

Definition of Done

When a task is “done”. You should be prepared to answer the following questions about it.

- What does it do?
- What did you use?
- How long did it take?

Before a task is put into done it must first be.

1. Tested on your local for styling, compilation, and run-time errors.
2. Have reviewers ready to approve or offer feedback.
3. Incorporate feedback into the product or explain why the feedback is unnecessary.
4. Have pulled from master branch and your working branch and pushed after.
5. Solve merge conflicts, please do not change unless you are sure you need to.
6. Make sure you did not push any test files into master.

Review Guidelines: Before you ask for a review, make sure you have

1. Created a test, it can be anything as long as it's proof it works.
2. Applied the tests as a standalone and provide your tests as a commit.
3. Tested it doesn't require any prior function to be created
4. Conforms to the style guides specified below.

Tester questions: For all people who are submitting a review please address

1. What you think it does!
2. How you think they did it!
3. What you think could be added!!!
4. What you think could be changed!
5. Say thanks for your hard work. Seriously.

We know a task is done when all the above is complete and:

- Feature has been merged into master branch
- Master branch passes all CI tests

Priority: Standalone Versions, Handling Integration and Requests, Styling/CSS

1. CRUD Operations go first (Backend API Calls, Frontend Forms/Requests)
2. Advanced Database Logic(searching/Foreign Keys/Joining)
3. New tables (Users)

Please help each other out by calling the faults early on. Let's avoid more technical debt.

The Long Awaited Styling Guide.

Routing for Flask and React

Backend API routes: `api/<table>/<key>/<method>/<parameters>`

The (key) is optional, and is only used to query by IDs

See routes.py from Discord as an example (It's under files)

Frontend routes: `myurl.com/<feature>/<feature_dependency>/<feature_args>`

Language Styling Guide:

OTBS: One True Brace Style

```
“Just don’t do                                GoodFunction {
badFunction_everything                        ....
{                                              }
...
}                                              ”
```

Python: Pep8/flake8. **DO NOT MERGE IF IT IS NOT PEP8 Compliant.**

Javascript: snake_case_names, Class CapitalizeAlways

SQL: Only One (1) word as a name. two_words looks like this if needed.

Tests: `<Expectation><DescriptiveFunction><WhatYouAreDoingToTest>`

About Adding Features

All features will be separated into their own component files!

Under react/rentmyjunk/src/components

This also makes it less likely to have a merge conflict :)

DO NOT CODE BLOCKS IN APP.JS

Development Environment Folders

react/rmj_oss is all backend

react/rentmyjunk is all frontend

react/rentmyjunk/data is for testing .json files for frontend display and React Routes

Testing Style

Everyone must download Postman to test APIs, before approving any Backend commit.

Frontend will be tested by sending a message to the web console or writing to output.txt.

Make sure label it by putting `###` in the Comment above the line that is not integrated.