1 Homework 3

Due: Apr 3 by 11:59p.

To be turned in as a PDF using the upload link for Homework 3 on mycourses.

Important Reminder: As per the course *Academic Honesty Statement*, cheating of any kind will minimally result in your letter grade for the entire course being reduced by one level.

1. When maintaining a large JavaScript codebase, a programmer decides to blindly replace all occurrences of anonymous functions which use the `function` keyword with fat-arrow functions. Why is such a replacement likely to be problematic. *5-points*

2. Many of the nodejs programs discussed in class contain `async` code. This `async` code can only be called using `await` which can only occur within `async` functions. The resulting *turtles all the way up* means that the top-level of any script must be called using an `await`. However, the top-level of most of our scripts contain top-level code which typically looks something like:

   ```
   (async function() { await someFn(); })();
   ```

   without any top-level `await`.

   Why does this code seem to work? **Hint**: Research the conditions necessary for a nodejs script to terminate.

   [Note that there is a proposal to add a top-level `await` to JavaScript]. *5-points*

3. An object `obj` has a `secure` property which contains security information. How can you set things up so that any use of the `obj.secure` property while the program is running is written into an audit log. You are not allowed to make any changes in the code which uses `obj.secure`. *10-points*

4. You are designing a program to manage the grades for a **single** course. The specification requires that your program exposes the following web services (a partial list):

   (a) A service for listing the grades for all students for a particular assignment.

   (b) A service for listing all the grades for all assignments for a particular student.

   (c) A service for listing a matrix of grades for specific students and assignments.

(d) A service to obtain details about a student, like their ID number, email address, etc.

(e) A service to obtain details about an assignment, like the date assigned, the date it was due, etc.

(f) A service for adding a new assignment to the course along with the grades obtained by the students in that assignment.

(g) A service updating an individual grade for a particular student and assignment.

For this proposed system spec-out each of the above web services using a **REST**ful approach. For each service discuss:

- Suitable HTTP methods.
- Possible URLs for that service.
- What would be the input parameters for the service.
- What would be the output for the service. Describe the links which can be included in the output in order to implement HATEOS.

Hint: You can refactor the above services. *20-points*

5. REST web services can use different representations like JSON, XML or HTML. Discuss the suitability of each representation for supporting HATEOAS. You are expected to use online resources to research HATEOAS and these representations. As usual, you should reference any resouces which you use in your answer. *20-points*

6. The specifications for *Project 3* have some HATEOAS deficiencies. Analyze the specifications and identify those deficiencies. *10-points*

7. What kind of HTTP caching directives would you recommend for each of the following web pages:

   (a) A blog which is updated at most once a day.

   (b) A page which only contains a search form for a library.

   (c) A page which shows the results for a library search.

   (d) A bank statement.

   (e) A page which displays the contents of a shopping cart.

   You may make any reasonable assumptions. *15-points*

8. Discuss the validity of the following statements. What is more important than whether you ultimately classify the statement as **true** or **false** is your justification for arriving at your conclusion. *15-points*

   (a) If two JavaScript objects share the same prototype, then any update to a property of the first object will also affect the second object.

(b) Both `POST` and `PUT` can be used to create resources.

(c) A session context in a server-side web framework can be used for storing user preferences.

(d) The session storage provided by modern browsers is tied to the session within a server-side web framework.

(e) JavaScript prototypes can be used to support multiple inheritance.