SOAPované distribuované systémy

Róbert Novotný

(UINF/KOPR, 18. november 2013)

20. storočie v Remote Procedure Calls

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

RPC

Čo ~~bol~~ je SOAP?

- *„SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment."*

  -- Simple Object Access Protocol 1.1

SOAP == webservisy

SOAP

envelope

- obsah správy
- adresáti
- povinnosť

encoding rules

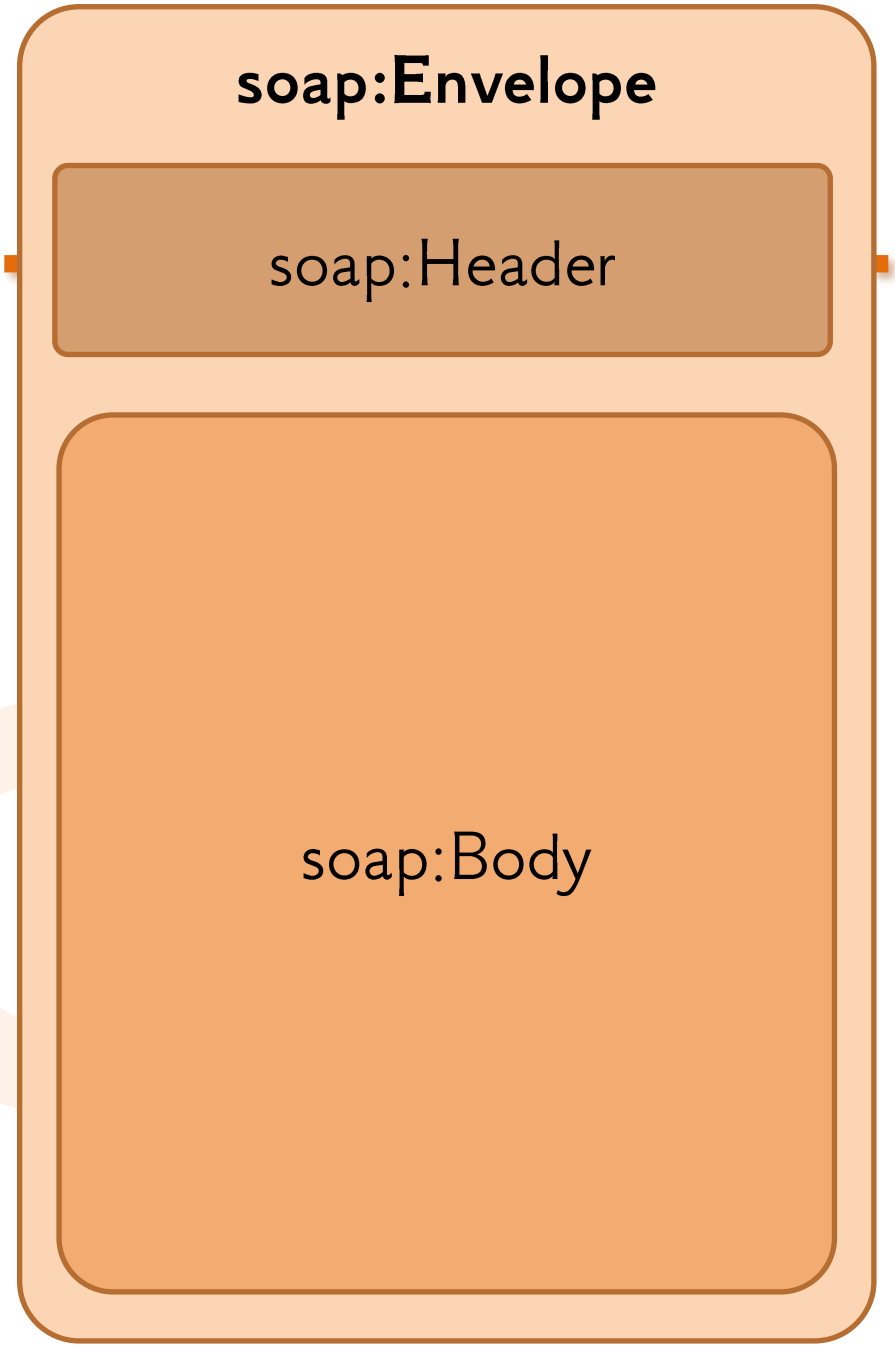
- serializácia objektov
- formát dát v obsahu

