**Test plan**

**Trello Board**

# Introduction

## Objectives

Trello board is a web-based tool that brings all the tasks, teammates and tools together. Trello board keep tasks organized and work moving forward. The QA is responsible for testing the product and ensuring it meets their needs.

## 

# Scope

The scope of this Test plan is limited to CRUD operations on the Trello board and includes a set of both positive and negative test cases. Test cases are written at the Business layer and Presentation layer.

1. Create a manual test with as many steps as necessary
2. Save it
3. Retrieve it and have the ability to view it when running the test
4. Enter results and appropriate comments
5. View results

Automated tests are also part of this Test Plan and covers the CRUD operations for API and UI.

# Test Approach

Exploratory testing will play a large part of the testing as learning will be on the go.

## Unit and Integration Tests

Automated unit and integration tests are executed oon building the project.

## System Test

Functional tests are written and recorded in this document.

These tests are performed once the application is deployed on the Test Environment.

## Test Automation

Automated functional tests are part of this Test Plan.

# Test Environment

The application is deployed on the Test Environment to execute tests.

## Deliverables

|  |  |
| --- | --- |
| **Deliverable** | **Date / Milestone** |
| Test Plan | 10/7/2023 |
| Test Results | 10/7/2023 |
| Test Status report | 10/7/2023 |

**Trello Board Test Cases**

**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version # | Date | Author | Changes Summary | Review Comments Location |
| 1.0 | 10/07/2023 | Tester1 | Initial test case |  |

|  |  |  |
| --- | --- | --- |
| Number | Action | Comments |
| 1 | Fetch a board |  |
| 2 | Create a board |  |
| 3 | Update a board |  |
| 4 | Delete a board |  |

**Object:** Perform CRUD operations on a Trello board by creating a new board, fetching it, updating the board by editing the details, deleting a board.

