

## راهنما: ایجاد گردش کار گیت هاب اکشن برای وب اسکرپینگ

فرض کنید می‌خواهیم از طریق وب اسکرپینگ آخرین تیترهای خبری وبسایت‌های عمده خبری را گردآوری کنیم. کدی نوشته‌اید که با استفاده از پایتون و کتابخانه بیوتیفول سوپ قرار است صفحه اصلی این وبسایت‌ها را اسکرپ و تیترهای خبری را استخراج کند.

این وبسایت‌های خبری به طور مداوم به روز می‌شوند و ما می‌خواهیم هر ساعت مجموعه‌ای از تیترهای خبری داشته باشیم. برای انجام این کار ما می‌توانیم هر ساعت این کد را اجرایی کنیم که به نظر روش خیلی خوبی نمی‌آید. روش بهتری نیز داریم. گیت هاب اکشنز (GitHub Actions) به ما این امکان را می‌دهد تا کار اسکرپینگ به صورت منظم و طی زمان‌های مشخصی انجام شود. با این کار ما می‌توانیم به جای اجرای کد به صورت دستی، اسکرپینگ را به حالت خودکار در بیاوریم.

این‌ها مواردی هستند که می‌توان در آن از گیت هاب اکشنز استفاده کرد:

- گردآوری اخبار: می‌توان کار گردآوری اخبار از منابع مختلف، ذخیره‌سازی یا انتشار آن‌ها را به صورت خودکار درآورد.
- به کمک گیت هاب اکشنز می‌توان وبسایت‌ها را طی زمان‌های مشخصی اسکرپ کرد و آخرین اخبار را در اختیار گرفت.
- نظارت بر شبکه‌های اجتماعی: با اسکرپ کردن شبکه‌های اجتماعی می‌توان آن‌ها را تحلیل محتوایی کرد، کلیدواژه‌های خاصی را در آن‌ها دنبال کرد، و ...
- نظارت بر روند قیمت‌ها: با اسکرپ کردن و بررسی داده‌های وبسایت‌های تجاری، بورس یا رمزارز، می‌توان قیمت‌ها را بررسی کرد.

### نحوه استفاده از گیت هاب اکشنز

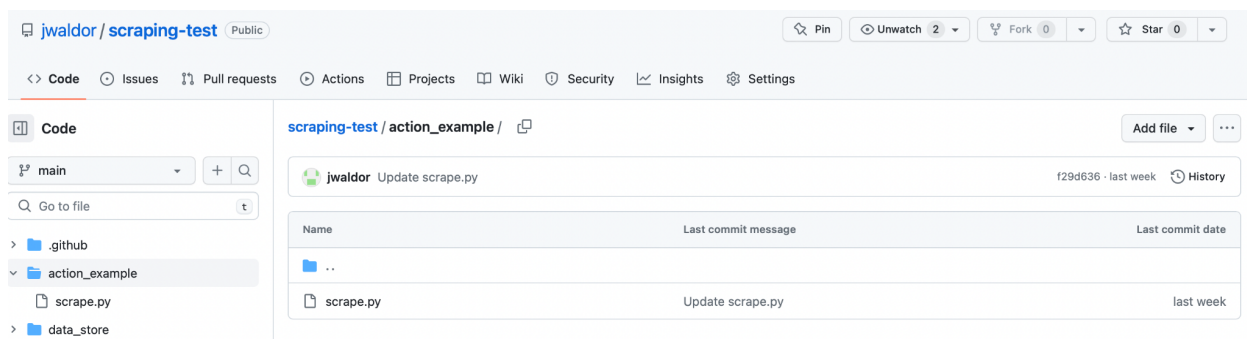
در این راهنما، نحوه ایجاد یک گردش کار (workflow) را در گیت هاب اکشنز توضیح می‌دهیم که می‌تواند یک وبسایت (در این مثال وبسایت [سی‌ان‌ان](#)) را هر ساعت به صورت خودکار اسکرپ، و داده‌ها را در یک فایل ذخیره کند:

