

CBWA

C.A. // Project - First Part

8th October - **Due by 29th October**

Bug Tracker (API)

Description	1
Examples/Inspiration	1
Functional Requirements:	2
Technical Requirements:	2
Submission	2
Details	3
Projects	3
Users	3
Issues/Comments	4
Structure	5

Description

You need to build an API for a bug tracker.

https://en.wikipedia.org/wiki/Bug_tracking_system

Examples/Inspiration

<https://github.com/davealbert/cbwa/issues>

<https://www.atlassian.com/software/jira>

<https://trello.com/>

Functional Requirements:

- Projects
 - Get all projects
 - Get individual projects
 - Add new projects individually
- Users
 - Get all users
 - Get individual users
 - Add new users individually
- Issues
 - Get all issues
 - Get all issues for a project
 - Get individual issues
 - **BONUS:** Updated the status of an issue
 - Add new issues to a project individually
 - Issues have Comments
 - Get all comments for an issues
 - Get individual comments for an issues
 - Add new comments to an issue

Technical Requirements:

- Use nodejs
- Use expressjs
- Use mongodb on Atlas
- Deployed on Heroku
- Code in Github

Submission

1. Link to GitHub Repository
2. Link to Heroku

Details

Projects

Projects

- `_id`
- `slug`
- `name`
- `description`

You should have at least 2 Projects (**bookstore** and **bugtracker**) added to your API's database on submission.

`_id`: Automatic from Mongo

`slug`: Capitals A-Z (like BOOK or BUG)

`name`: string

`description`: string

Users

Users

- `_id`
- `name`
- `email`
- `usertype` - (admin/user)

You should have at least 2 users (you and me) added to your API's database on submission.

`_id`: Automatic from Mongo

`name`: string

`email`: string

`usertype`: string

Issues/Comments

Issues

- `_id`
- `issueNumber`: `{project.slug + number}`
- `title`
- `description`
- `status`: `{open, wip, blocked, closed}`
- `project_id`: `{project._id}`

You should have at least 2 issues per project

Comments

- `comment_id`
- `text`
- `author { user._id }`

You should have at least 2 comments per issue

`_id`: Automatic from Mongo

`issueNumber`: The slug plus an automatic number based on the count of items at issue creation (BOOK-1, BOOK-2, BUG-1, BUG-2)

`title`: string

`description`: string

`status`: string { `todo`, `wip`, `blocked`, `closed` }

`Project_ID`: ObjectId of the project "**`_id`**" : **`ObjectId("5f742fe68a6f7864e83f5cc3")`**,

Comments: [

{

`_id`: Automatic from Mongo

`text`: string

`author`: ObjectId of the user "**`_id`**" : **`ObjectId("5f742fe6ca6f7f64e83f5573")`**,

]

Structure