RPC representation

- URI cieľa
- špecifikácia vzdialenej metódy

SOAP

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



```
POST  http://www.webservicex.net/stockquote.asmx
SOAPAction:  "http://www.webserviceX.NET/GetQuote"
Host:  www.webservicex.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

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
```

```
<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!

Vytvorte "Hello World" endpoint v JAX-WS 2.0

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

JAX-WS: webservices v Jave

- **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

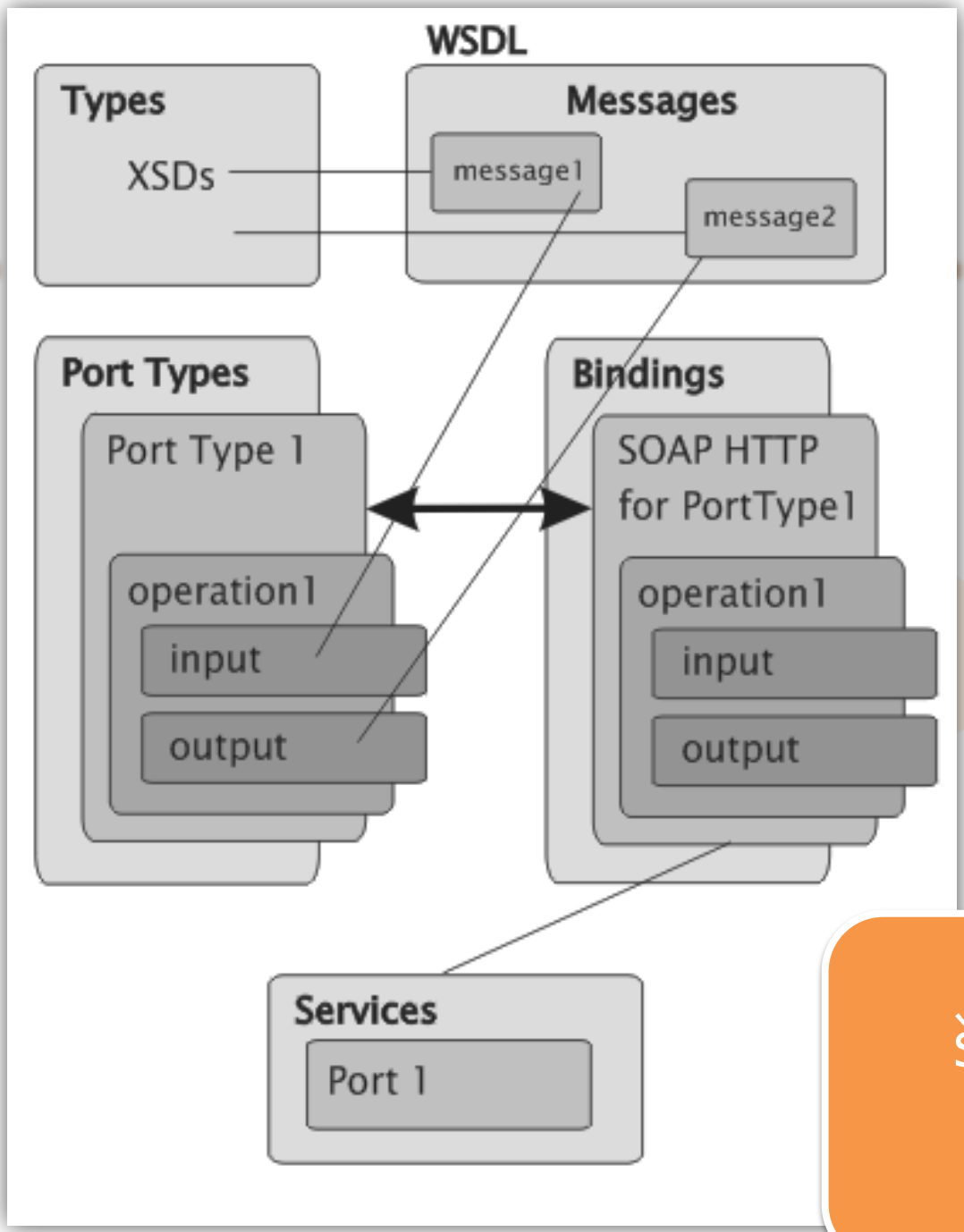
DEMO

[JAX-WS 2.0]

WSDL

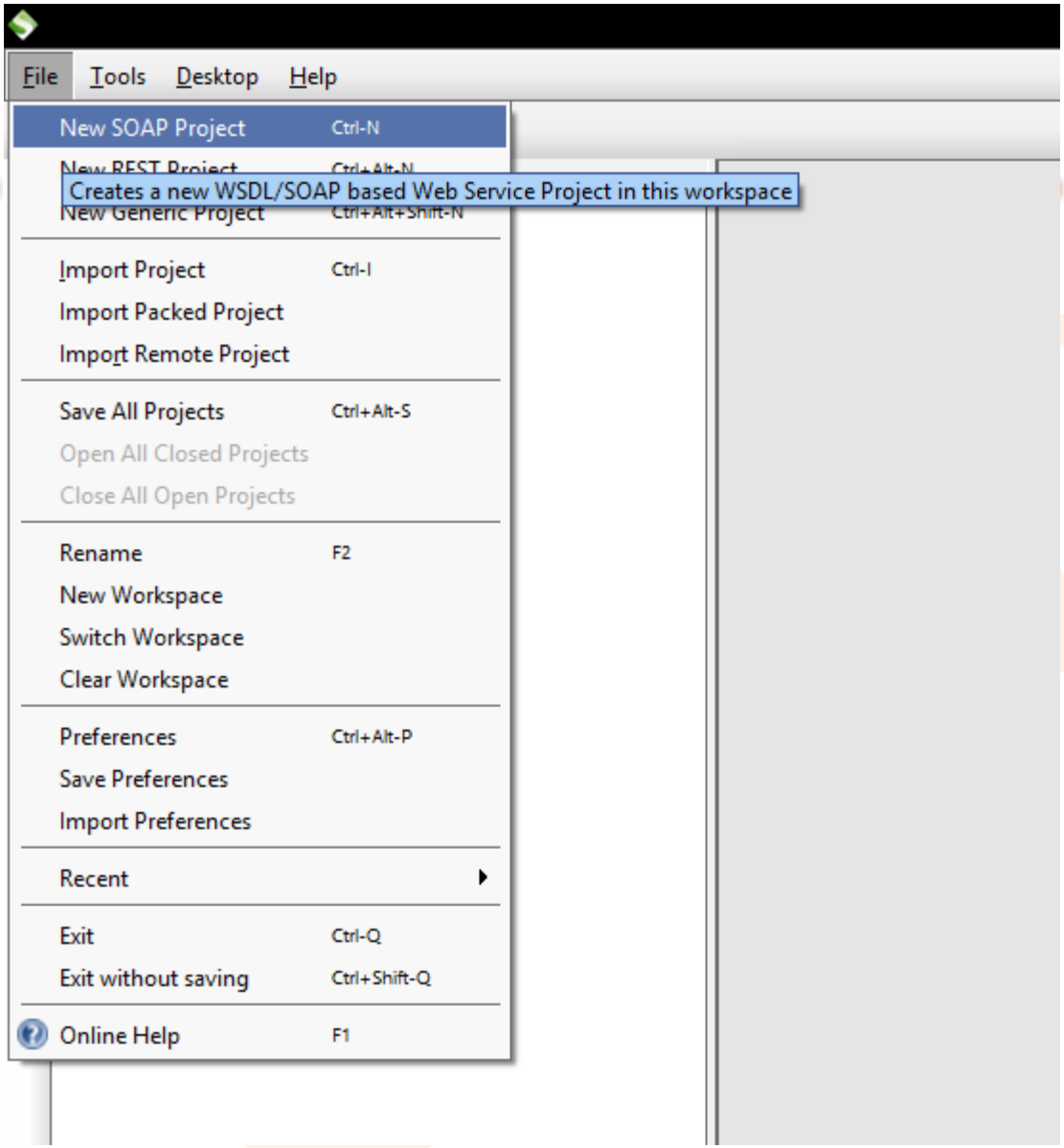
Web Services Description Language