**قدم اول: یک ریپازیتوری (مخزن) جدید در گیت هاب ایجاد کنید.**

**قدم دوم: کد اسکرپینگ را بنویسید:**

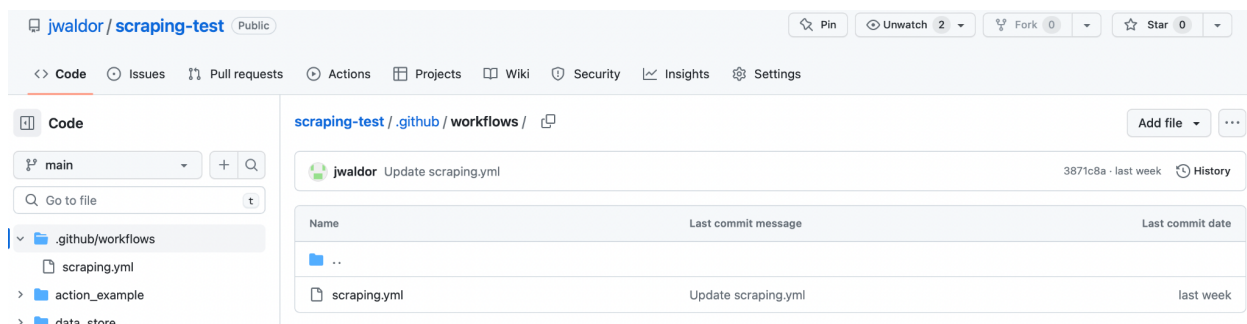
۱. یک فایل جدید در ریپوی تازه با نام `scrape.py` ایجاد می‌کنیم.

۲. [این کد](#) را کپی و در فایل `scrape.py` پیست کنید. این کد تمامی تیترهای خبری صفحه اصلی سی‌ان‌ان را گردآوری می‌کند (برای آشنایی با اصول ابتدایی وب اسکرپینگ به نوت‌بوک «مقدمات اسکرپینگ» مراجعه کنید).



**قدم سوم: یک فایل گردش کار ایجاد کنید:**

۱. در ریپوی خود یک فایل جدید به نام `github/workflows/scraping.yml` بسازید. این فایل گردش کار شما خواهد بود.
۲. [این کد](#) را کپی و در فایل `scraping.yml` پیست کنید.



حال درباره عناصر کلیدی فایل `scraping.yml` بالا توضیحاتی می‌دهیم؛ بخش‌هایی که برای زمانبندی به کار می‌آیند و حاوی قسمت‌هایی از پایتون هستند که برای اسکریپینگ استفاده می‌شوند.

`on: schedule: - cron: '0 * * * *'`

- بخش آن (`on`) برای تعیین مختصات گردش کار است. در مثال ما، این بخش، برای اجرایی شدن گردش کار طبق یک برنامه زمانی مشخص تنظیم شده است.
- بخش `cron` در قسمت `schedule` از سینتکس (`syntax`) کرون استفاده می‌کند که روشی است برای تعیین جدول زمانی. در اینجا

`'0 * * * *'`

به معنای اجرایی شدن گردش کار هر ساعت است. این فرمت خاص، پنج بخش دارد و هر بخش آن جنبه ویژه‌ای از جدول زمانی را شامل می‌شود.

این‌ها نمونه‌های دیگری از سینتکس کرون هستند. برای اطلاعات بیشتر می‌توانید به [این‌جا](#) مراجعه کنید.

● `* * * * *`

مسئولیت مد نظر را در هر دقیقه اجرایی می‌کند.

● `0 * * * *`

مسئولیت مد نظر را هر ساعت اجرایی می‌کند.

● `0 12 * * *`

مسئولیت مد نظر را روزی یک بار هر ساعت ۱۲ ظهر اجرایی می‌کند.

● `0 0 * * 1`

مسئولیت مد نظر را نیمه‌شب هر دوشنبه اجرایی می‌کند.

