Knowledge Base Decision Tree

Build and Deployment Instructions

kbbuilddeploy.pdf

December 10, 2020

Prepared by: Brandon Martin

CS 4800.02

TABLE OF CONTENTS

1.0 REQUIREMENTS

1.1 Hardware Requirements

Computer

I/O Devices

Operating System compatible with Environment Requirements

Modem/Router with Internet connection

1.2 Environment Requirements

Web Browser

Java

Git

Netbeans IDE with JDE

Maven

Heroku

Postman

1.3 Required Source Code Files

KBTest.java

KBServlet.java

1.4 Recommended Source Code Folder

com/mycompany/kbfakenews/

2.0 BUILD INSTRUCTIONS

Building the Java code for this project is not necessary for deployment in this case, since our implementation of the Heroku service will be building the code each time automatically.

However, building the code can be extremely useful for further development or testing purposes, and as such instructions will be placed here to build a local version of the code.

If you have the recommended source folder import the project into Netbeans and, skip to step

1. Import the source code files into a new WebApp Project on Netbeans
    a. Choose MAVEN as your webapp manager
2. Through Dependencies, add the following dependencies into your project, downloading the corresponding jar files if required:
    a. javaee-api-8.0.jar
    b. json-20201115.jar
    c. jsoup-1.13.1.jar
    d. activation-1.1.jar
    e. hamcrest-core-1.1.jar
    f. javax.mail-1.6.0.jar
    g. junit-4.10.jar
3. Ensure that your pom.xml file corresponds to these dependencies as well.
4. Once complete, create an html file that will act as your url endpoint, then add into the corresponding folder in your project.
5. Setup a run configuration server within your project's properties on Netbeans. The built-in wizard will help you setup a server using one of many setup methods.
6. Run your configuration on Netbeans, using KBServlet as the main class.
7. Use a web browser to access your endpoint at localhost:8080

3.0 DEPLOYMENT INSTRUCTIONS

Deploying the Java code via Heroku is a relatively simpler process, especially when pulling directly from the github repository.

This does however, require that you know how to operate Github and Git and also are required to have a Heroku account associated with it.

1. Fork our repository containing all source files using Git: https://github.com/Flootah/CS-4800-Project
    a. Name the repository anything you want, for purposes of the instructions, we will assume the new repository is named *forkedproj*.
2. Go to the Heroku website and create an account: https://www.heroku.com/
3. Login to Heroku
4. Once at the dashboard, click "New App", and name your app
    a. you may name this app anything else, it need not be the same as your Github repo name
    b. you need not add this app to a pipeline
5. Select your deployment method as "Github" and select your forked repository
    a. you may need to login to Github to connect Heroku
6. Select your preferences for automatic deployment
7. Manually deploy the app!

Heroku will deploy the application and specify what url it can be reached at. From here, you may send POST requests to this url and have the expected return value given back.