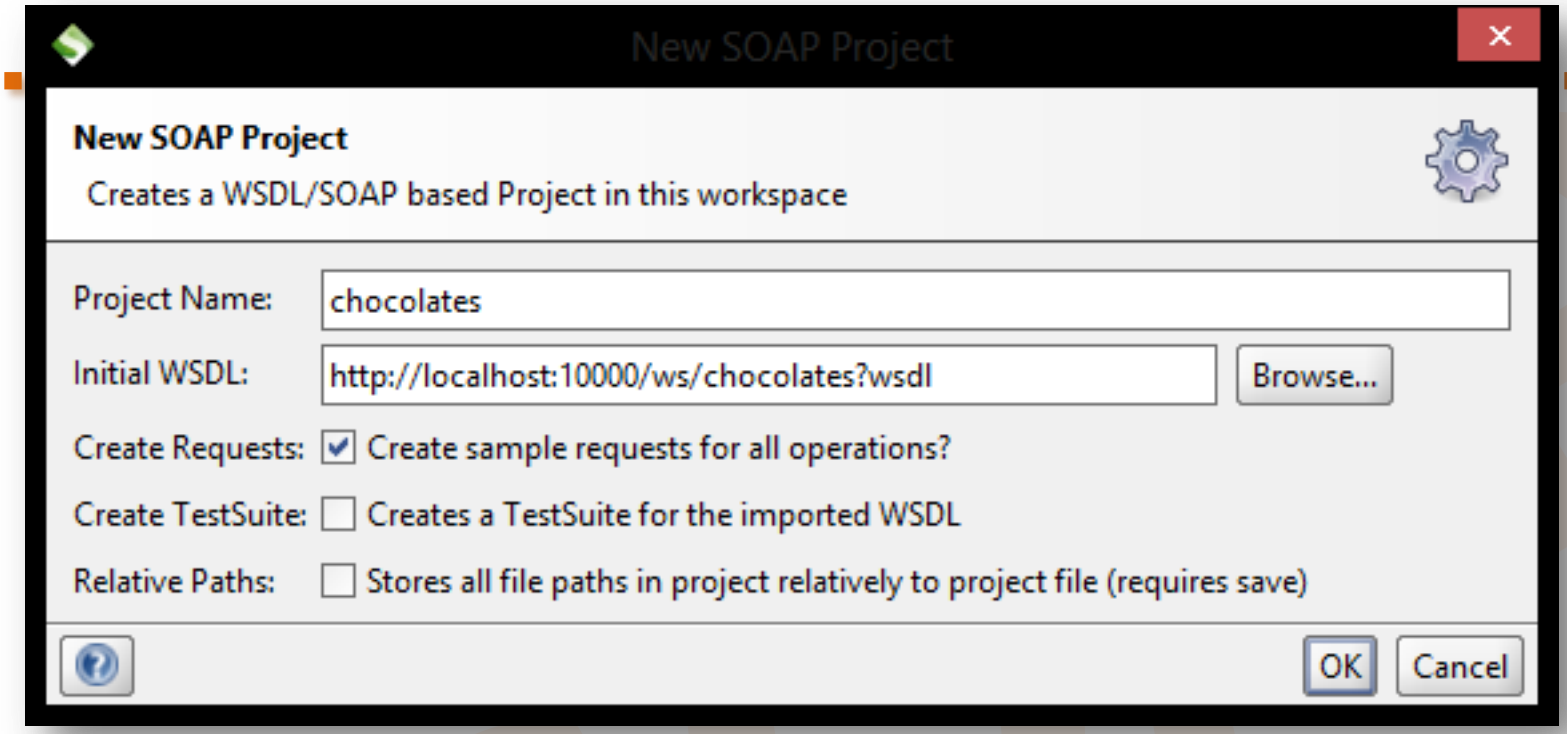
- metadátový popis webovej služby
  - operácie?
  - dátové typy?
  - štýly?
  - bindings?
- XML formát pre čítanie strojom
- z neho možno generovať **klientov**!

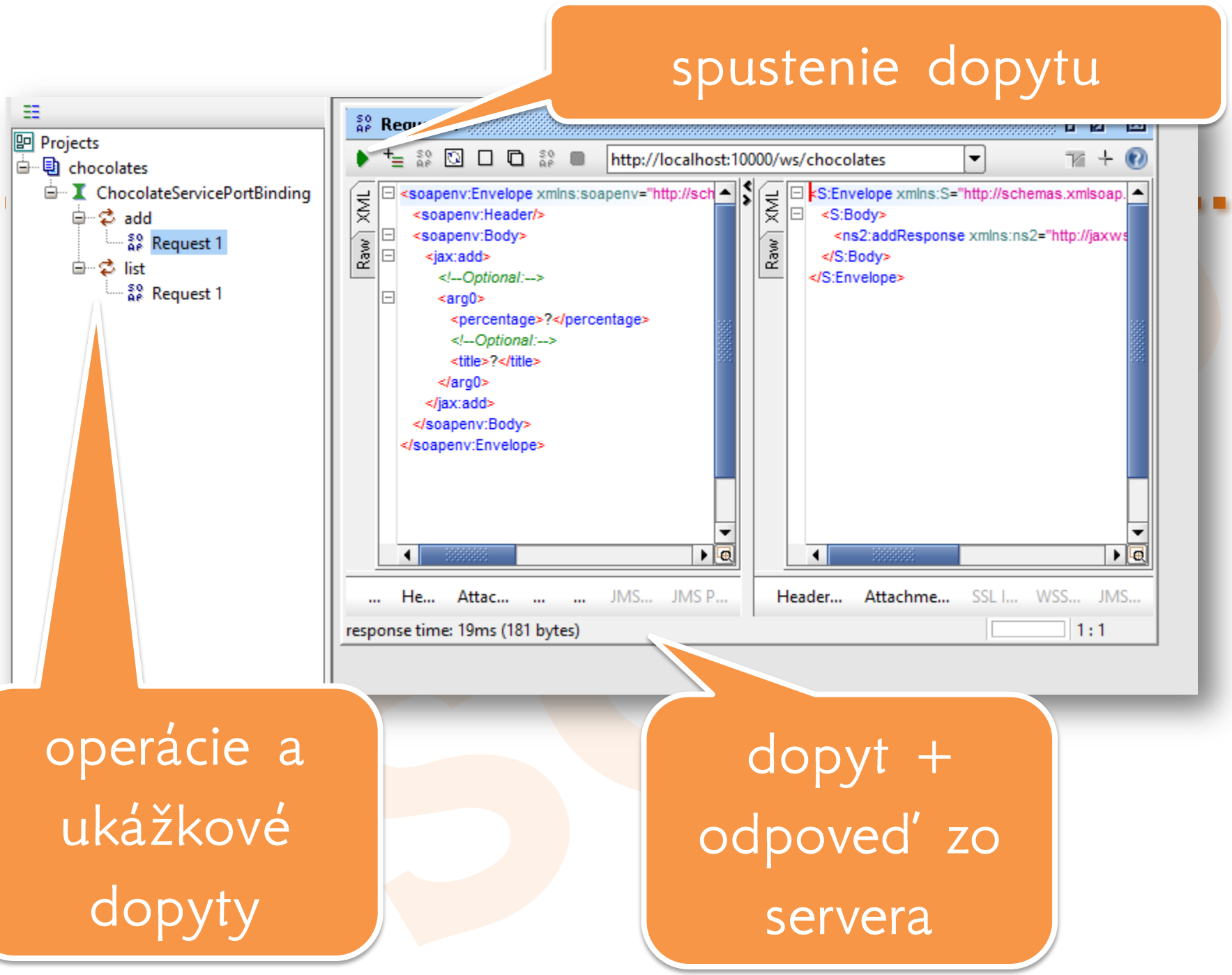


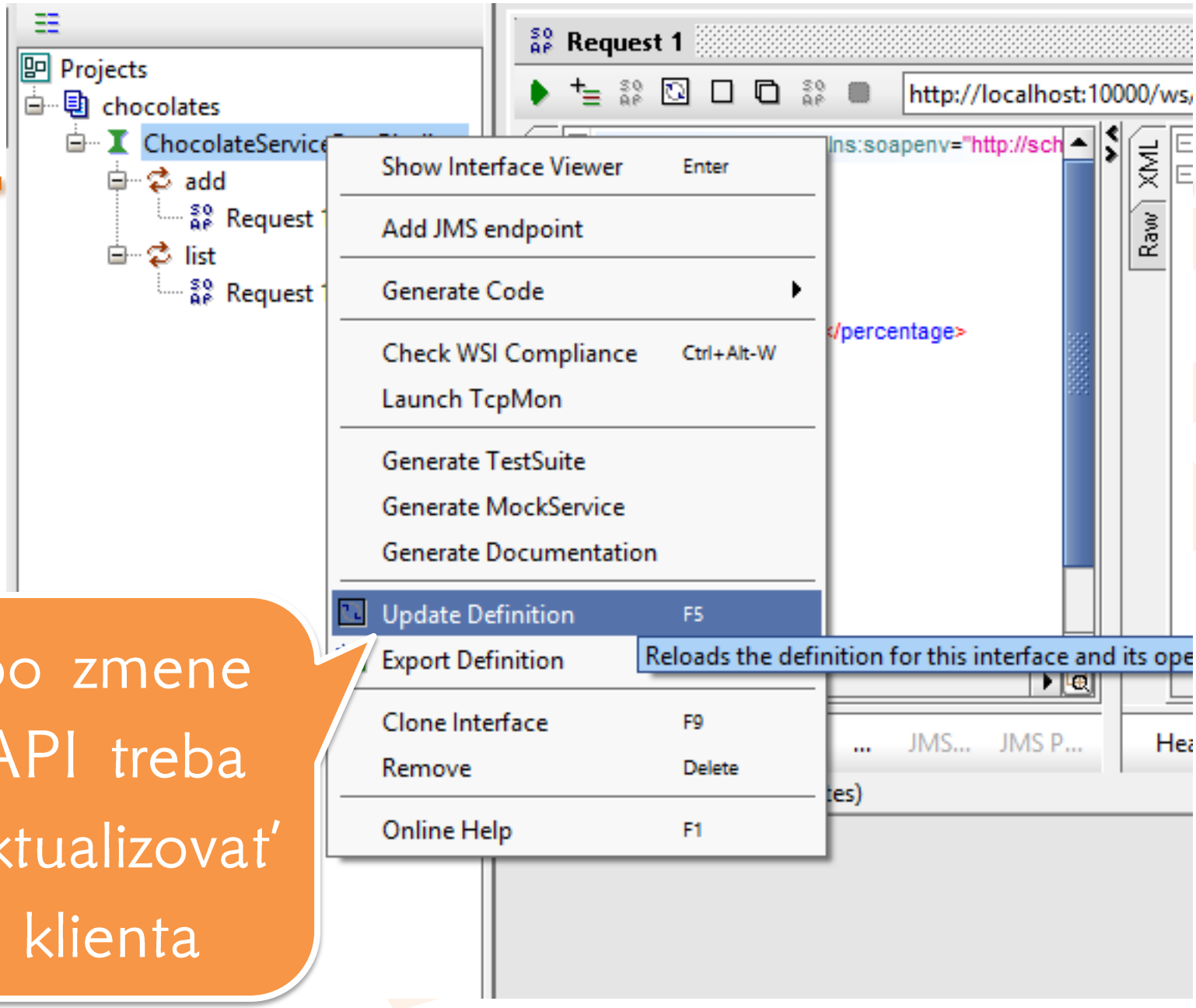
štruktúra
WSDL

Vytvorte klienta pre službu v JAX-WS 2.0 pomocou SoapUi

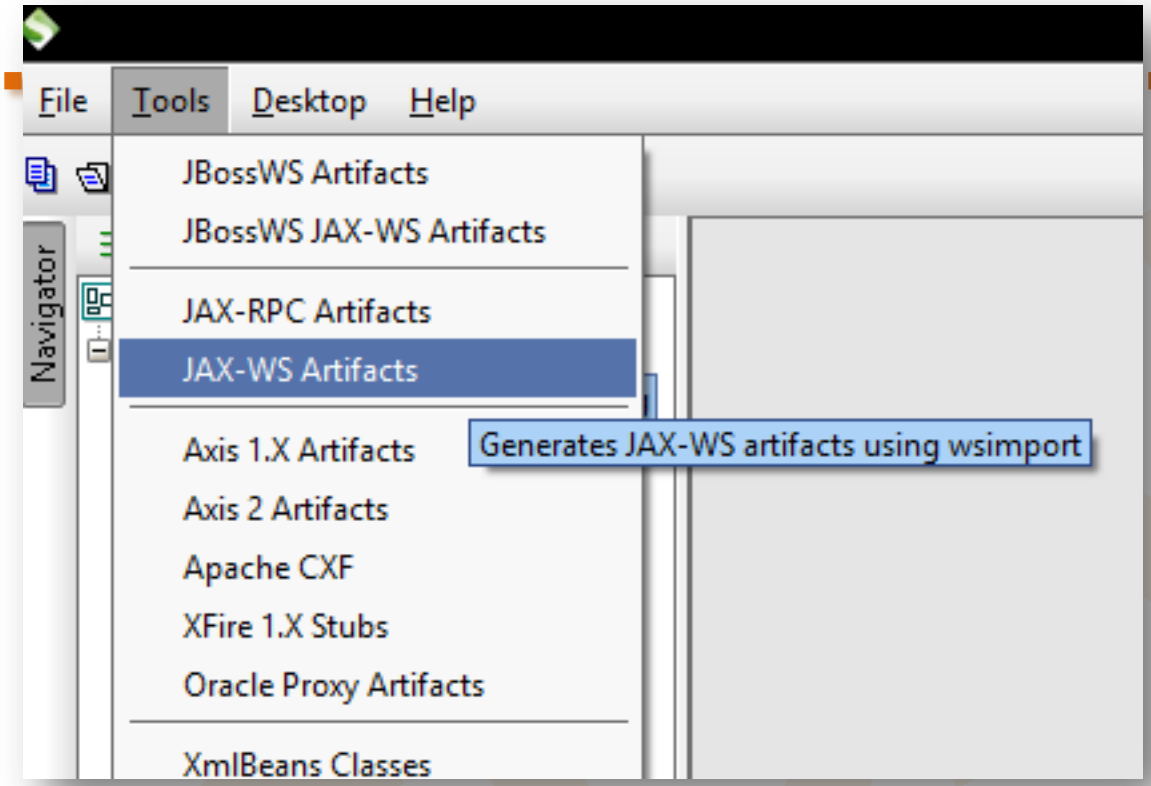

File
Tools
Desktop
Help
New SOAP Project Ctrl-N
Creates a new WSDL/SOAP based Web Service Project in this workspace
Import Project Ctrl-I
Import Packed Project
Import Remote Project
Save All Projects Ctrl+Alt-S
Open All Closed Projects
Close All Open Projects
Rename F2
New Workspace
Switch Workspace
Clear Workspace
Preferences Ctrl+Alt-P