**Testing type:** Functional testing

**Test cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Action | Expected result | | Test result |
| **1** | **Fetch a Board - API** | | | |
| **1.1** | **Get a board details using a valid board id and authorization** | | | |
|  | Pre steps:   1. A board is available and can be identified by a board id 2. A valid API key and token are available |  |  | |
| 1.1.1 | Steps:   1. Add the API key and token as query parameters 2. Add header to accept “application/json” |  |  | |
| 1.1.2 | Steps:   1. Fetch the endpoint, <https://api.trello.com/1/boards/>{id},   using the request verb “GET”.   1. Supply the id of the board in the path 2. Submit the request | Request is successful and the board details are displayed |  | |
| 1.1.3 | Steps:   1. Verify the status code and status line 2. Verify the response body | 1. Status code is 200  Status line is 200 OK  2. The response body is displayed and it contains the board id supplied. |  | |
| **1.2** | **Get a board details using a valid board id and invalid authorization** | | | |
| 1.2.1 | Pre steps  A board is available and can be identified by a board id  A valid API key is available  An invalid token is provided |  |  | |
| 1.2.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  |  | |
| 1.2.3 | Steps:  Fetch the endpoint, <https://api.trello.com/1/boards/>{id},  using the request verb “GET”.  Supply the id of the board in the path  Submit the request | Request is not successful and the board details are not returned |  | |
| 1.2.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body for the message | Status code is 401.  Status message is 401 Unauthorized.  Response body shows “invalid token” message. |  | |
| **1.3** | **Get a board details using an invalid board id and valid authorization** | | | |
| 1.3.1 | Pre steps  A valid API key and token is available  An invalid board id is provided |  |  | |
| 1.3.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  |  | |
| 1.3.3 | Steps:  Fetch the endpoint, <https://api.trello.com/1/boards/>{id},  using the request verb “GET”.  Supply the id of the board in the path  Submit the request | Request is not successful and the board details are not returned |  | |
| 1.3.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body for the message | Status code is 400.  Status message is 400 Bad Request.  Response body shows “invalid id” message. |  | |
| **1.4** | **Get a board details without a board id** | | | |
| 1.4.1 | Pre steps  A valid API key and token is available  A board id is not provided |  | |  |
| 1.4.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  | |  |
| 1.4.3 | Steps:  Fetch the endpoint, <https://api.trello.com/1/boards/>{id},  using the request verb “GET”.  The board id is not added to the path.  Submit the request | Request is not successful and the board details are not returned. | |  |
| 1.4.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body for the message | Status code is 404.  Status message is 400 Not Found. | |  |
| **1.5** | **Validate the schema of the Get request** |  | |  |
| 1.5.1 | Pre steps  A valid API key and token is available  A board id is provided |  | |  |
| 1.5.2 | Steps:  Add the API key and token as query parameters  Add the board id to the path  Add header to accept “application/json”  Save the schema as a JSON file |  | |  |
| 1.5.3 | Steps:  Fetch the endpoint, <https://api.trello.com/1/boards/>{id},  using the request verb “GET”.  The board id is added to the path.  Submit the request | Request is not successful and the response is verified. | |  |
| 1.5.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body to verify the response body adheres to the schema. | Status code is 200.  Status message is 200 OK. | |  |
| 1.5.5 | Assert the response body to verify the response body adheres to the schema. | The response body is validated according to the schema successfully. | |  |
| **2** | **Create a board** |  | |  |
| **2.1** | **Create a new board using valid credentials** |  | |  |
| 2.1.1 | Pre steps  A valid API key and token is available |  | |  |
| 2.1.2 | Steps:  Add the API key and token as query parameters  Add the name of the board as query parameter “name”  Add header to accept “application/json” |  | |  |
| 2.1.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/,  using the request verb “POST”.  Submit the request | The request is successful and a new board is created by the name provided. | |  |
| 2.1.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body for the id and name. | Status code is 200.  Status message is 200 OK.  Response body has a unique id and the name is the same as the name provided in the request. | |  |
| **2.2** | **Create a new board with invalid credentials** | | | |
| 2.2.1 | Pre steps  A valid API key is provided  An invalid token is provided |  | |  |
| 2.2.2 | Steps:  Add the API key and token as query parameters  Add the name of the board as query parameter “name”  Add header to accept “application/json” |  | |  |
