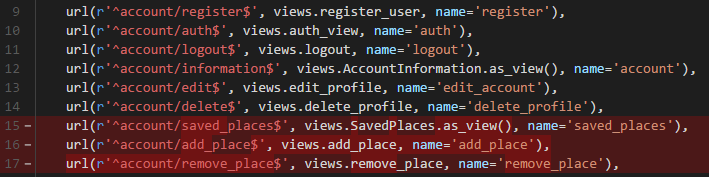
Design review – 21/10/2017

1. The project API must be RESTful.

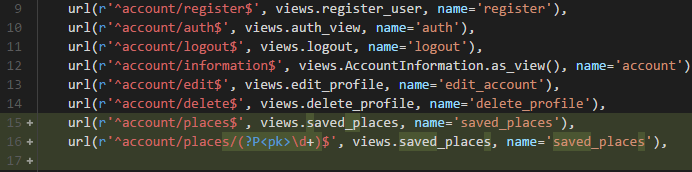
Even though our website does not provide REST API service, it is a best practice to adopt RESTful API design for our URL endpoints. The reason being that consistent url structure and methods as well as response structure make our website easy to maintain and more understandable.

Before:



So far, the project was strictly focused on functionality. There must be improvements to the server endpoint design. User’s saved places are accessed via “account/saved\_places”, added by sending POST request to “account/add\_place”, and removed by sending POST request to “account/remove\_place”.

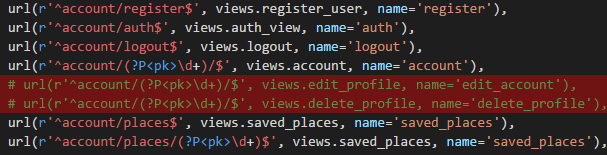
After:



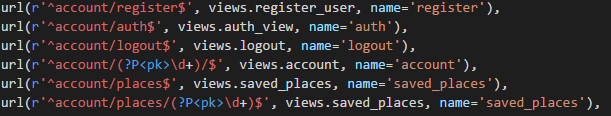
The end points were about “saved places” so it was changed to “account/places”. The naming of the URI was incorrect as the action must be determined by the HTTP methods not the URI. All requests other than GET were made to “account/place/{pk}”. Instead of using POST for removing the saved place, “DELETE” was used.

All relevant view methods were updated accordingly.

Before:

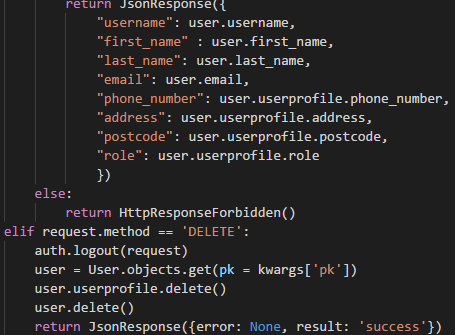


After:



Viewing, Editing and deleting account are done by sending GET, PUT and DELETE requests to “/account/{pk}” respectively. This is following the RESTful principles of using HTTP methods explicitly and using directory like URIs.

JSON format was used as the response where possible (if the client side code needs to parse the response body):

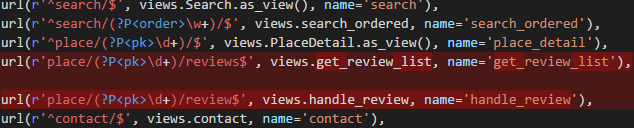


This ensures that all responses that needs to be processed on the client side code are consistent, reducing the number of programmer errors.

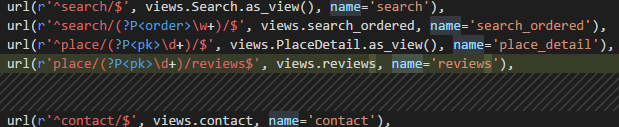
Design Review - 22/10/2017

1. Continuing to refactor the code

Before:



After:



Endpoints for place/pk/reviews and place/pk/review merged into place/pk/reviews and relevant view methods updated accordingly.

Before:



After:



Making sure URI has a directory like structure. It makes it consistent and generalized. I had to update href attributes for all anchor tags referencing these URIs.