Save Preferences
Import Preferences
Recent
Exit Ctrl-Q
Exit without saving Ctrl+Shift-Q
Online Help F1

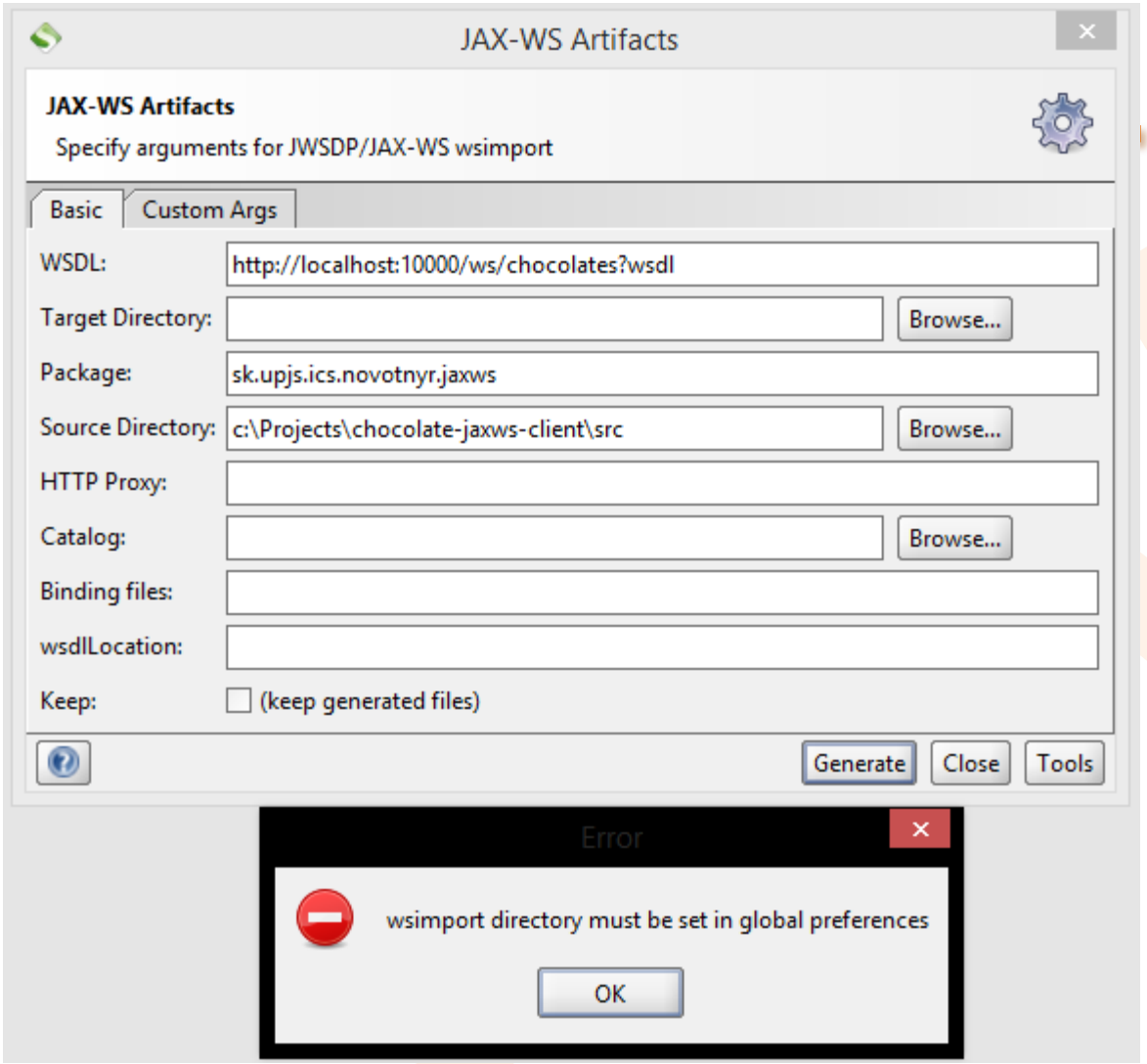

New SOAP Project
New SOAP Project
Creates a WSDL/SOAP based Project in this workspace
Project Name:
chocolates
Initial WSDL:
http://localhost:10000/ws/chocolates?wsdl
Browse...
Create Requests:
Create sample requests for all operations?
Create TestSuite:
Creates a TestSuite for the imported WSDL
Relative Paths:
Stores all file paths in project relatively to project file (requires save)
OK
Cancel

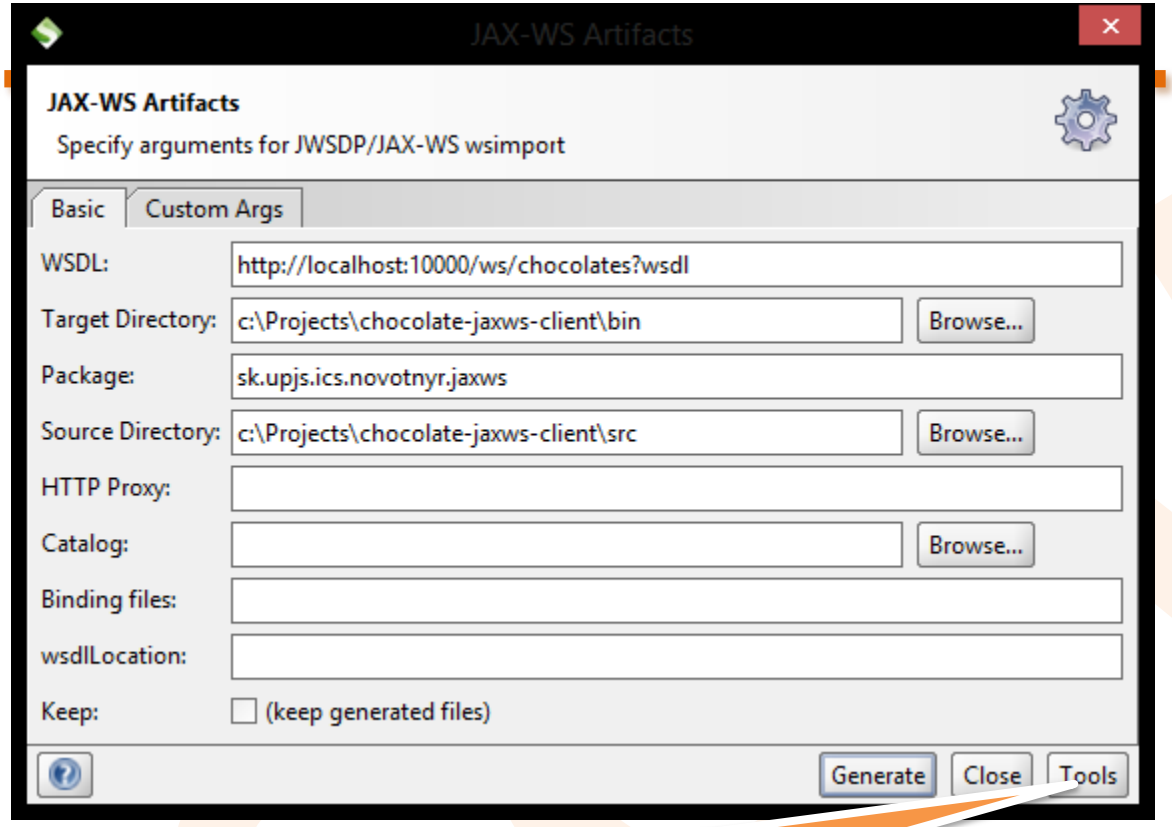

spustenie dopytu
Projects
chocolates
ChocolateServicePortBinding
add
Request 1
list
Request 1
http://localhost:10000/ws/chocolates
XML
Raw