● `0 0 1 1 *`

مسئولیت مد نظر را اول ژانویه هر سال و رأس نیمه‌شب اجرایی می‌کند.

```
- name: Install dependencies run: | pip install beautifulsoup4 pip install requests
```

- نمونه بالا، با کلیدواژه `run` فرمان شیل را برای نصب بخش‌هایی از پایتون که برای `scrape.py` لازم است، اجرایی می‌کند.
- این روش با استفاده از `pip install` پکیج‌های `beautifulsoup4` و `requests` را نصف می‌کند.

```
- name: Run scraping code and save to file run: python action_example/scrape.py --output action_example/data.txt
```

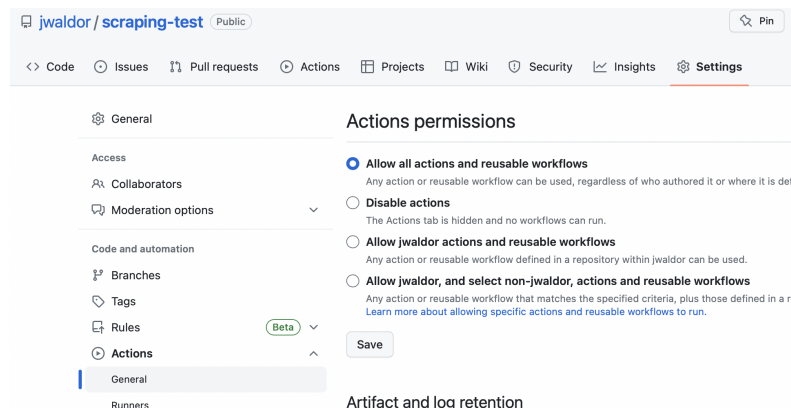
- گام نهایی، کد اسکریپت را با اجرای اسکریپت `scrape.py` اجرایی می‌کند.
- آرگومان `output--` مسیر (path) خروجی فایل را تعیین می‌کند.

تمامی قدم‌های بخش‌های مختلف مسئول انجام کارهای مشخصی هستند نظیر ایجاد فضا، نصب بخش‌های مورد نیاز و اجرای کد اسکریپت. با تعریف این قدم‌ها، گردش کار گیت‌هاب می‌تواند تمامی روند اسکریپت کردن وبسایت و ذخیره آن را به صورت خودکار انجام دهد.

**قدم چهارم: تنظیمات ریپوی خود را به نحوی تغییر دهید که بتواند فایل‌ها را به ریپو ارسال (push) کند:**

۱. به قسمت تنظیمات ریپوی خود بروید.

۲. در قسمت چپ صفحه روی **Actions** بروید و روی **General** کلیک کنید.



۳. به قسمت **Workflow Permissions** بروید، **Read and Write Permissions** را انتخاب و روی **Save** کلیک کنید.

## Workflow permissions

Choose the default permissions granted to the GITHUB\_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. [Learn more.](#)

☒ **Read and write permissions**

Workflows have read and write permissions in the repository for all scopes.

☐ **Read repository contents and packages permissions**

Workflows have read permissions in the repository for the contents and packages scopes only.

Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.

☐ **Allow GitHub Actions to create and approve pull requests**

Save

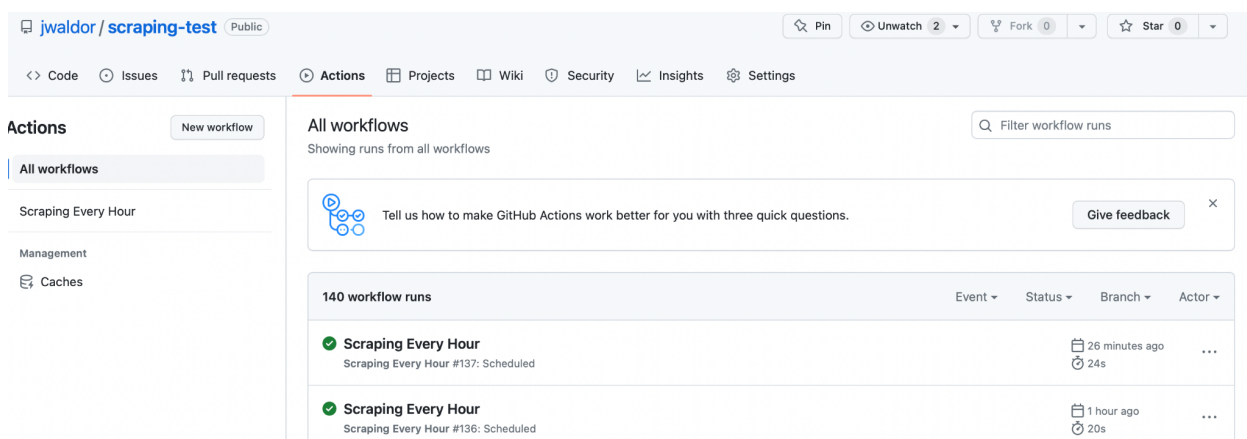
قدم پنجم: کامیت (Commit) و ارسال (Push) در گردش کار:

۱. تغییرات را در فایل `scraping.yml` ذخیره کنید.