| 2.2.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/,  using the request verb “POST”.  Submit the request | The request is NOT successful and a new board is not created. | |  |
| 2.2.4 | Steps:  Verify the status code and status line  Verify the response body | Status code is 401.  Status message is 401 Unauthorized.  Response body shows “invalid token” message. | |  |
| **2.3** | **Create a new board without a name** |  | |  |
| 2.3.1 | Pre steps  A valid API key and token are provided  A board name is not provided |  | |  |
| 2.3.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  | |  |
| 2.3.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/,  using the request verb “POST”.  Submit the request | The request is NOT successful and a new board is not created. | |  |
| 2.3.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body for message and error. | Status code is 400.  Status message is 400 Bad request.  Response body shows “invalid value for name” message. | |  |
| **3** | **Update a board** |  | |  |
| **3.1** | **Update a board details valid credentials** |  | |  |
| 3.1.1 | Pre steps  A valid API key and token is available  The board id is added to the path  The parameters to be updated are added as query parameters. |  | |  |
| 3.1.2 | Steps:  Add the API key and token as query parameters  Add the “closed” parameter as query parameter.  Add header to accept “application/json” |  | |  |
| 3.1.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/{id},  using the request verb “PUT”.  Submit the request | The request is successful and the board is closed. | |  |
| 3.1.4 | Steps:  Verify the status code and status line  Verify the response body  Assert the response body for the id and name. | Status code is 200.  Status message is 200 OK.  Response body is is same as the id provided. The value of the ‘closed’ tag is true. | |  |
| **3.2** | **Update a board details with invalid credentials** |  | |  |
| 3.2.1 | Pre steps  A valid API key is available.  An invalid token is provided.  The board id is added to the path.  The parameters to be updated are added as query parameters. |  | |  |
| 3.2.2 | Steps:  Add the API key and token as query parameters  Add the “closed” parameter as query parameter.  Add header to accept “application/json” |  | |  |
| 3.2.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/{id},  using the request verb “PUT”.  Submit the request | The request is NOT successful and the board is NOT closed. | |  |
| 3.2.4 | Steps:  Verify the status code and status line  Verify the response body | Status code is 401.  Status message is 401 Unauthorized.  Response body shows “invalid token” message. | |  |
| **3.3** | **Update a board details without a board id** |  | |  |
| 3.3.1 | Pre steps  A valid API key and token is provided.  The board id is NOT added to the path.  The parameters to be updated are added as query parameters. |  | |  |
| 3.3.1 | Steps:  Add the API key and token as query parameters  Add the “closed” parameter as query parameter.  Add header to accept “application/json” |  | |  |
| 3.3.2 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/{id},  using the request verb “PUT”.  Submit the request | The request is NOT successful and the board is NOT closed. | |  |
| 3.3.3 | Steps:  Verify the status code and status line | Status code is 404.  Status message is 404 Not found. | |  |
| **4** | **Delete a board** |  | |  |
| **4.1** | **Delete a board using valid credentials** |  | |  |
| 3.1.1 | Pre steps  A valid API key and token is available  The board id is added to the path |  | |  |
| 3.1.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  | |  |
| 3.1.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/{id},  using the request verb “DELETE”.  Submit the request | The request is successful and the board is deleted. | |  |
| 3.1.4 | Steps:  Verify the status code and status line  Verify the response body | Status code is 200.  Status message is 200 OK.  Response body contains null. | |  |
| **4.2** | **Delete a board with invalid credentials** |  | |  |
| 4.2.1 | Pre steps  A valid API key is available.  An invalid token is provided.  The board id is added to the path. |  | |  |
| 4.2.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  | |  |
| 4.2.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/{id},  using the request verb “DELETE”.  Submit the request | The request is NOT successful and the board is NOT deleted. | |  |
| 4.2.4 | Steps:  Verify the status code and status line  Verify the response body | Status code is 401.  Status message is 401 Unauthorized.  Response body shows “invalid token” message. | |  |
| **4.3** | **Delete a board without a board id** |  | |  |
| 4.3.1 | Pre steps  A valid API key and token is provided.  The board id is NOT added to the path. |  | |  |
| 4.3.2 | Steps:  Add the API key and token as query parameters  Add header to accept “application/json” |  | |  |
| 4.3.3 | Steps:  Fetch the endpoint, https://api.trello.com/1/boards/{id},  using the request verb “DELETE”.  Submit the request | The request is NOT successful and the board is NOT deleted. | |  |