<soapenv:Envelope xmlns:soapenv="http://sch
<soapenv:Header/>
<soapenv:Body>
<jax:add>
<!--Optional:-->
<arg0>
<percentage>?</percentage>
<!--Optional:-->
<title>?</title>
</arg0>
</jax:add>
</soapenv:Body>
</soapenv:Envelope>
<S:Envelope xmlns:S="http://schemas.xmlsoap.
<S:Body>
<ns2:addResponse xmlns:ns2="http://jaxws
</S:Body>
</S:Envelope>
... He... Attac... ... ... JMS... JMS P...
Header... Attachme... SSL I... WSS... JMS...
response time: 19ms (181 bytes)
1 : 1
operácie a ukážkové dopyty
dopyt + odpoveď zo servera


Projects
chocolates
ChocolateService
add
Request
list
Request
Request 1
http://localhost:10000/ws/
ns:soapenv="http://sch
</percentage>
XML
Raw
Show Interface Viewer Enter
Add JMS endpoint
Generate Code
Check WSI Compliance Ctrl+Alt-W
Launch TcpMon
Generate TestSuite
Generate MockService
Generate Documentation
Update Definition F5
Export Definition
Reloads the definition for this interface and its ope
Clone Interface F9
Remove Delete
Online Help F1
JMS...
JMS P...
Hea
tes)
po zmene API treba aktualizovať klienta

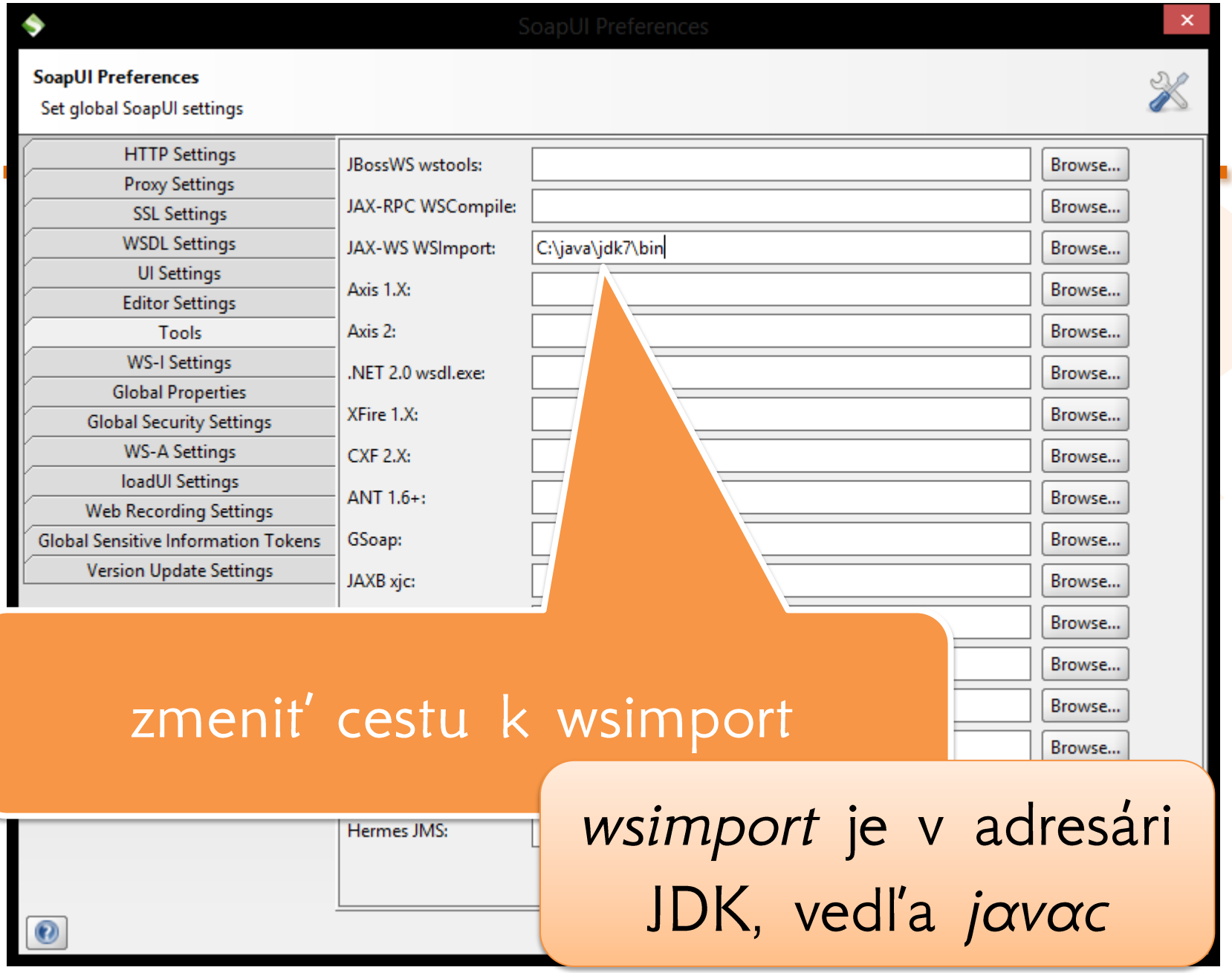
Vygenerujte JAX WS 2.0 klienta pomocou SoapUI.

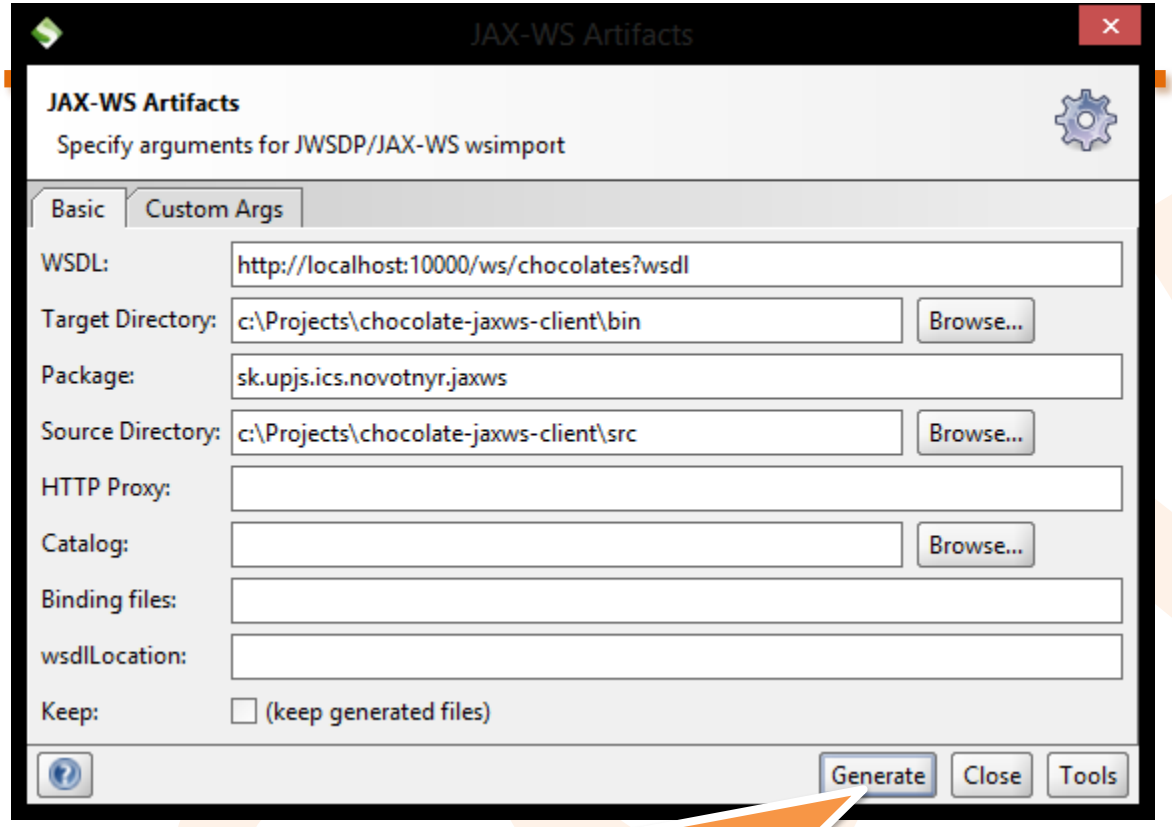

File
Tools
Desktop
Help
Navigator
JBossWS Artifacts
JBossWS JAX-WS Artifacts
JAX-RPC Artifacts
JAX-WS Artifacts
Axis 1.X Artifacts
Axis 2 Artifacts
Apache CXF
XFire 1.X Stubs
Oracle Proxy Artifacts
XmlBeans Classes
Generates JAX-WS artifacts using wsimport


JAX-WS Artifacts
JAX-WS Artifacts
Specify arguments for JWSDP/JAX-WS wsimport
