Test Design Requirements: Google Calendar integration

Introduction

This document should include necessary information so that a QA can start testing Google Calendar integration feature.

Documentation and References

- Todoist - Google calendar integration doc:
  https://todoist.com/help/articles/use-google-calendar-with-todoist
- Google Calendar API doc: https://developers.google.com/calendar/api/v3/reference
- Cypress doc: https://docs.cypress.io/

Testing Scope

- Todoist can integrate into Google Calendar
- Ability to choose either all Todoist projects or a specific project to sync
- Todoist can stop integrate into Google Calendar
- Tasks created in Todoist are synced in Google Calendar and vice versa (2-way sync)
- Updates on tasks on either Todoist side or Google Calendar are reflected on the other side
- Integration options
- Ability to manually sync from Todoist

Out-of-scope Testing

N/A

Test Limitation

Limitations if any to be described in Test Approach section

Test Strategy

- Use 3 Todoist accounts if possible to reduce the effort of testing synced projects options: one All projects is chosen, you have to remove the integration before setting it to a specific project.
- Focus on how changes on sync options are reflected on both Todoist & Google Calendar side: Task content, task status, task content format, sync speed, etc.
- Also check what happens if Google Calendar is configured to ignore an integration

Test Approach

Both manual and automation testing can be applied for testing this feature. There are some pros and cons with each of approach:

- Manual testing:
    - Pros:
        - No need to work with Google Calendar API, Google API Console
        - Setup: Integration process from Todoist UI is straightforward
        - Test steps and verification is easy to be done
        - Cover more detailed tests. If automation covers all, the risk of test failing caused by calling Google APIs is high.
    - Cons:
        - Testing effort is high as there are so many test cases
- Automation testing:
    - Pros:
        - Reduce testing effort; produce quick test reports
        - Can integrated into CI/CD
    - Cons:
        - To get data from Google Calendar for test verification, it's required to have knowledge of Google API Console, Google API in general and Google Calendar API in specific.
        - Google Calendar is an external system, therefore, it's risky to some certain extent that there might be some errors by the time automated tests are running. Test monitoring and analysis effort are required.
        - A test will mix up steps of UI interaction and API call. In the process, the test plan for manual testing is not applicable for automation.

If automation is applied, following points should be considered:
- Rule of thumb: Consider between manual and automation testing, which is better to choose
- 3 pairs of Todoist and Google accounts should be used. One for all projects sync, one for a specific project sync and one, if chosen to be automated, for **testing integration removal**. Removing then integrating again involves test precondition and postcondition. One of them if fails will get not only the current test but also its next tests fail. Using each pair will prevent changes on this account not to affect that account.
- Integration remove feature should not be automated as it is supposed not to be run frequently when it has been confirmed working at least once. This test can be done manually quickly but for automation, again, it requires test precondition and postcondition.
- UI interactions regarding Google Calendar should not be automated as UI elements on Google Calendar can be changed according to calendar views or by the development team on their side. Use Google Calendar API for fetching data that will be used for verification.

The above automation approach involves UI interactions. There is another automation approach that will use only APIs (Both Google Calendar API and Todoist API). To apply this approach, 3 pairs of accounts should be already set up. One they are properly set up, steps are to work with API calls only.

- Pros:
    - Less failures from UI issues such as server availability
    - Faster test run
    - Simpler test development for the team only focus on dealing with API stuff
- Cons:
    - Requires manual proper setup at the beginning (actually this might be an advantage because it may be done once)
    - Requires manual effort for UI-involved tests

High Level Test Plan

Integrate with Google Calendar

- Verify a Todoist account that has never integrated to Google Calendar can integrate to Google Calendar with various options
- Verify a Todoist account that has removed its integration to Google Calendar can re-integrate to Google Calendar
- Verify a Todoist account has integrated to a Google Calendar keeps integration status after reload browsers

Sync options

- Verify after integrated, tasks created in Todoist are displayed on the integrated Google Account under Todoist label:
    - Verify every task is shown in Google calendar if All projects was selected
    - Verify only tasks of a specific project is shown in Google calendar if that project was selected
    - Verify full date tasks are displayed or not on Google calendar depending on "Todoist tasks without due time" option
        - Verify full date tasks on Todoist are also displayed as full date on Google calendar
    - Verify duration of a task with specific due date on Todoist are displayed on Google calendar accordingly to "Google event duration for new Todoist tasks" option
    - Verify completed tasks are displayed or not on Google calendar depending on "Completed Todoist tasks" option
- Verify when user chooses to sync from Todoist, Google calendar is synced accordingly
- Verify when there is an update on Google calendar, it's reflected on Todoist in real-time manner and vice versa

Integration removal

- Verify an integration can be removed from Integrations setting
- Verify after the integration is removed, there is no longer sync between Todoist and Google calendar

Appendix

Google API analysis

- In order to start using Google API, request authorization must be set up. See
https://developers.google.com/calendar/api/guides/auth
- Google Calendar API should be enabled

- Use endpoint https://developers.google.com/calendar/api/v3/reference/calendarList/list to get your Todoist calendar

200   ✕

  },
  {
    "kind": "calendar#calendarListEntry",
    "etag": "\"16~~~~~~~~~~\"",
    "id": "~~~~~~~~~~~~~~beinqb5hvrk@group.calendar
    "summary": "Todoist",
    "timeZone": "Asia/Ho_Chi_Minh",
    "colorId": "22",
    "backgroundColor": "#f691b2",
    "foregroundColor": "#000000",
    "selected": true,
    "accessRole": "owner",
    "defaultReminders": [],
    "conferenceProperties": {
      "allowedConferenceSolutionTypes": [
        "hangoutsMeet"
      ]
    }

- Use endpoint https://developers.google.com/calendar/api/v3/reference/events/list to fetch events. Note the "organizer" field in each item to filter Todoist synced tasks