Trello CLI

This CLI program allows you to add a card to a specified Trello board list with labels and a comment. It uses the Trello API to perform the operations. The project can be found on [Github](#).

Prerequisites

- Python 3.x
- `requests` library (`pip install requests`)
- Trello API key and token

Usage

Example CLI Command

```
python trello_cli.py --api_key YOUR_API_KEY --token YOUR_TOKEN --board_name "Board
Name" --list_name "List Name" --card_name "Card Name" --labels "Label1,Label2" --
comment "This is a comment"
```

Arguments

- `--api_key` :

    - **Description**: Your Trello API Key.
    - **Required**: Yes

- `--token` :

    - **Description**: Your Trello Token.
    - **Required**: Yes

- `--board_name` :

    - **Description**: The name of the Trello board where the card will be added.
    - **Required**: Yes

- `--list_name` :

    - **Description**: The name of the list within the board where the card will
      be added.
    - **Required**: Yes

- `--card_name` :

    - **Description**: The name of the card to be added.
    - **Required**: Yes

- `--labels` :

    - **Description**: (Optional) Comma-separated list of label names to be added
      to the card.
    - **Required**: No

- `--comment` :

    - **Description**: (Optional) A comment to be added to the card.

- **Required**: No

Setup Instructions

1. Clone the repository:

   ```
   git clone https://github.com/username/trello-cli.git
   cd trello-cli
   ```

2. Install the required Python package:

   ```
   pip install requests
   ```

3. Run the CLI program using the example command provided above.

Next Development Steps

- Update and Delete Cards: Add functionality to update and delete cards.
- Error Handling: Improve error handling and user feedback for better user experience.
- Unit Tests: Implement unit tests to ensure code quality and reliability.
- Additional Trello Actions: Add support for more Trello actions such as adding due dates, and assigning members.

Time Estimate

- Initial setup and coding: 1 hour
- Testing and debugging: 40 minutes
- Writing documentation: 20 minutes
- Total time spent: 2 hours