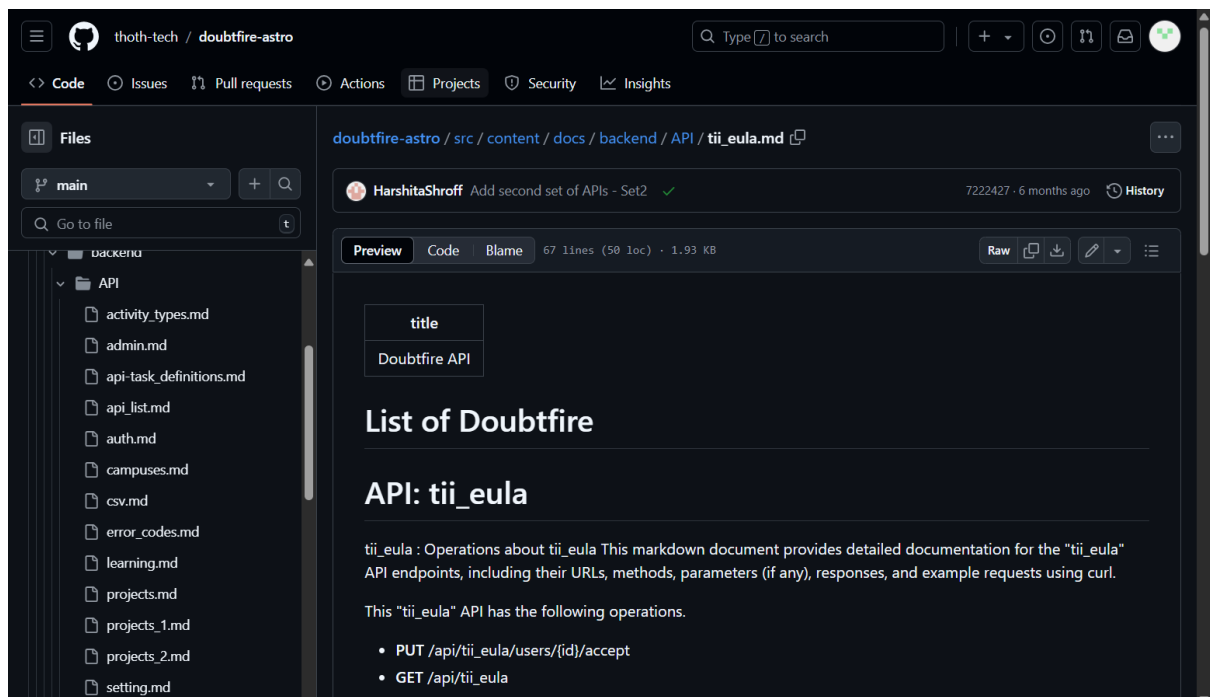


# Survey current progress and gaps in documentation - backend

## 1.Introduction

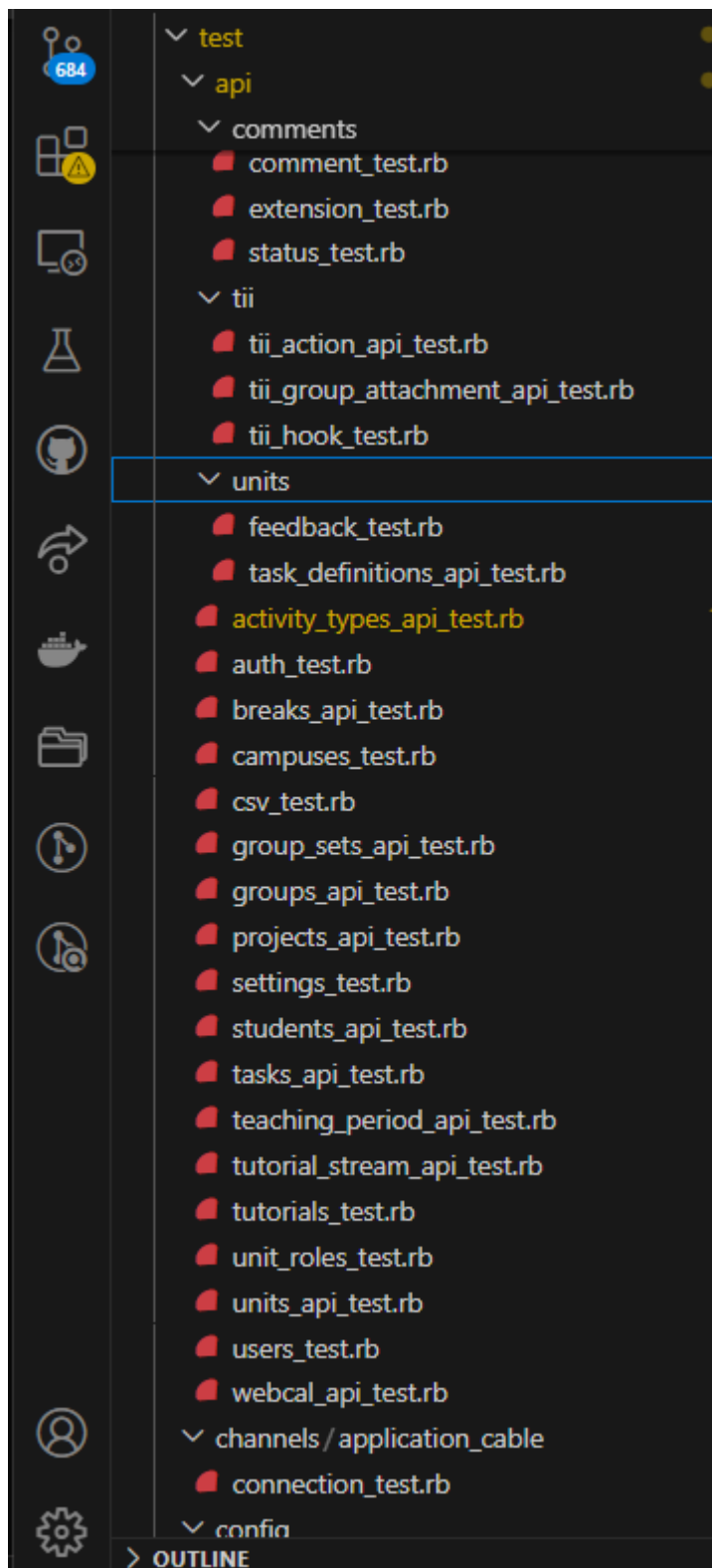
This document aims to look at the current state of backend documentation. It will check which files have documentation and which ones don't. It will also identify the test files for each backend component, helping to highlight areas that need more attention or improvement.

