

# Critique & Reflection for project 1 (Web Analytics)

Andrew Song (Github ID: andrew90)

From the user's perspective, the analytics site seems intuitive to use. The newcomers can click on the sign to obtain the Javascript snippet with a unique id. Users can take a look at the tracking data at deeper level, to the depth of finding out which visit was made when. Since I didn't want any unwanted malicious modifications on existing data, I left out the implementation for edit, delete, and update. One aspect that the users might not be fond of is that all the data are shown public, without any authentications and authorizations.

From the developer's perspective, it was good to see that the application tried to adhere to the "fat model & skinny controller" principle. However, the code itself could have been written more clearly if I were more familiar with the Ruby language. Other than the code and they syntax, there are also few points that could have been improved:

- 1) Several situations in which failures or exceptions could happen were not considered; what if the HTTP status codes were 4XX, 5XX instead of typical 200 OK that the code had expected? What if the user decided to put the Javascript snippet with same unique ID in several websites he owns without any knowledge of the trouble he might cause? (If he did, the URL shown on the web analytics site would actually keep on changing, messing up the results) There should have been some internal conditions to prevent such situations.
- 2) Writing specifications for classes and methods are still not clear. The specifications, as professor Jackson pointed out in the lecture, shouldn't be about the implementation details. I think I fell into the pit of trying to write specification for how the implementation works. This should definitely be improved for the upcoming projects.

The most successful aspect of the project would be that I managed to produce first fully-working functional web application with Ruby on Rails, Javascript, and HTML, the web languages I haven't used at all until 3 weeks ago. In the process, I was able to pick up basic Rails concepts that are important (MVC separation, associations, and etc ....) Although there are still lots of rooms to improve overall, I would like to congratulate myself on getting the first project done. I certainly hope that I could take on upcoming projects with more confidence and ease.

On the other hand, the least successful aspect was having built the application without understanding REST in complete sense. I might have programmed in RESTful way without really recognizing that I have done so. Routing and paths got sort of messed up. Moreover, the

unfamiliarity to the Ruby language has restrained my ideas of how the applications should work. From what I have learned in the class, the implementation should not hinder idea generation for the architecture.

The most valuable lesson I have learned from the project was the importance of thinking about the design and the architecture thoroughly before jumping into the actual implementation. I've always thought brainstorming and drawing out the object model diagram were tedious progress that could be removed from the process of app development. However, such naïve thought has caused problems here and there in the development stage. For instance, on realizing that I needed another model for the project, I needed to rewrite the code for almost the entire project, which costed me dearly in time and efficiency. This will certainly be improved in upcoming projects.