Carnegie Mellon University
Master of
Software Engineering

**17-625: Design Patterns & Algorithms**
**Assignment A4: API Evolution**

**Learning Objectives:** To practice implementing and testing an API using a RESTful approach:

- Follow design process for RESTful API
- Implement REST interface for an API
- Properly test to avoid regressions

**Assignment Description:** Your task is to build a RESTful API, test it, and to reflect on the experience. You are to implement this in Java (using the latest version of Java that is compatible with Gradle). This class has two deliverables: first, a specification and corresponding tests (Design Task), and second, an implementation (Implementation Task).

The Design Task is due **Tuesday Nov 23rd**, and the Implementation Task is due **Friday Dec 3rd**.

**Background:**

Your client *Doctor's Assistant Inc* (DAI) has been very successful, but you want to expand into other markets. To better support clients in a wide range of industries, you have decided to create a RESTful API for your scheduling library.

You should follow the RESTful API design process as outlined in class, to ensure that you have identified potential clients and are developing a RESTful API that meets their needs.

**API Design Task:**

The API should support the existing functionality from Assignment 3:

- Get the current date
- Get the current day
- Get the current month
- Get the current year
- Add/modify/delete event
- Get event(s) for a given date

For this task, you have two tasks. The first task is to deliver documentation of your design process. Your documentation should include a list of activities, decomposed into steps. You should include a table that list:

- Activity
- Step
- Participants
- Description
- Endpoint (including HTTP Verb)

To complement this documentation, you should create a series of automated tests for each endpoint you describe in your documentation. NOTE: These tests are NOT expected to pass at this stage, as there is no implementation yet (they should, however, pass once the implementation is complete). For each endpoint, there should be a "happy path" test (showing everything working properly), and a test that demonstrates at least one error condition.

You might find it useful to use a library to help you. One library you might find useful is https://rest-assured.io/, but you may use other libraries as well.

You must test the contents of the response, not just the response code.

**Implementation Task:**

To implement your solution, you may use any of your existing code as a part of your solution.

We are giving you an empty repo for this assignment so that you may build on your existing code. We suggest your first commit should be importing your existing solution. The link to create a repo is here: https://classroom.github.com/a/RRt44Q3c

However, you will need to expose your existing functionality via RESTful approaches. You may use a library to help you implement the HTTP listeners. You might find one of the following libraries helpful, but you may use another library if you are more familiar with it:

- https://restlet.talend.com/
- https://sparkjava.com/
- https://javalin.io/

Your code should make all the tests you previously wrote pass. If you must change a test, you should discuss what changes are needed, and why the initial design was not correct in your reflection.

You will also need to provide well written execution instructions that work on all machines and platforms. No additional software (other than the latest version of Java) will be downloaded and the software will be executed from the command line (no IDEs will be installed). Be sure to test your software on a clean machine. If we can't execute your software by following your directions you will not get credit for the implementation.

**Be sure your code is well written, has appropriate comments, and handles exceptions gracefully**

**Code and Test Cases:**

Because there are several libraries you may potentially be using, you should also document how to run your tests and turn in those instructions so we can run your code. The instructions should be straightforward enough so that we can clone your repo in a clean directory and then run your code.

**Reflection Task:** With your code implementation, you should also turn in a reflection on your design and implementation answering the following questions:

- How does your RESTful design differ from your previous (library-style) design
- What changes did you need to make to your tests (if any) to get them to pass. Why were those changes needed, and do they shed any light on your design?
- Pick one design principle discussed in class and describe how your design adheres to this principle.

**Submission Instructions:** You will submit your source code and execution instructions to Gradescope, from Github. You must also submit a PDF document with your reflection, you may submit that to gradescope too. Structure the document clearly so that we can easily see the answers to the reflection questions.

**Assignment Grading:** Your grade will be calculated as follows (160 points total):

- ☐ Part 1:
  - ☐ Specification includes all existing functionality: 40pts
  - ☐ Tests specify both correct and incorrect API calls: 20pts
  - ☐ API design follows best practices: 20pts
- ☐ Part 2:
  - ☐ Code compiles: 20pts
  - ☐ Test cases adequately cover specified tests: 20pts
  - ☐ Reflection: 40pts