Assignment 1: Servlets
Release Date: September 14, 2017
First Deadline: September 22, 2017 (Friday, 11:59 pm)
Due Date: Midnight September 24, 2017 (Sunday, 11:59 pm)
Total points: 100
(This is an individual assignment)

Objective:
In this assignment you will learn about Servlets, query parameters, and session tracking using cookies.

Introduction:
In this assignment you will implement a Servlet which will allow easy querying of OpenStack meeting data from OpenStack's eavesdrop website. You will also implement *sessions* for tracking user requests.

OpenStack (https://www.openstack.org/) is a project that is developing open source software for building infrastructure for cloud systems. Everything in OpenStack, such as meetings, discussion sessions, collaboration happen on IRC channels (Internet Relay Chat), and everything gets logged. For instance, the logs of different projects' meetings are available on the following site:
http://eavesdrop.openstack.org/meetings
On this website you will find folders for different OpenStack projects. Each folder further contains sub-folders for different years which contain logs for meetings that happened in that year.

Details:
You will implement a Servlet that will handle three query parameters:
- session
- project
- year

The session parameter will be used to control creation and termination of a session. The project and the year parameters will be used for determining what data should be shown in response to a request.

Servlet setup:
You should run your servlet to respond to following URL pattern: /openstackmeetings
You should set the context root of your Servlet to: /assignment1
With above settings, your servlet will be accessible at:
http://localhost:8080/assignment1/openstackmeetings

Interaction:
User can interact with your Servlet with or without initiating a session. Below is the interaction when user interacts by first initiating a session:

Step 1: Session start
http://localhost:8080/assignment1/openstackmeetings?session=start

Step 2: User requests data
http://localhost:8080/assignment1/openstackmeetings?project=heat&year=2017
http://localhost:8080/assignment1/openstackmeetings?project=solum&year=2015

Step3: Session end
http://localhost:8080/assignment1/openstackmeetings?session=end

Each call to your Servlet should result in displaying two sections:
–A history of all requests made to your Servlet. This will be shown under “History” section of the output.
–Data obtained from the eavesdrop site by parsing and interpreting project and year query parameter values. This will be shown under “Data” section of the output.

For example, after *Step 1*, the output would be History and Data sections that are empty.

History

Data

After second call of *Step 2*, the output would be:

History
http://localhost:8080/assignment1/openstackmeetings?session=start
http://localhost:8080/assignment1/openstackmeetings?project=heat&year=2017

Data
```
solum.2015-03-17-21.00.html
solum.2015-03-17-21.00.log.html
solum.2015-03-17-21.00.log.txt
solum.2015-03-17-21.00.txt
```

The “History” section contains the first two links (corresponding to the previous requests to your servlet). The Data section contains data obtained by using the query parameters, “solum” and “2015” from the second request and querying the eavesdrop website.

The output of *Step 3* (with session=end parameter) would contain History section which shows the list of all the request URLs to your servlet till that point. The Data section will be empty (since this particular request did not result in any data). So the output would look like:

History
http://localhost:8080/assignment1/openstackmeetings?session=start
http://localhost:8080/assignment1/openstackmeetings?project=heat&year=2017
http://localhost:8080/assignment1/openstackmeetings?project=solum&year=2015

Data

After a session has ended, subsequent interactions with your Servlet should not be tracked. Effectively, the History section will be empty after a session is terminated (or when no session is active).

Interaction without a session
Request:
http://localhost:8080/assignment1/openstackmeetings?project=solum&year=2015

History

Data
```
solum.2015-03-17-21.00.html
solum.2015-03-17-21.00.log.html
solum.2015-03-17-21.00.log.txt
solum.2015-03-17-21.00.txt
```

Allowed values for the query parameters are as follows:

Query parameter	*Allowed Values*	*Case sensitivity, other restrictions*
session	start, end	Should not be treated as case sensitive (Start, start, sTART, etc. should be treated as same)
project	Names of projects from eavesdrop site	Should not be treated as case sensitive (solum, Solum, soLuM, etc. should be treated as same)
year	Year numbers from eavesdrop site	Only digits are allowed (2016 is correct, twentysixteen is incorrect)

Other constraints
1. The query parameters themselves are case-sensitive (i.e. “session” and “Session” are different). You are required to support query parameters that are only lower case (“session”, “project”, “year”).
2. The “project” and “year” parameters are required to be passed in together. If either one is missing you should respond back with following message: “Required parameter <name> missing”.
3. The “session” parameter will be provided by itself. If “project” and “year” parameters are passed in the same request, you can ignore those.
4. Output format for the History section can be plain text or HTML.
5. If an invalid value is provided for project or year parameters you should respond with following error messages, as applicable: “Project with <name> not found”. Invalid year <year-number> for project <project-name>”. If in a single request both the project name and the year are invalid then your response should be “Project with <name> not found”.

Design details and how to get started:
Conceptually, there are four main parts to the implementation:
(a) Parsing query parameters
(b) Querying the eavesdrop website to obtain information based on the values parsed from the first step
(c) Creating, maintaining, and terminating user sessions
(d) Displaying the output (History section and Data section)

Start with setting up your development environment (Eclipse/IntelliJ/other IDE or text editor with command line). Try out sample servlet projects from the class github repository (listed under Useful material section below). Doing hw2 and hw3 will help you get started with the sample servlet projects.

Grading criteria:
We will grade your submission on following criteria:
- Correct working when using sessions
- Correct working when not using sessions
- Correct behavior when only project or year is provided
- Correct behavior when invalid value is provided for project or year
- Correct behavior when invalid values are provided for both project and year

Submission Instructions:
Name your submission folder as “assignment1”
Name your Servlet Java class “OpenStackMeetingsServlet.java”
Structure of assignment1 directory should be as follows:
assignment1/
    pom.xml
    src/OpenStackMeetingsServlet.java
    WebContent/WEB-INF/web.xml
    README.txt

Feel free to create additional Java classes if required.
When you create “Dynamic web project” for Servlet from scratch, following additional directories will get created. You don’t need to modify these.
    WebContent/WEB-INF/lib
    WebContent/META-INF/MANIFEST.MF

Include following information in README.txt
name: <your name>
eid: <your ut eid>
bitbucketid: <your bitbucket id>
comments: <Comments, if any>

Create a *private* repository on bitbucket (https://bitbucket.org/product/pricing?tab=host-in-the-cloud sign up using the free option if you don't have bitbucket account). Name the repository assignment1 and push/submit all the files that you create for your assignment to this repository.
Grant “read” access to me and Sailesh
- Usernames: devdattakulkarni, svsailesh

We will use the latest commit ID for grading. You don't need to submit anything on canvas.

Useful material:
Github:
https://github.com/devdattakulkarni/ModernWebApps/tree/master/Servlets
Following examples: hello-world, query-parameters, session-example, test-jsoup-1

Class notes on Servlets

Following chapters from Java for Web Applications book: 1, 2, 3, and 5

About deadlines:
First deadline: Initial submission which is will be graded for 3 points. You don't need to have assignment implementation completed by this date. This means if your first commit is on or before September 22 11:59 pm, you will receive points out of 100. If your first commit is after that then you will receive points out of 97. This deadline is to ensure that you complete all the required setup (Eclipse, Tomcat, etc.) early and not postpone it until last minute.

Due date: Deadline by which your assignment needs to be completed.

Late penalty:
5 points for each late day after the due date.

Collaboration policy:
This is an individual assignment. You are allowed to discuss concepts and high-level implementation questions with each other. But you are not allowed to copy or share code with each other or students who might have taken this class before. Final submission should reflect your own code.