COMP3220 REST Coursework 2021/22

Key Information

Deadline: 15/12/21
https://handin.ecs.soton.ac.uk/handin/2122/COMP3220/1
The usual late penalties apply for late handins.
Lecturers: Nick Gibbins, Heather Packer
Value: 50% of the marks available for this module.

This coursework is to be completed individually.

Setting the Scene

You've just started your new job as a Web Architect at the IT department in the university. Your predecessor built a REST API to allow people to view details about workstations and workstation clusters in the university, however they did not finish the API's documentation.

Before they left for a meeting, your boss hurriedly scribbled this URI on a piece of paper:
http://comp3220.ecs.soton.ac.uk/workstations/27
Try to GET this, as a starting point. (You must connect to the ECS Virtual Private Network to access it.)

Part 1

Your first task is to browse and document the University Workstations API. You have one starting URI, but use what you have learned about HTTP and REST to explore the API and make note of what you find. Your colleagues recommend that you use an HTTP tool like Postman, RESTClient or cURL to do this.

Some questions that might help guide your exploration. You should include answers to these in your documentation:

- What methods can you use on each URI?
- What format can URIs take?
- What other URIs are possible for this service?

You must:

1. Add your choice of a new workstation to a cluster of your choice in the University Workstations API (figured out from the URI above), so that it appears in the list.

2. Write an HTML5 page to document how it the API works. You should write about each endpoint in the API that you find. You must have one page for the whole API, ordered and split into sections as appropriate. It is expected that the documentation for an API endpoint will include:

a. The URI template.
b. What each parameter in the URI means, and what types of value it takes.
c. Which HTTP methods can be used on the resource.
d. Provide example output for each type of resource.
e. Provide example input for each interaction that modifies a resource.

Note: We are assessing your ability to suitably structure an HTML5 document to represent the data, and to use some basic CSS to make the document readable and navigable. Therefore, **you are NOT allowed to use any frameworks to write your HTML5 or CSS**. Your artistic ability is not being assessed!

Part 2

Your boss now wants you to design an extension to the API which enables students and staff to book workstations. And enable administrators to manage the workstation clusters. The system should handle:

- Booking of workstations
- Cancelations
- Reporting of misuse and broken workstations

It's restrictions include:

Staff can:

- Book any number of workstations;
- Cancel a booking with no penalty;

Students can:

- Book a single workstation
- Cancel a booking with no penalty if 24 hours notice is given;
- If it's reported that If they do not use a booking then they are unable to book again for 14 days;

Staff and Students can:

- Report damaged workstations
- Report an unused workstation, or exceeding the time of the booking

You may include any additional support in your design for the needs of students and staff, which you see fit. This might include and is not limited to: allowing staff to override a student's booking; or appealing penalties of reported misuse.

You must:

1. Plan each of the steps that go into the booking process. Define URIs to identify the required additional resources and the methods that interact with them to perform the process and move through the states.
2. Draw a state diagram that shows a user's (both staff and students) interactions with the API. (like the one in Figure 4.1 of REST in Practice).
3. You must provide a structured representation of all your new API endpoints.

Please document your extension to the API using the Swagger framework, making sure you document:

a. The URI template.
b. What each parameter in the URI means, and what types of value it takes.
c. Which HTTP methods can be used on the resource?
d. Provide example output for each type of resource.
e. Provide example input for each interaction that modifies a resource.

Submission

You will be expected to submit the following:

- For Part 1:
  - Evidence of submitting a new workstation (screenshots or a text log of request and response).
  - 1 .html files and accompanying .css, documenting the API in the existing system.
  - Text document with critique of existing API as a RESTful interface

- For Part 2:
  - A state diagram (format up to you, expecting .png, .jpg, .pptx or .pdf)
  - Swagger documentation, detailing your designed extensions to the API to add the ability to book workstations.

Mark scheme

Section	Notes	Marks
Presentation of documentation LO: Use common Web technologies	Use of appropriate HTML5 and CSS to present the documentation professionally	12%

API discovery and documentation LO: Identify the key characteristics of the Web Architecture	Coverage of API endpoints and documentation of endpoints with demonstration of use (sample output) Critique of existing API as a RESTful interface	28% 5%
Design of extension to the API LO: Design RESTful Web applications, Apply the Representational State Transfer (REST) architectural style to Web application design	State diagram Appropriate design of URIs Appropriate use of HTTP methods Application of HATEOAS Richardsons maturity level of API design	55%
	Total:	100%