| 4.3.4 | Steps:  Verify the status code and status line | Status code is 404.  Status message is 404 Not found. | |  |
|  |  |  | |  |
|  | **UI Test Scenarios** |  | |  |
| 1 | **Login** |  | |  |
| **1.1** | **Login using valid credentials** | | | |
| 1.1.1 | Pre steps  An account is created on the Trello board and have valid username and password. |  | |  |
| 1.1.2 | Steps  Navigate to the ‘<https://trello.com/login>’ page  Enter the valid Email and click on the ‘Continue’ button.  Enter the valid password and login |  | |  |
| 1.1.3 | Verify whether login is successful | The user login is successful and the boards page is opened for the user. | |  |
| **1.2** | **Login using invalid credentials** |  | |  |
| 1.2.1 | Pre steps  An account is created on the Trello board |  | |  |
| 1.2.2 | Steps  Navigate to the ‘<https://trello.com/login>’ page  Enter the Email and click on the ‘Continue’ button.  Enter a wrong password and login |  | |  |
| 1.2.3 | Verify whether login is successful | The user login is not successful. | |  |
| **2** | **Create a new board** |  | |  |
| **2.1** | **Create a new board from the workspace** |  | |  |
| 2.1.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user. |  | |  |
| 2.1.2 | Steps  On the ‘YOUR WORKSPACES’ section, the workspace is displayed for the user. |  | |  |
| 2.1.3 | Steps  Click the ‘Create new board’ option. | A ‘create board’ dialog box opens. | |  |
| 2.1.4 | Steps  Leave the Background to the default  Enter the title on the ‘Board title’ input.  Keep the ‘Visibilty’ to ‘Workspace’  Click on the ‘Create’ button. | Create button is enabled only on entering the title | |  |
| 2.1.5 | Verify whether the board is created | A new board is created successfully and gets navigated to the created board page.  The URL and the title section shows the board name.  The title section shows the ‘Workspace visible’ as the visibilty.  The lists, ‘To Do’, ‘Doing’ and ‘Done’ are displayed on the board.  The board name is also shown under ‘Your boards’ section on the board page. | |  |
| **2.2** | **Create a new board from the workspace without a title** |  | |  |
| 2.2.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user. |  | |  |
| 2.2.2 | Steps  On the ‘YOUR WORKSPACES’ section, the workspace is displayed for the user. |  | |  |
| 2.2.3 | Steps  Click the ‘Create new board’ option. | A ‘create board’ dialog box opens. | |  |
| 2.2.4 | Steps  Leave the Background to the default  Leave the ‘Board title’ as empty | Pop-up shows an error message  ‘👋  Board title is required  ’ | |  |
| 2.2.5 | Verify whether the board is created | The ‘Create’ button is disabled and a new board is not allowed to create. | |  |
| **2.3** | **Create a new board from the workspace with a selected background and private visibility** |  | |  |
| 2.3.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user. |  | |  |
| 2.3.2 | Steps  Select a background for the board  Enter the title on the ‘Board title’ input.  Select the visibility as ‘Private’  Click on the ‘Create’ button. | A new board is created with the background given.  The title section shows the board name and the visibility as ‘Private’. | |  |
| **3** | **Update a board** |  | |  |
| **3.1** | **Update a board by adding a card** |  | |  |
| 3.1.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section. |  | |  |
| 3.1.2 | Steps  Click the board that was created before | The board opens | |  |
| 3.1.3 | Steps  Add a new card to the list  Click the ‘Add a card’ option of the ‘To Do’ list. | The default cards ‘To Do’, ‘Doing’ and ‘Done’ are displayed on the board.  An input text-box appears with a text ‘Enter a title for this card...’ | |  |
| 3.1.4 | Steps  Enter the card title and click on the ‘Add card’ button | The card is added successfully and is visible under the ‘To Do’ list. | |  |
| **3.2** | **Update a board by adding a list** |  | |  |
| 3.2.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section. |  | |  |
| 3.2.2 | Steps  Click the board that was created before | The board opens | |  |
| 3.2.3 | Steps  If default lists are not available, click on the ‘Add a list’ option | An input text box appears showing ‘Enter list title...’ | |  |
| 3.2.4 | Steps  Enter the list title and click on the ‘Add list’ button | The list is added successfully and is shown on the board | |  |
| 3.2.5 | Steps  Click the ‘Add a card’ option of the created list.  Enter the title of the card and click on the ‘Add card’ button | A new card is added to the created list. | |  |
| **3.3** | **Update a board by adding a watcher** |  | |  |
| 3.3.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section.  A list is available on the board |  | |  |
| 3.3.2 | Steps  Click the additional options menu on the list | The ‘List actions’ menu opens | |  |
| 3.3.3 | Steps  Select the ‘Watch’ option from the list actions. | A check mark is added to the ‘Watch’ list action on the menu.    The symbol is added to the list. | |  |
| **3.4** | **Update a board by copying a list** |  | |  |
