Test Design Requirements: UI & API

Project: Test Project

Introduction

This document should include necessary information so that a QA can start testing Item Management features.

Documentation and References

- Todoist API spec: https://developer.todoist.com/sync/v8/#items
- Cypress doc: https://docs.cypress.io/

Testing Scope

- Features to cover:
    1. Todo Item Management feature:
        - Create todo item: only test with todo item name and due date
            - Item name accepts plain text
            - Item name accepts markdown formatted text
            - Item name accepts containing hyperlink
        - Update todo item
            - Rename todo item
            - Mark todo item complete

- Only happy cases are covered
- Verification extent is minimal

Out-of-scope Testing

For only happy cases are covered, following points are out of scope:

- Negative tests
- Skip irrelevant verifications
- For UI testing, do not test with sub-project. Instead, “Inbox” project is used
- For API testing, only test with Async API
- Do not test with historical todo items

Test Limitation

- For API testing, calling too many requests in a specific duration might cause a Limits Reached issue and cause a test run to fail expectedly. For more details, see https://developer.todoist.com/sync/v8/#request-limits

Test Strategy

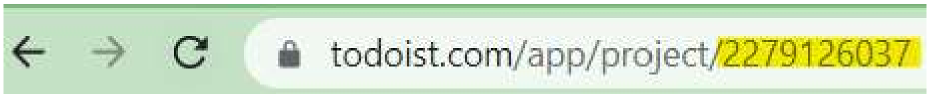
- For API testing, because requests are async, after the main request gets responded, make another request to get item details. Use the details to verify if the api works correctly
- For UI testing, after an action of creation/update has been done, make 2 verifications:
    - Verify the creating/update result is reflected on UI
    - Verify the item is persistent after browser reloaded

Test Approach

Both UI & API testing can apply automation. Cypress is chosen to be the test automation framework/tool as it supports Chrome (also Chromium-based browsers) and Firefox. Cypress also supports making http requests that can be used to automate API tests. Cypress also includes a test framework inside itself and can integrate into a CI/CD workflow. For more details, read https://docs.cypress.io/guides/overview/why-cypress.

Test Setup

- Todoist API token: https://developer.todoist.com/sync/v8/#getting-started
- Project ID from browser: browse to project then get it from browser address bar



todoist.com/app/project/2279126037

Test Plan

Test name	Precondition	Test Steps	Data Set	Priority
API				
Create new todo item		- Make add item request with name and due date - Verify the response code is 200 - Verify the sync_status field is “ok” - Make get item info request	1. **content** is plain text; **due date** is any valid value 2. **content** is markdown formatted; **due date** is any valid value 3. **content** contains hyperlink; **due date** is any valid value 4. **content** is any valid value; **due date** is full date	

		- Verify the response code is 200 - Verify **content** and **due** fields are correct	5. **content** is any valid value; **due date** is floating date with time 6. **content** is any valid value; **due dat**e is with time and timezone	
Rename todo item	Create a new todo item. See add item request	- Make update request with new name - Verify the response code is 200 - Verify the **sync_status** field is “ok” - Make get item info request - Verify the response code is 200 - Verify **content** field is the new name	1. **content** is plain text 2. **content** is markdown formatted 3. **content** contains hyperlink	
Mark todo item complete	Create a new todo item. See add item request	- Make complete item request with new name - Verify the response code is 200 - Verify the **sync_status** field is “ok” - Make get item info request - Verify the response code is 200 - Verify **checked** field is 1	1. **content** is any valid value	

UI				
Create new todo item	- Login - Go to Inbox	- Click Add task button - Verify Add task panel appears - Enter item name - Select due date - Click Add task button - Verify the tasked is added in todo list with correct content and due date - Reload browser - Verify the created task with name and due date is still there	1. **content** is plain text; **due date** is any valid value 2. **content** is markdown formatted; **due date** is any valid value 3. **content** contains hyperlink; **due date** is any valid value 4. **content** is any valid value; **due date** is full date 5. **content** is any valid value; **due date** is floating date with time 6. **content** is any valid value; **due dat**e is with time and timezone	
Rename todo item	- Login - Go to Inbox - Create new todo item	- Hover the todo item - Click Edit button - Verify Edit task panel appears - Enter new name - Click Save button - Verify the task has new name in todo list - Reload browser - Verify the task's new name still	1. **content** is plain text 2. **content** is markdown formatted 3. **content** contains hyperlink	

		remains the new value		
Mark todo item complete	- Login - Go to Inbox - Create new todo item	- Check the radio button for the item - Verify the item is no longer shown in todo list - Verify the toast shows “1 task completed”		