Servlets and WebServices

Integration and Frameworks

Lecture 1

Servlet

- Servlet8.pptx

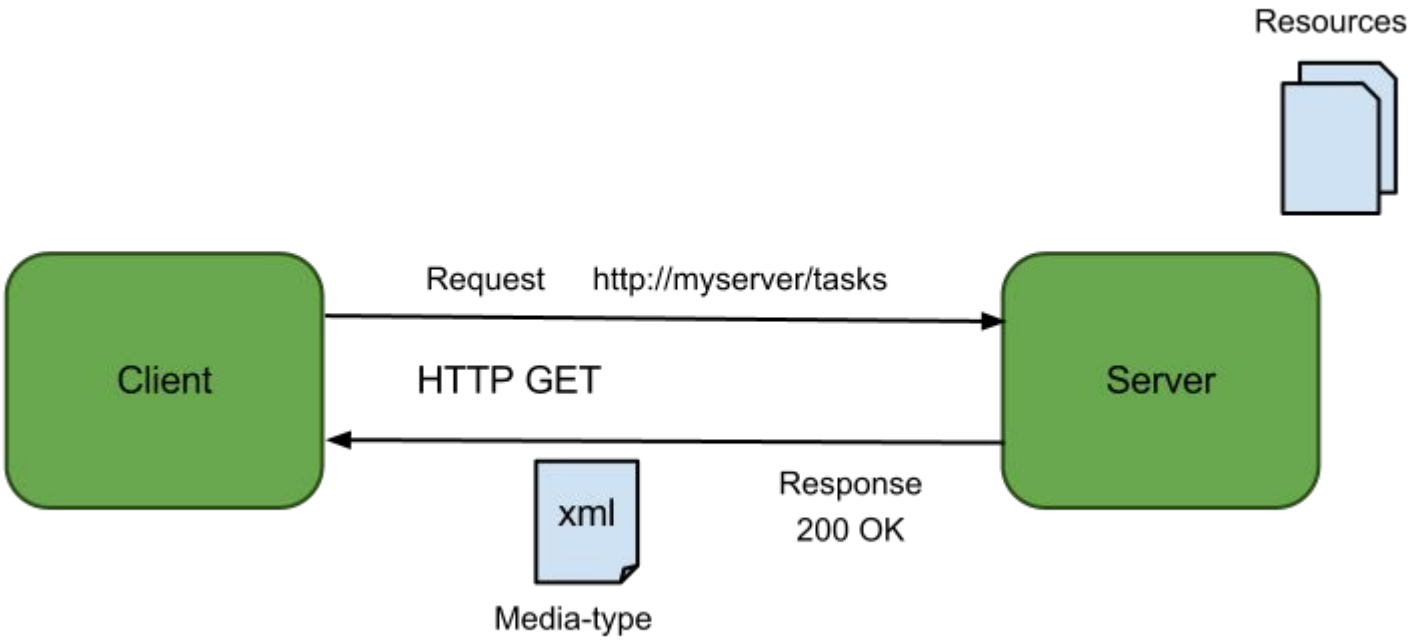
Servlet API version	Released	Platform	Important Changes
Servlet 4.0	Under development	Java EE 8	HTTP/2
Servlet 3.1	May 2013	Java EE 7	Non-blocking I/O, HTTP protocol upgrade mechanism (WebSocket)
Servlet 3.0	December 2009	Java EE 6, Java SE 6	Pluggability, Ease of development, Async Servlet, Security, File Uploading
Servlet 2.5	September 2005	Java EE 5, Java SE 5	Requires Java SE 5, supports annotation
Servlet 2.4	November 2003	J2EE 1.4, J2SE 1.3	web.xml uses XML Schema
Servlet 2.3	August 2001	J2EE 1.3, J2SE 1.2	Addition of Filter
Servlet 2.2	August 1999	J2EE 1.2, J2SE 1.2	Becomes part of J2EE, introduced independent web applications in .war files
Servlet 2.1	November 1998	Unspecified	First official specification, added RequestDispatcher, ServletContext
Servlet 2.0		JDK 1.1	Part of Java Servlet Development Kit 2.0

