

Python Script: GitHub Repository Management

This Python script interacts with the GitHub API to help manage your repositories by identifying those that haven't been modified in the last 30 days. It is particularly useful for keeping track of inactive or "stale" repositories and organizing your GitHub account effectively.

Python Script

```
import requests
from datetime import datetime, timedelta

# Your GitHub Personal Access Token
TOKEN = 'your_github_token_here'

# GitHub API endpoint to list repositories
API_URL = "https://api.github.com/user/repos"

# Number of days to consider a repository "stale"
DAYS_THRESHOLD = 30

def get_stale_repositories():
    """
    Fetches all GitHub repositories for the authenticated user and checks
    which repositories have not been modified in the last specified number of days.
    """
    headers = {
        "Authorization": f"token {TOKEN}",
        "Accept": "application/vnd.github.v3+json"
    }

    response = requests.get(API_URL, headers=headers)

    if response.status_code != 200:
        print("Failed to fetch repositories:", response.json())
        return []

    repositories = response.json()
    stale_repos = []
    threshold_date = datetime.now() - timedelta(days=DAYS_THRESHOLD)
```

```

for repo in repositories:
    # Parse the last modified date of the repository
    last_modified_date = datetime.strptime(repo['updated_at'], "%Y-%m-%dT%H:%M:%SZ")

    # Check if the repository is stale
    if last_modified_date < threshold_date:
        stale_repos.append(repo['name'])

return stale_repos

def main():
    print("Checking for stale repositories...")
    stale_repos = get_stale_repositories()

    if stale_repos:
        print("The following repositories have not been updated in the last",
              DAYS_THRESHOLD, "days:")
        for repo in stale_repos:
            print(f"- {repo}")
    else:
        print("All repositories are up to date!")

if __name__ == "__main__":
    main()

```

Explanation

- **Authentication:** The script uses a GitHub Personal Access Token (PAT) for authentication. Make sure to replace `your_github_token_here` with your actual GitHub token.
- **Fetching Repositories:** It makes an API request to fetch the user's repositories using the GitHub API.
- **Filtering Stale Repositories:** The script calculates a threshold date based on the current date minus the number of days defined by `DAYS_THRESHOLD`. It compares each repository's last updated date with this threshold to determine if it's stale.
- **Output:** The script prints a list of repositories that haven't been updated in the last `DAYS_THRESHOLD` days.

Requirements

Ensure you have the `requests` library installed:
`pip install requests`