

ClickUp Plugin Challenge - Project Documentation

1. Project Overview

This project is a Python-based ClickUp Plugin Challenge implementation. It contains modular plugins that interact with the ClickUp API for performing task management operations such as creating tasks, updating tasks, searching tasks, deleting tasks, and more. The structure is organized for scalability and ease of testing.

2. Project Folder Structure

The following is the folder structure of the project:

```
plugin/
├── bundles/
│   ├── clickup/
│   │   ├── bundle_dependency.py    # Defines shared PluginInput, Output, and Credentials models
│   │   ├── config.py              # Configuration constants (e.g., proxy)
│   │   ├── __init__.py            # Marks directory as a Python package
│   │   ├── shared/
│   │   │   ├── __init__.py
│   │   │   └── api_client.py      # Handles HTTP requests to ClickUp API (async with aiohttp)
│   │   ├── resources/
│   │   │   ├── i18n/
│   │   │   │   ├── en.yml         # Translations / localization support
│   │   │   │   ├── bundle_schema.yml # Defines schema for all plugins
│   │   │   │   └── icon.png       # Bundle icon
│   │   └── plugins/
│   │       ├── create_task/       # Plugin: Create new tasks
│   │       │   ├── plugin.py      # Logic to call ClickUp Create Task API
│   │       │   ├── plugin_schema.yml # Schema: input/output definition
│   │       │   └── test_create_task.py # Test script for plugin
│   │       ├── update_task/       # Plugin: Update task details
│   │       │   ├── plugin.py
│   │       │   ├── plugin_schema.yml
│   │       │   └── test_update_task.py
│   │       ├── update_task_status/ # Plugin: Update only task status
│   │       │   ├── plugin.py
│   │       │   ├── plugin_schema.yml
│   │       │   └── test_update_task_status.py
```

```

|—— delete_task/      # Plugin: Delete tasks
|   |—— plugin.py
|   |—— plugin_schema.yml
|   |—— test_delete_task.py
|—— get_task/         # Plugin: Retrieve a single task
|   |—— plugin.py
|   |—— plugin_schema.yml
|   |—— test_get_task.py
|—— search_tasks/     # Plugin: Search tasks with filters
|   |—— plugin.py
|   |—— plugin_schema.yml
|   |—— test_search_task.py
|—— create_list/      # Plugin: Create lists inside spaces/folders
|   |—— plugin.py
|   |—— plugin_schema.yml
|   |—— test_create_list.py
|—— add_task_comment/ # Plugin: Add comments to tasks
|   |—— plugin.py
|   |—— plugin_schema.yml
|   |—— test_add_task_comment.py

```

3. Implemented Plugins & Tasks

- Create Task

Creates a new task in a specified list with name, description, status, and priority.

- Update Task

Updates details of an existing task such as name, description, status, and priority.

- Update Task Status

Updates only the status of a task (e.g., ☒ resolved, ☐ new issue).

- Delete Task

Deletes a task by task ID. Returns confirmation of deletion.

- Get Task

Retrieves details of a single task including ID, name, status, list_id, and timestamps.

- Search Tasks

Searches tasks across a team with filters like query, list_id, status, and assignee.

- Create List

Creates a new list inside a space or folder for task organization.

- Add Task Comment

Adds a comment to a specific task by task ID.

4. Workflow Example

1. Create a new task using the ``create_task`` plugin.
2. Retrieve task details with ``get_task``.
3. Update the task's status using ``update_task_status``.
4. Add comments using ``add_task_comment``.
5. Search for tasks with ``search_tasks`` to confirm updates.
6. Delete the task with ``delete_task`` if no longer needed.

5. Future Enhancements

- Support for subtasks creation and management
- Attachment handling (files, links)
- Bulk task operations
- More advanced search filters