17625 A1 Reflection

by Yucen Xu

1. What were some of the alternative design options considered? Why did you choose the selected option?

An alternative design is to make getting date/day/month/year a post or put request, and put the userID in the request body. Current option (GET /date/dat/user={userID}) is selected because it adhere to the RESTful style. Post request is not idempotent and used to add a new resouree, put request is used to update a resouce, while get request is idempotent and used to get a resource. In this case, getting date is idempotent, therefore I chosed get for this use case.

2. 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?

To pass the test cases for deleting an event, I removed the body for the delete request. Instead, the eventID is put in the URL. This change is maded because using request body for delete request is not a good practice.

I also changed the endpoint for getting current day, month and year. They are using the same endponint as getting current date, and passes a parameter after this endpoint. This do shed light on my design, because those get requests are handled by the same class(ServerResource), so it's better to use the same endpoint.

3. Pick one design principle discussed in class and describe how your design adheres to this principle.

The design adheres to the "Easy to evolve" and "Easy to maintain code that uses it" principle. Currently for adding/editing/deleting an event, userID is a field in the request body. Although now we only check if userID is a valid interger larger than 0, in the future, different userID can has different privilege. We can easily check if the user has permission to perform the action from server side without changing the code in client side.