| 3.4.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section.  A list is available on the board |  | |  |
| 3.4.2 | Steps  Click the additional options menu on the list | The ‘List actions’ menu opens | |  |
| 3.4.3 | Steps  Select the ‘Copy list...’ option from the list actions. | A ‘copy list’ dialog box opens with input text to enter a new name and the ‘Create list’ button | |  |
| 3.4.4 | Steps  Choose a new name for the list and click on the ‘Create list’ button | The list is copied to the board with the chosen name. | |  |
| **4** | **Delete a board** |  | |  |
| **4.1** | **Close and delete a board** |  | |  |
| 4.1.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section. |  | |  |
| 4.1.2 | Steps  Click the board that was created before | The board opens | |  |
| 4.1.3 | Steps  Click the additional options menu displayed on the top right. | The menu opens | |  |
| 4.1.4 | Steps  Select the ‘… More’ option from the menu | The ‘More’ menu opens | |  |
| 4.1.5 | Steps  Click the ‘Close board…’ option | A ‘Close board?’ dialog box appears. | |  |
| 4.1.6 | Steps:  Click the ‘Close’ button | A dialog box appears showing the board is closed. | |  |
| 4.1.7 | Steps:  Click the ‘Permanently delete board’ option | A ‘Delete board?’ dialog box appears with a ‘Delete’ button. | |  |
| 4.1.8 | Click the ‘Delete’ button | The board is deleted successfully and board is not visible on ‘Your workspaces’. | |  |
| **4.2** | **Close and delete a board from ‘Your boards’** |  | |  |
| 4.2.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section. |  | |  |
| 4.2.2 | Steps  Click the board that was created before | The board opens | |  |
| 4.2.3 | Steps  Click the additional options menu displayed next to the board name shown under ‘Your boards’ | A menu opens with the board name as the title and a ‘Close board’ option. | |  |
| 4.2.4 | Steps  Click the ‘Close board’ option on the menu opened | A ‘Close board?’ dialog box opens with the ‘Close’ button | |  |
| 4.2.5 | Steps  Click the ‘Close’ button | A dialog box appears showing the board is closed. | |  |
| 4.2.6 | Steps:  Click the ‘Permanently delete board’ option | A ‘Delete board?’ dialog box appears with a ‘Delete’ button. | |  |
| 4.2.7 | Steps:  Click the ‘Delete’ button | The board is deleted successfully and board is not visible on ‘Your workspaces’. | |  |
| **4.3** | **Reopen a closed board** |  | |  |
| 4.3.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section. |  | |  |
| 4.3.2 | Steps  Click the board that was created before | The board opens | |  |
| 4.3.3 | Steps  Click the additional options menu displayed on the top right. | The menu opens | |  |
| 4.3.4 | Steps  Select the ‘… More’ option from the menu | The ‘More’ menu opens | |  |
| 4.3.5 | Steps  Click the ‘Close board…’ option | A ‘Close board?’ dialog box appears. | |  |
| 4.3.6 | Steps:  Click the ‘Close’ button | A dialog box appears showing the board is closed. Also a ‘Reopen board’ button is shown | |  |
| 4.3.7 | Steps:  Click the ‘  Reopen board’ option | The ‘Select a workspace’ dialog opens. | |  |
| 4.3.8 | Choose the workspace and click on ‘Reopen board’ | The board is reopend and opens up.  The board is visible under ‘Your workspaces’. | |  |
| **4.4** | **Archive list and cards of a board** |  | |  |
| 4.4.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section. |  | |  |
| 4.3.2 | Steps  Click the board that was created before | The board opens | |  |
| 4.3.3 | Steps  Add cards tot the ‘To Do’ list | Cards are added | |  |
| 4.3.4 | Steps  From the additional options menu of the list, select the ‘Archive all cards in the list…’ option | A dialog box appears with the ‘Archive all’ button | |  |
| 4.3.5 | Steps  Click the ‘Archive all’ button | All cards are removed from the list successfully | |  |
| 4.3.6 | Steps:  From the additional options menu, click the ‘Archive this list’ option. | The list is removed successfully | |  |
| **5** | **Open Card and Add details** |  | |  |
| **5.1** | **Open a card and edit it** |  | |  |
| 5.1.1 | Pre steps  The user is logged in and the boards page is opened.  A workspace is created and available for the user.  A board is created and shown under the ‘Your workspaces’ section.  A ‘To Do’ list is available and a card ‘Implement CI/CD pipeline’ is added |  | |  |
| 5.1.2 | Steps  Click the board that was created before | The board opens | |  |
| 5.1.3 | Steps  Click the button next to the card. | A list of options appears:  Open card  Edit labels  change members  Change cover  Move  Copy  Edit dates  Archive | |  |
| 5.1.4 | Steps  Click the ‘Open card’ option  Add description for the card and click on the ‘Save’ button | The card opens with the title showing the card name and the list it belongs to.  The description is saved and is shown on the card details | |  |
| 5.1.5 | Steps  Click on the ‘Members’ option from the ‘Add to card’ menu.  Select the member from the ‘Board members’. | The board member is added and the log is added under the ‘Activity’. | |  |
| 5.1.6 | Verify whether the card details are updated | The card details are updated successfully. | |  |