Web Services

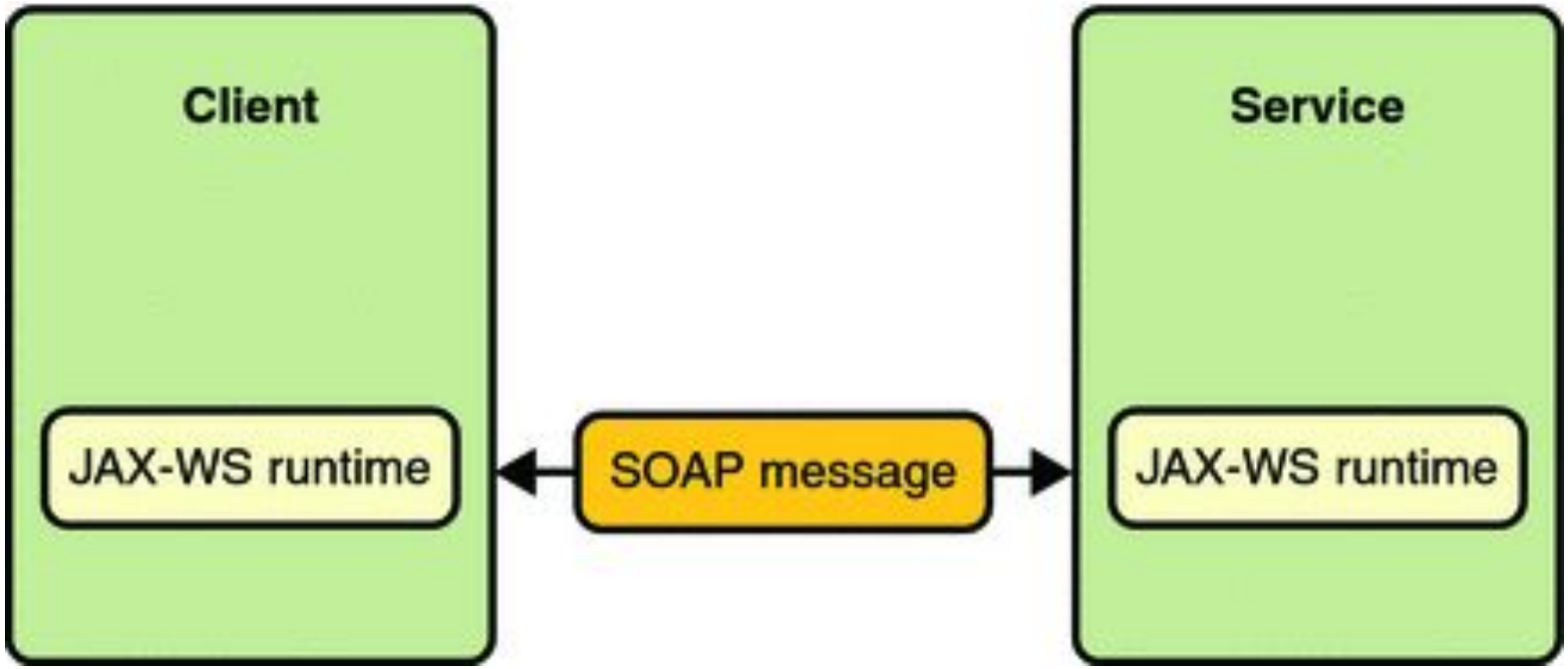
Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP)

- Java API for XML Web Services (JAX-WS)
- Java API for RESTful Web Services (JAX-RS)

RESTful Web Services



SOAP Web Services



REST vs SOAP

- REST – style of application architecture
- SOAP – family of protocols and standards
- REST is faster, easier and more scalable
- Use REST unless you have a specific reason to use SOAP

REST vs SOAP

- REST only supports HTTP Transport
- RESTful services does not maintain state (is stateless)
- RESTful services are not ACID compliant and does not support two-phases commits across transactional resources
- REST error handling is based on HTTP errors

RESTful Web Services: Methods

HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/ Replace	404 (Not Found), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	404 (Not Found), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

RESTful Web Services: Key instruments

- @GET, @POST, @PUT, @DELETE
- @Path
- @Produces, @Consumes
- @PathParam, @QueryParam, @FormParam, @CookieParam
- ClientBuilder.*newClient*()
- Validation - javax.validation.constraints

To read

- Java EE Tutorial. Servlets
  https://docs.oracle.com/javaee/7/tutorial/servlets.htm
- Servlet 3.1 specification
  https://java.net/downloads/servlet-spec/Final/servlet-3_1-final.pdf
- Java EE Tutorial. Web Services
  https://docs.oracle.com/javaee/7/tutorial/partwebsvcs.htm
- REST vs SOAP
  http://www.slideshare.net/vpgmck/soap-vs-rest-which-is-right-for-your-needs
- Using HTTP Methods for RESTful Services
  http://www.restapitutorial.com/lessons/httpmethods.html
- RESTEasy
- https://www.mkyong.com/webservices/jax-rs/resteasy-hello-world-example/

Homework

1. Configure you environment (Tomcat + IDE)
2. Write an application, which will allow to create a new person and save it to database (please, reuse previous homeworks) in 3 ways:
   1. Servlets
   2. Rest Service
   3. Soap Service (optionally)
3. Implement functionality, which allows to load a json file with persons via UI and save it to database

dbconnection.txt