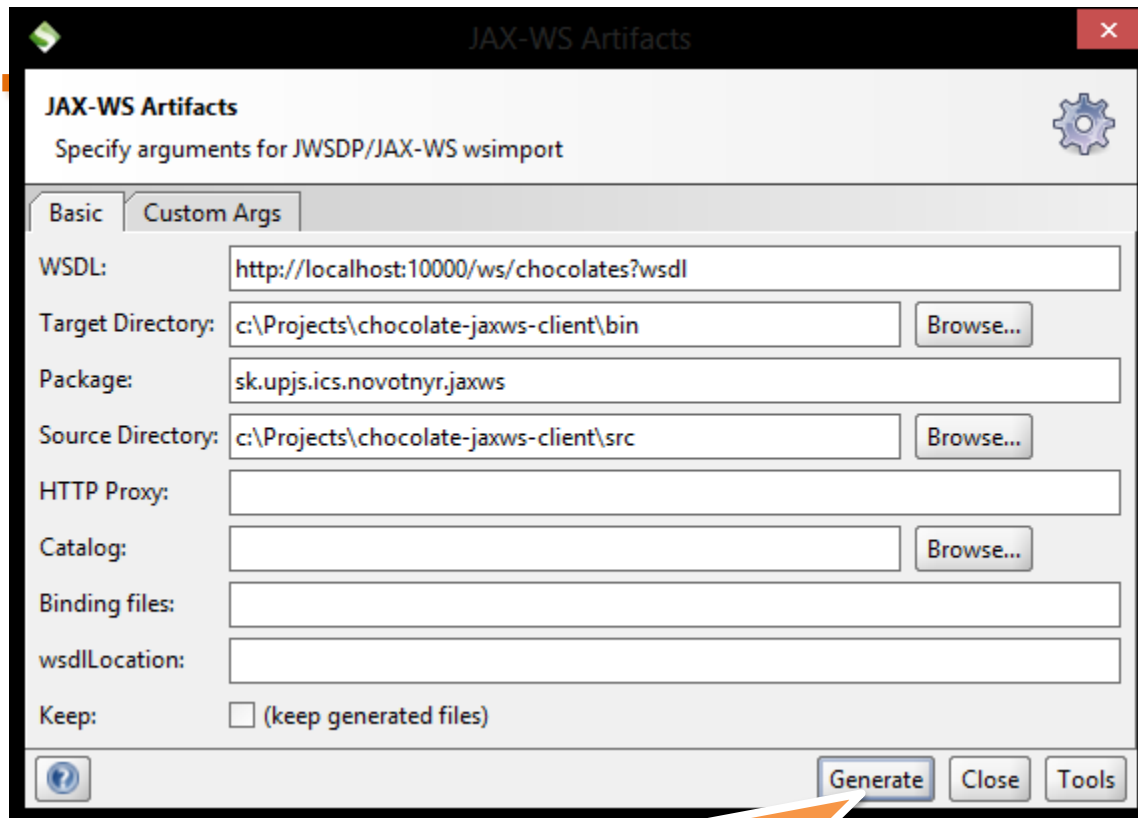
zmenit' cestu k wsimport

The image shows the SoapUI Preferences dialog box. The title bar reads "SoapUI Preferences" and "Set global SoapUI settings". On the left is a tree view of settings categories. The main area lists various settings, each with a text field and a "Browse..." button. An orange callout points to the "JAX-WS WSImport:" field, which contains the path "C:\java\jdk7\bin".