Basic
Custom Args
WSDL:
http://localhost:10000/ws/chocolates?wsdl
Target Directory:
Browse...
Package:
sk.upjs.ics.novotnyr.jaxws
Source Directory:
c:\Projects\chocolate-jaxws-client\src
Browse...
HTTP Proxy:
Catalog:
Browse...
Binding files:
wsdlLocation:
Keep:
(keep generated files)
Generate
Close
Tools
Error
wsimport directory must be set in global preferences
OK

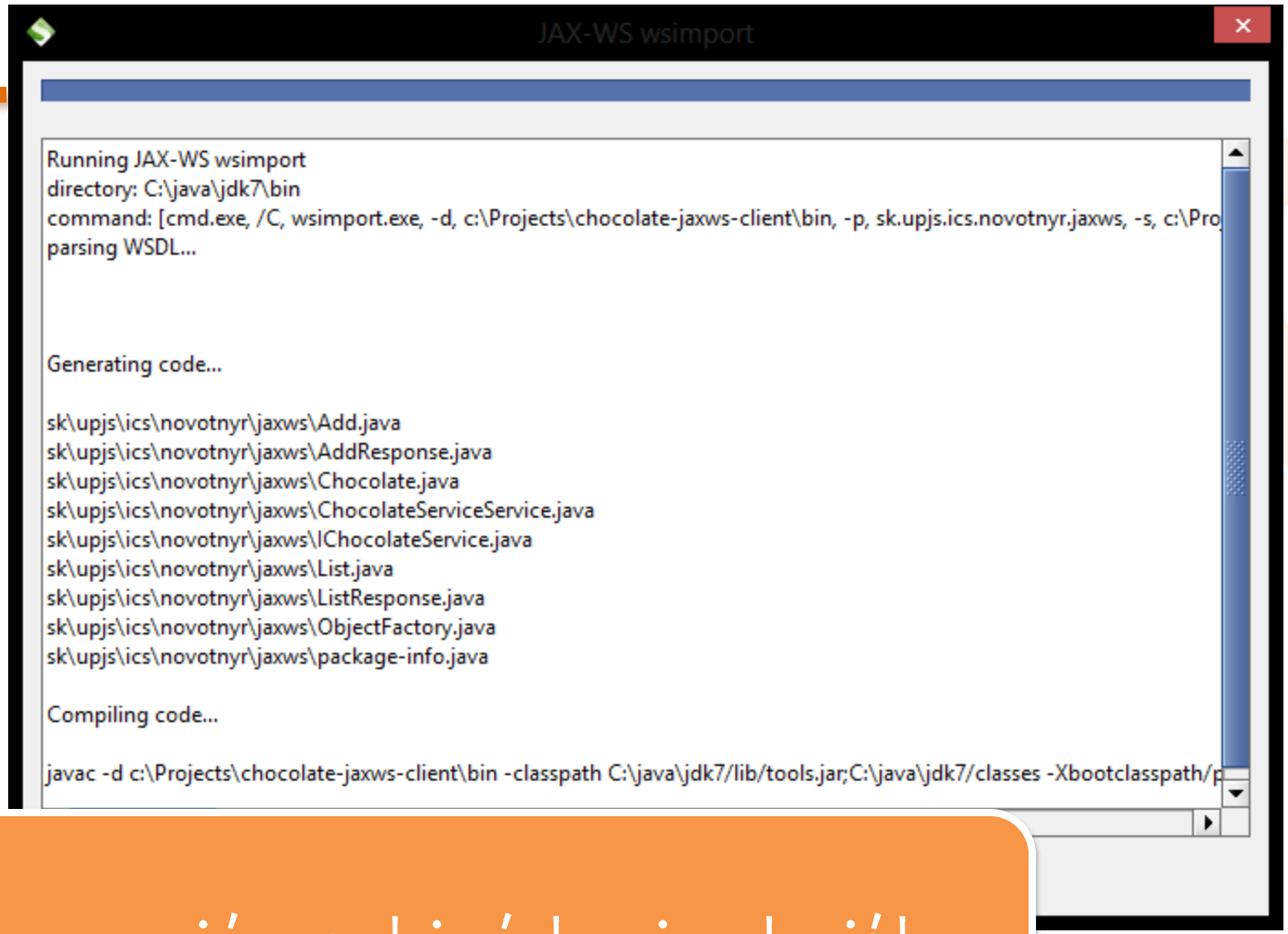


zmeniť cestu k wsimport


SoapUI Preferences
SoapUI Preferences
Set global SoapUI settings
HTTP Settings
Proxy Settings
SSL Settings
WSDL Settings
UI Settings
Editor Settings
Tools
WS-I Settings
Global Properties
Global Security Settings
WS-A Settings
loadUI Settings
Web Recording Settings
Global Sensitive Information Tokens
Version Update Settings
JBossWS wstools:
JAX-RPC WSCompile:
JAX-WS WSImport:
C:\java\jdk7\bin
Axis 1.X:
Axis 2:
.NET 2.0 wsdl.exe:
XFire 1.X:
CXF 2.X:
ANT 1.6+:
GSoap:
JAXB xjc:
Hermes JMS:
Browse...
zmeniť cestu k wsimport
wsimport je v adresári JDK, vedľa javac


JAX-WS Artifacts
JAX-WS Artifacts
Specify arguments for JWSDP/JAX-WS wsimport
Basic
Custom Args
WSDL:
http://localhost:10000/ws/chocolates?wsdl
Target Directory:
c:\Projects\chocolate-jaxws-client\bin
Browse...
Package:
sk.upjs.ics.novotnyr.jaxws
Source Directory:
c:\Projects\chocolate-jaxws-client\src
Browse...
HTTP Proxy:
Catalog:
Browse...
Binding files:
wsdlLocation:
Keep:
(keep generated files)
Generate
Close
Tools

Generovať!



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