

Basic Use Cases:

- To-Do List
 - Each item has description, deadline, completion status and a tag/tags
 - CRUD operations for each item
- Tags
 - Each tag is associated with a to-do item
 - CRUD operations for each tag
 - Tags are associated to the to-do item when performing CRUD operations on the to-do item only
 - Clicking on tags to filter out relevant to-do items

Execution Plan:

Task	Status
Create Rails Application	Done
Configure Routing	Done
Create MVCs	Completed for to-do list In-Progress for tags
CRUD operations for to-do list	Done
CRUD operations for tags	In-Progress
Tag Filtering	Not-started
CSS stylings	Not-started
User Authentication (possibly)	Not-started
Deploy to Heroku	Not-started

Some Minor Changes:

- Change text-input to checkbox control for completion status
- Remove Task-Id
- Implement active record validation
- Change dateline column type to a ruby datetime object
- Try implementing tag auto-completion

Issues Faced

1. There is limited up-to-date tutorials that focus on implementing React with Ruby on Rails. It took me a while to piece together all the web-development concepts, such as MVC, routing, RESTful API etc. I still do not entirely grasp some of these concepts.
2. For now, I am stuck with getting and posting APIs when associative databases are involved (specifically, the 'posting' of tasks and its relevant associated tagging). I will also need to account for the fact that multiple tags can be created in an instance.