David Arnold

dmarnol2

Lab5 SER422 Spring 2018

4/25/2018

The submitted lab took the monolith provided and broke it into 3 components. A servlet and two microservices as instructed. The microservices are unaware of each other. The servlet takes in data and forwards that data to the Lab5calc microservice which interacts with the database and returns numerical grade to servlet in JSON format. Servlet passes grade to lab5map microservice which then returns a letter grade in json format. Servlet parses data and displays to user.

Apache HTTPClient 4.5 is used to make calls to microservices.

JSON-Simple is used to parse json.

Service is lightweight only passing data needed by the microservices.

Quick Start

1. Pull both Docker images using:
   1. $ docker pull dmarnold/lab5calc
   2. $ docker pull dmarnold/lab5map
   3. or build locally using dockerfile following the steps provided in next section.
2. Run both Docker images using: (external ports are preconfigured as init params in lab5-app web.xml as 8001 for lab5 calc, and 8002 for lab5map. If changing ports, be sure to change them in both the following run commands and the lab5-app web.xml)

$ docker run -p 8001:8080 dmarnold/lab5calc

$ docker run -p 8002:8080 dmarnold/lab5map

1. Change lab5-app tomcat property in properties file found in the lab5-app folder, to your tomcat path on your local system.
2. Open a CLI and run “ant deploy” from the provided lab5-app folder
3. Open a web browser and go to the localhost your tomcat is configured for and include context. For example: <http://localhost:8080/lab5-app>
4. Enter values into field and click submit.

Dockerhub Images

Images for the lab’s microservices can be found on Docker Hub at locations:

dmarnold/lab5calc

dmarnold/lab5map

Both images are exposed on port 8080

Both images contain the war file it needs in the TomCat webapps directory to run the service.

Dockerfiles

Folders for each image are provided and contains dockerfile and resources needed for that build. The folders are named: calcDockerBuild and mapDockerBuild To build locally, open a CLI for the provided directory and type:

1. $ docker build -t <name> .
2. $ docker run – p <external port of your choosing>:8080 <name>

Both Dockerfiles have been slightly modified from given as follows:

For lab5calc – addition of copy command to copy lab5calc war file into TomCat webapps directory

For lab5map – addition of copy command to copy lab5map war file into TomCat webappd directory

Removed all build references to postgres database

Removed shell script since only need CMD to execute

Source Code

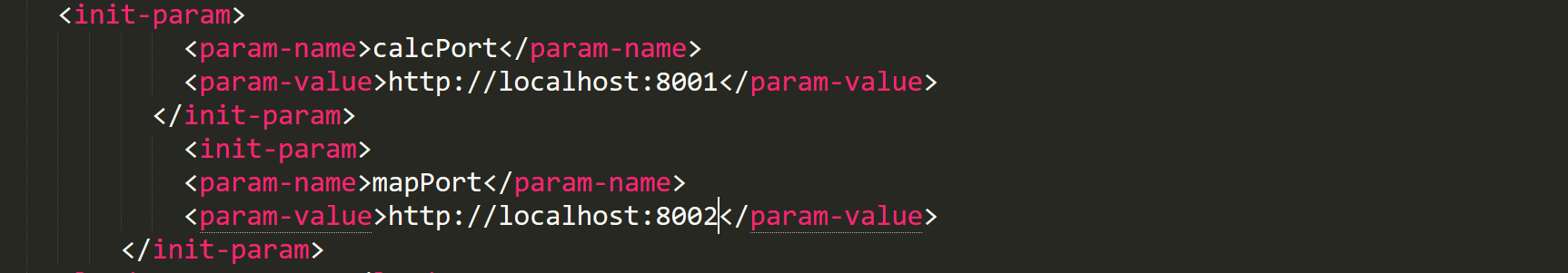
There are folders provided for each component: lab5-app, lab5-calc, and lab5-map. Each folder contains source code, build, properties, web.xml, and resources needed to build war file and distribute to Tomcat webapps directory.

Open a CLI for each directory and execute “ant deploy” to build.

In the lab5-app properties file, change the path for the tomcat directory to that on your local system

In the lab5-app web.xml file, the external ports for mapping the Docker container are provided as init params and can be modified for easy changing of ports. They are currently 8001 for lab5calc, and 8002 for lab5map.

To change ports, edit <param-value> with a new value using format as shown in image below taken from web.xml:



These values are used to form a String object in the servlet to pass to the HTTPClient. The String object representing the URL is created as shown in the image below:

