ACP Assignment 1 Specifications (Programming Task)

19.01.2024

The first assignment has been designed as an preliminary to tackling larger and more complex situations in the second assignment.

Your main tasks can be summarized as follows:

1. Create a Java-REST-Service
   - Preferred with Spring Boot, though other frameworks can be used as well
   - Port 8080 is consumed
   - Implement one endpoint each for POST and GET
   - Proper parameter handling
   - Proper return code handling
   - JSON handling

2. Place the service in a docker image
   - **amd64** as target architecture – **not arm64** (this is relevant for the Mac users!)

3. save the docker image in a file called **acp_submission_image.tar**
   (it is in TAR format anyhow)

4. **place** the file **acp_submission_image.tar into your root directory of your solution**

   Your directory would look something like this:

   acp_submission_1
   acp_submission_image.tar
   src (the Java sources…)
   main
   …
   …

5. Create a ZIP file of your solution directory
   - Image
   - Sources
   - IntelliJ (or whatever IDE you are using) project files

6. upload the ZIP as your submission in Learn

The REST-Service has to provide the following endpoints:

1. **`uuid`** (GET)

   Return your student id in a HTML page embedded in a `<h1></h1>` tag.

2. **`writevalue?value=theValueToWrite`** (POST with **`theValueToWrite`** being the actual value to write)

   Write a value to the system. Any previous value will be overwritten

3. **`readvalue`** (GET)

   Read the current value from the system and return as `text/plain`. If not set an empty string is to be returned

4. **`callservice`** (POST with a body consisting of a JSON object)

   The body of the request contains a JSON-object in the following format:

   ```
   {
     “externalBaseUrl”: “the URL of the remote server”,
     “parameters”: “the parameters to combine with the URL”
   }
   ```

   The endpoint is to construct a URL from the “passed in” data (externalBaseUrl + parameters), call the endpoint with a GET, and return the data in the response as the same type as the original response was (so `application/json` remains `application/json`, etc.).

   Should this not be possible, you can degrade to `text/plain` and pass back what you can.

   Just be careful with constructing a valid URL as the passed in data might be incorrect (consider /, illegal constructs, etc.).

   This is a typical example where a webservice calls other services to for example consolidate data and returns the aggregation / analysis to the caller.

   For testing purposes you can use the URL `https://ilp-rest.azurewebsites.net/orders/2023-10-10` which will produce an `application/json` response

The following should be considered when implementing the REST-service:

- Do proper checking for URLs, data, etc. Don't handle anything not accurate (you will receive error data and requests!)
- Your endpoint names have to match the specification
- Storing data in a REST service either has to be done on a per session, or (as not differently specified) could be done on a global basis. Up to you.
- Test your endpoints using a tool like Postman or curl. Plain Chrome / Firefox, etc. will do equally for the GET operations
- The filename for the docker image file has to be exactly as defined as well as the location of it in the ZIP-file. Should you be in doubt, use copy & paste to get the name right

Should you need help:

- See the literature links in Week 2 and 3. You should find most information there
- If you cannot find an answer to your question, please post it on Piazza, though try finding it yourself first, please (as we have only limited capacity)

Disclaimer: We will not able to answer last minute questions right before the deadline, so please make sure you start the assignment in good time

Marking:

This programming task has a maximum mark of 35 / 100 points in relation to the entire ACP course.

The marks will be allocated purely on auto-tests based on the following criteria:

- Proper runnable docker image
- Proper behavior (functionality)
- Proper error handling
- Proper status codes

Should you fail to provide a runnable docker image according to the specification or provide no source code in the submission, no marking will be possible, and you will receive 0 points.