Task-2

1.1 *Task-2.1* - Web Service with Web API (using Servlets and Jetty)

Goal

Learn to use SOAP Web Services and Servlets.

Prerequisites

Learn what a SOAP web services and Java Servlets are. Review lecture materials and suggested there learning materials.

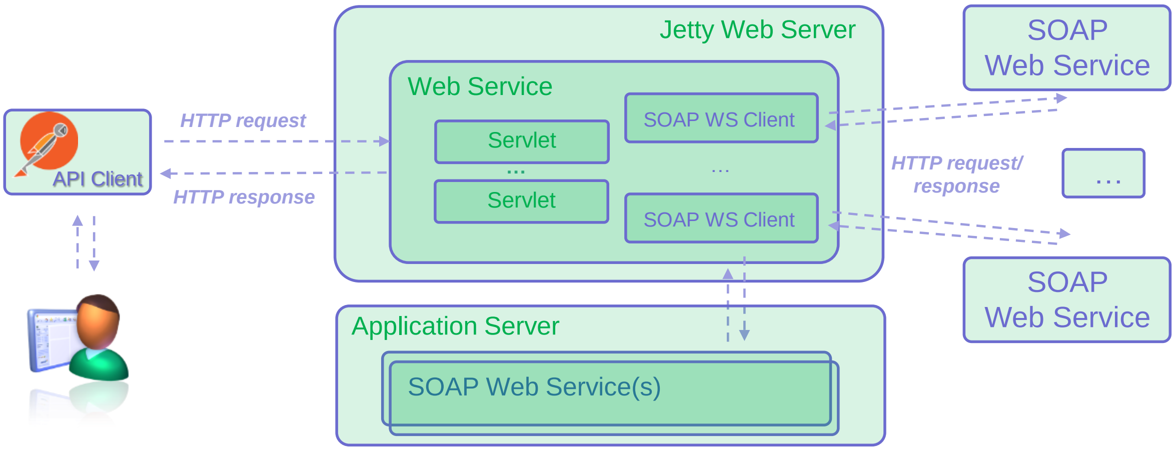
Task

This assignment contains a programming part as well as couple of slides explaining background behind your implementation (service(s) functionality, API, etc.). The main idea of the task is to build a Servlet based Web Service with corresponding Web API. This service should utilize other SOAP-based Web Service(s) to implement own business logic (you should implement at least one own SOAP-based Web Service, in addition, you may use any other third party SOAP Web Services).

- think about functionality that your service will provide;
- design a corresponding API *(similar to the example in the lecture materials)*;

path	method	parameters	response	failures
/…	…	…	…	…

- implement own SOAP Web Service(s) with necessary supportive functionality;
- implement Jetty-based API of the service that will utilize other SOAP Web Services.



1.2 *Task-2.2* - A Simple REST Client Exercise

Goal

Learn to use a RESTful API. The API used here is an example. The goal is that the student can use the experience from implementing against this RESTful API in another context as well.

Prerequisites

Learn what a RESTful web service is. Review lecture materials and suggested there learning materials. In addition, following IBM article https://developer.ibm.com/articles/ws-restful/ , https://developer.ibm.com/tutorials/i-rest-web-services-server1/ and the Wikipedia article http://en.wikipedia.org/wiki/Representational_state_transfer will be also useful. Interested readers can also read the original doctoral dissertation in which Roy Fielding introduced representational state transfer (REST) http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm mainly chapter 5 is of interest. The student is supposed to have basic knowledge about XML and/or JSON. Also basic knowledge of the HTTP protocol is useful (see also the Wikipedia article about it at http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol).

Task

This assignment contains a programming part as well as couple of slides explaining extra action(s) you chose. The service used is Dropbox (http://www.dropbox.com/). Dropbox is a great service for static file hosting. Anyone can put their files on Dropbox and use retrieved URL for accessing them from anywhere. Connecting to Dropbox using REST API allows manipulation with content via REST Clients.

First of all follow the following steps:

- create a basic account on DropBox for accessing its basic as well as Developer services.
- create an app for accessing its developer features (https://www.dropbox.com/developers/apps/create).
- fill out some information like *redirect-url* and *app name* etc. It will give you ***app-key*** and ***app-secret*** which will be used later in accessing its developer services.

A lot of supportive information you may find following the link: https://www.dropbox.com/developers. The RESTful API of Dropbox is described on this webpage https://www.dropbox.com/developers/documentation/http/documentation (https://dropbox.github.io/dropbox-api-v2-explorer/), read through the description of the interface. (You can also find an SDK from other parts of the developer documentation, but you cannot use it for this exercise since it hides all HTTP communication, which is exactly what this exercise is about).

Client is able to access Dropbox and perform various actions having an ***access token.*** An *access token* is just like a key that can be used for subsequent request to Dropbox for its various services like file upload, search, copying, etc. *Access token* could be retrieved via two steps authorization (https://www.dropbox.com/developers/reference/oauth-guide):

- Make GET call and use ***app-key***, ***app-secret*** and ***redirect uri*** etc. to request authorization and get the ***code*** that will be used in the next step to retrieve an ***access token***. Dropbox page would be open and ask user to authentication. After successful login Dropbox will ask user for required permission for app and at last redirect the user to mentioned ***redirect-url***.
- Make POST call and use ***code, app-key***, ***app-secret*** and ***redirect uri*** etc. to request ***access token***. Just parse the JSON and use ***access token*** for any service at Dropbox you like.

There are might be some problems in getting token and further interaction if you do not provide corresponding permissions in scope… (Collaboration – share.read….) while create and setup App in DropBox service.

Finally implement a Dropbox Client that enables following actions:

- *Connect* to your app in Dropbox via user authentication (has been mentioned before);
- *Get token* to be used for further manipulation with content (has been mentioned before);
- *Get information* about the user’s account (https://www.dropbox.com/developers/documentation/http/documentation#users-get_account);
- *Upload* an image (or some other file) to the folder of your Dropbox app (https://www.dropbox.com/developers/documentation/http/documentation#files-upload);
- *Do some extra action(s)* on your choice (do not forget to mention it in the presentation slides).

1.3 *Returning the task*

Provide the result of the task as an archive of the following files:

- ***presentation slides for both sub-tasks*** *(do not forget to list your group members)*
- ***TIES4560-Task-2_1.zip (project source codes)***
- ***TIES4560-Task-2_2.zip (project source codes)***

Send the Task-2 result as an archive ***TIES4560-Task-2.zip*** to lecturer (oleksiy . khriyenko @ jyu . fi)

Deadline: **19.09.2023 (before 12:00)**