## 2. Sources



Here for the documentation, we, have been using the same repository which is doubtfire-astro

## 3. Backend Files currently in used



#### 4. Backend Files with documentation.

<b>Files with documentation</b>	<b>Corresponding Backend test files in use</b>
<b>Activity_types.md</b>	<b>Activity_types_api_test.rb</b>
<b>api-task_definitions.md</b>	<b>Task_definitions_api_test.rb</b>
<b>csv.md</b>	<b>Csv_test.rb</b>
<b>projects.md</b>	<b>Projects_api_test.rb</b>
<b>settings.md</b>	<b>Settings_test.rb</b>
<b>students.md</b>	<b>Students_api_test.rb</b>
<b>teaching_periods.md</b>	<b>Teaching_period_api_test.rb</b>
<b>tii_actions.md</b>	<b>Tii_action_api_test.rb</b>
<b>tii_hook.md</b>	<b>Tii_hook_test.rb</b>
<b>tutorials.md</b>	<b>Tutorials_test.rb</b>
<b>unit-group_sets.md</b>	<b>Group_sets_api_test.rb</b>
<b>unit-tutorial_streams.md</b>	<b>Tutorial_stream_api_test.rb</b>
<b>unit_roles.md</b>	<b>Unit_roles_test.rb</b>
<b>units.md</b>	<b>Units_api_test.rb</b>
<b>users.md</b>	<b>Users_test.rb</b>
<b>webcal.md</b>	<b>Webcal_api_test.rb</b>

These files have all their documentation, and they are linked to the right test files.

The documentation in these files is easy to understand the functionality and behavior of the backend components, and the test files ensures the components are properly validated.

#### 5. Documentation which does not matches corresponding files

<b>Api_list.md</b>
<b>Campuses.md</b>
<b>Error_codes.md</b>
<b>Learning.md</b>
<b>Projects_1.md</b>
<b>Projects_2.md</b>

<b>Stats.md</b>
<b>Submission.md</b>
<b>Tii_eula.md</b>
<b>Units_id.md</b>

## 6. Files which are missing documentation

<b>Comment_test.rb</b>
<b>Extension_test.rb</b>
<b>Status_test.rb</b>
<b>Til_group_attachment_api_test.rb</b>
<b>Feedback_test.rb</b>
<b>Auth_test.rb</b>
<b>Breaks_api_test.rb</b>
<b>Breaks_api_test.rb</b>
<b>Tasks_api_tests.rb</b>
<b>Connection_test.rb</b>
<b>Deakin_config_test.rb</b>

These files do not currently have documentation.

They are associated with various test files, including those for user authentication, feedback processing, group management, and configuration settings.

## 7. Gaps and Recommendations

### Missing Documentation

Admin.md and error\_codes.md lack corresponding documentation, but these files likely contain important details related to administrative settings and error handling in the backend.

Documentation should be created for these files to ensure clarity in understanding backend operations for system administrators and developers.

## 8. Test Coverage for Non-Documented Files

It is important to document these test cases to clarify their purpose and ensure that the testing processes are fully understood.

## 9. Standardization of Documentation

Some files such as projects.md and csv.md are partially documented, but more detailed descriptions of error handling and API logic should be added.

All backend documents should follow a standard format to maintain consistency. This documentation should provide a clear and simple explanation of the backend code, making it easy to understand.

## **10. Conclusion**

The survey of backend documentation has provided a clear picture of which files are well-documented, and which require further attention. While many files have corresponding test files, there are gaps that need to be addressed, particularly around the missing documentation for critical backend components. By filling in these gaps and ensuring consistency in documentation practices, the backend will be more maintainable, and future development will be more efficient.

## **11. Next Steps to consider**

### **Immediate Actions to take**

Begin with the documentation for the files without documentation, focusing on important backend components like admin settings, error codes, and configurations.

**Standardization:** create and develop a consistent documentation format that includes endpoints, error codes, and test cases for all backend files.

**Review and Update:** Implement a periodic review of the documentation to ensure it stays up to date with any changes to the backend system.