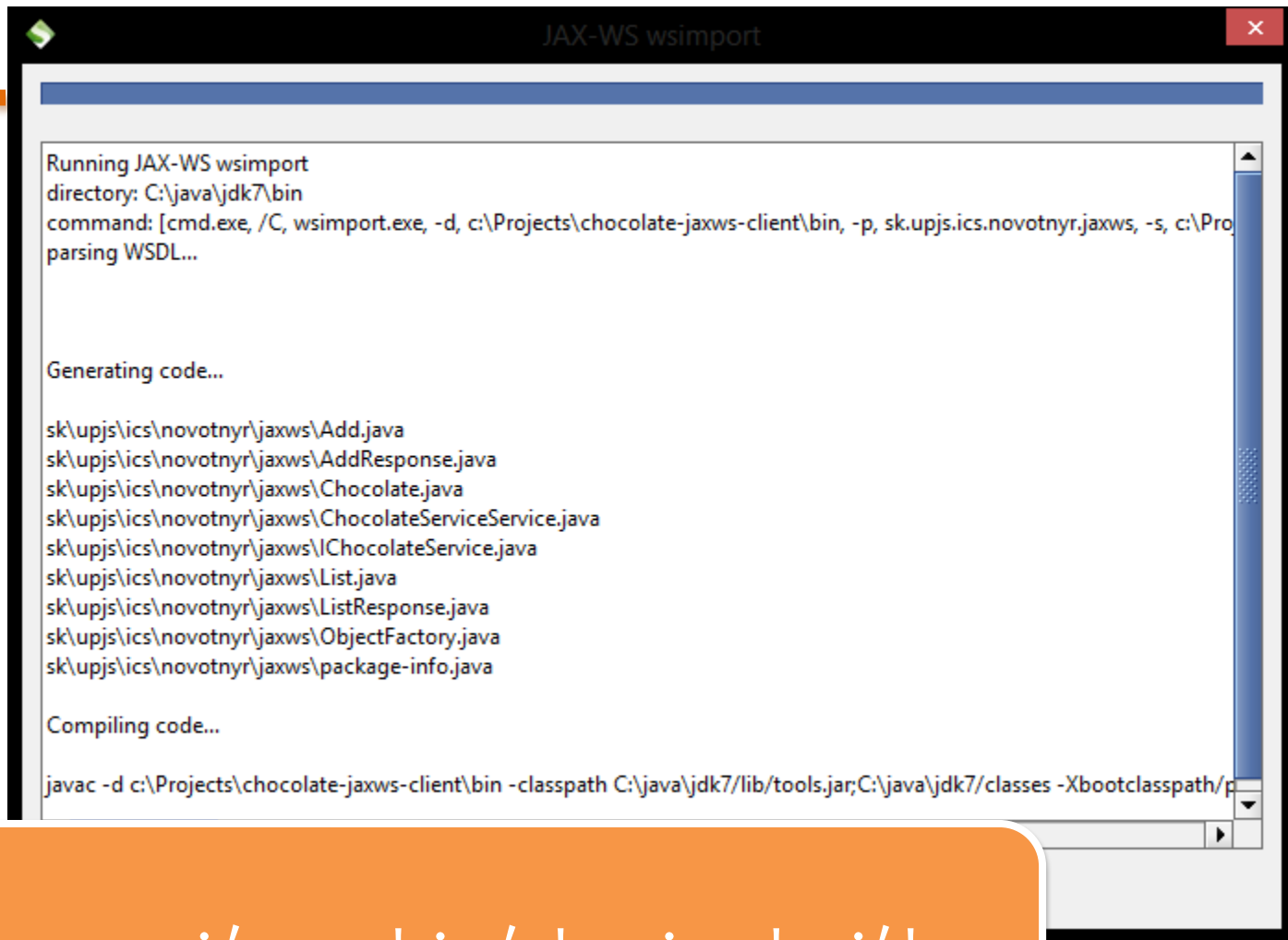
Setting Name	Value	Action
JBossWS wstools:		Browse...
JAX-RPC WSCompile:		Browse...
JAX-WS WSImport:	C:\java\jdk7\bin	Browse...
Axis 1.X:		Browse...
Axis 2:		Browse...
.NET 2.0 wsdl.exe:		Browse...
XFire 1.X:		Browse...
CXF 2.X:		Browse...
ANT 1.6+:		Browse...
GSoap:		Browse...
JAXB xjc:		Browse...
Hermes JMS:		Browse...

zmenit' cestu k wsimport

wsimport je v adresári
JDK, vedľa *javac*



Generovat!



```
JAX-WS wsimport

Running JAX-WS wsimport
directory: C:\java\jdk7\bin
command: [cmd.exe, /C, wsimport.exe, -d, c:\Projects\chocolate-jaxws-client\bin, -p, sk.upjs.ics.novotnyr.jaxws, -s, c:\Pro
parsing WSDL...

Generating code...

sk\upjs\ics\novotnyr\jaxws\Add.java
sk\upjs\ics\novotnyr\jaxws\AddResponse.java
sk\upjs\ics\novotnyr\jaxws\Chocolate.java
sk\upjs\ics\novotnyr\jaxws\ChocolateServiceService.java
sk\upjs\ics\novotnyr\jaxws\IChocolateService.java
sk\upjs\ics\novotnyr\jaxws>List.java
sk\upjs\ics\novotnyr\jaxws>ListResponse.java
sk\upjs\ics\novotnyr\jaxws\ObjectFactory.java
sk\upjs\ics\novotnyr\jaxws\package-info.java

Compiling code...

javac -d c:\Projects\chocolate-jaxws-client\bin -classpath C:\java\jdk7\lib\tools.jar;C:\java\jdk7\classes -Xbootclasspath/p
```

vygenerujú sa binárky i zdrojáky

Ako vytvoriť klienta v main()

- vytvoriť inštanciu SOAP service
- z nej získať inštanciu portu
- na porte volať metódy

```
ChocolateServiceService service = new ChocolateServiceService();  
IChocolateService port = getChocolateServicePort();  
List<Chocolate> chocolates = port.list();
```

port je nerozoznateľný od servisnej implementácie na serveri

Sumár

- vieme vytvoriť JAX-WS 2.0 server i klient
- Java<->Java
- čo však s **interoperabilitou**
 - medzi jazykmi a platformami?
- čo však so **stabilitou** API?

garantované riešenie: ručné WSDL