۲. تغییرات را در ریپوی خود کامیت (Commit) کنید.

۳. کامیت را ارسال (Push) کنید تا فضای گردش کارتان آن را اجرا کند.

حال می‌توانید با کلیک روی بخش Actions از اجرای درست فضای گردش کار خود مطمئن شوید:

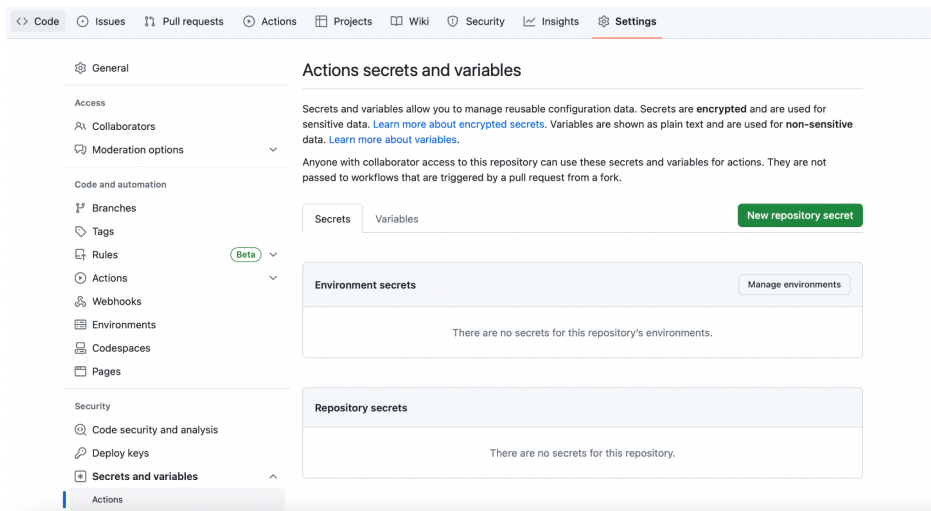


The screenshot shows the GitHub Actions interface for the repository `jwaldor / scraping-test`. The 'All workflows' section is active, displaying a table of workflow runs. The table has columns for Event, Status, Branch, and Actor. Two runs are listed, both for the 'Scraping Every Hour' workflow, with statuses 'Scheduled' and 'Scheduled'.

Event	Status	Branch	Actor
Scraping Every Hour	Scheduled		
Scraping Every Hour	Scheduled		

## نکته‌ای درباره Secrets

گاه می‌خواهید از کلید ای‌پی‌ای در کد اسکرپینگ خود استفاده کنید. کلید ای‌پی‌ای شما یک راز یا `secret` به شمار می‌رود چرا که نمی‌خواهید دیگران آن را ببینند. برای این‌کار می‌توانید از `Secrets` استفاده کنید. برای ایجاد `secret` می‌توانید به بخش `Actions` و سپس `Secrets and Variables` بروید و روی گزینه `"New repository secret"` کلیک کنید.



سپس می‌توانید به فایل `scraping.yml` خود امکان دسترسی به `secret` را بدهید:

```
- name: Set up environment variables
  env:
    API_KEY: ${ secrets.API_KEY }
```

در اینجا ما به کد پایتون امکان دسترسی به `API_KEY` را می‌دهیم. برای بارگذاری `secret` در اسکریپت پایتون، از کد زیر استفاده می‌کنیم:

```
import os
api_key = os.getenv('API_KEY')
```

حال می‌دانید که چطور می‌توان از گیت‌هاب اکشنز برای اسکریپ کردن استفاده کرد. می‌توانید دفعات اسکریپینگ را در `scraping.yml` بسته به نیازتان تغییر دهید و همچنین با ایجاد تغییرات در `scrape.py` تغییراتی در نحوه اسکریپینگ به‌وجود بیاورید.