Graz University of Technology

Institute for Technical Informatics

Smart Service Development UE, SS2022
Konrad Diwold
kdiwold@tugraz.at

Group Assignment 4: Service Implementation (6pt)

Deadline: 09.06 2022; 23:59 CET

Introduction In this assignment, you will instantiate a part of the service system you developed in the previous exercises 2 and 3.

Motivation Now that you came up with a new cool service idea and have specified it it's time to start the implementation of the underlying services and deploy them.

1. **Identify service (0.4 pt)** Identify three sub-services out of the service system you established in exercise 3 and create a REST server stub of the service using swagger [1]. You can reuse a service specification of Assignment 3 to do so.

2. **Implement service (4 pt)** Implement the logic of the services by extending the server stub. If the service depends on other services create mock services to test the behavior of your service or link it to an existing service which provides the functionality.

3. **Testing (0.6 pt)** Think about integration testing strategies while implementing the service and document them, i.e., how can we test your services once you are finished implementing. How do you ensure the service interoperability. Provide example testing strategies in order for others to test your service.

4. **Virtualization (0.6 pt)** Now that your sub-services are implemented, tested and running it's time to deploy it in the cloud. To do so, have a look at docker [2]. Use docker to create/instantiate containers which fulfill the requirements of your service in terms of the installed packages. Write a dockerfile [3] which allows to instantiate your services in docker containers and test it (as specified in the task before).

5. **GIT hand-in (0.4 pt)** By the deadline, hand in via email: an invitation to a git repository containing all the sources you generated and dockerfiles which allows to deploy the services in a container. In a markdown file (in the archive) called readme.MD please state your name and student id, pitfalls you encountered while working on the assignment, and instructions about how to run and virtualize your service (if non-obvious) and answers to the exercises. Please use UTF-8 encoding for your documents and avoid special characters. You must make sure that your submitted code must work on our machines as well, so use relative paths if you add external libraries to your project, and include these libraries with your code.

[1] https://swagger.io/
[2] https://www.docker.com/
[3] https://docs.docker.com/develop/develop-images/dockerfile_best-practices/