Assignment 0918
This is the last of our “setup and startup” assignments before we start digging into some face-deep database programming and development.

Outcomes
This assignment will affect your proficiency measures for outcomes 1a, 3a, 3d, 4d, 4e, and 4f.

Background Reading
Visit the course web site links to Tomcat, Hibernate, Jersey, and REST to get some further information about these parts of our technology stack.

For Submission
Software Setup
Make sure you have the following installed and running on your computer:
• Maven
• Tomcat
• Poster (Firefox add-on) or curl
For these tools, specifics don’t matter; just be able to run them on your rig.

Fork Headmaster
Fork the following GitHub public repository:
https://github.com/dondi/headmaster

Deploy Headmaster
Set up and deploy the Headmaster web service by following the Database Setup and Web Service Setup instructions on this wiki page:
https://github.com/dondi/headmaster/wiki/Project-Setup

Please submit an issue to the project if there are any glaring gaps or errors in these instructions.

Load Some Fake Data
The project wiki instructions create a very minimal Headmaster database. Just so you have some data to try right away, a headmaster.sql SQL script is available on the course web site. Run this script on your local headmaster database installation.

Make the (Web Service) Call
At this point, you should have an instance of Tomcat running the Headmaster web service on your machine. Now we can interact with it like any other web service on the Internet.
The service API wiki page for this project is a work in progress, but it should have some (if incomplete) useful information:
https://github.com/dondi/headmaster/wiki/Service-API

Make the following calls using any general-purpose HTTP request/response tool (e.g., Poster, curl):
1. A GET request on /headmaster/students with a query term of your choosing
2. A PUT request on /headmaster/students/id on a student of your choosing
3. A GET request on the student that you PUT in the previous request, to confirm that your modification did get saved
4. A POST request on /headmaster/events, taking note of the ID of the just-created event
5. A GET request on the event that you POSTed in the previous request, to confirm that your modification did get saved

Push the following to your private GitHub repository, under cmsi486/headmaster-service-practice:
• Precise representations of the requests that you made (URL, parameters, headers, payload)
• Precise representations of the responses that you got (relevant headers, payload)