

Fastcampus Data Science Extension SCHOOL

API Scraping & scheduling

Requirements

- requests
- bs4
- pymongo
- credit card(visa or mastercard)

API

- Application Programmable Interface

How to get data from API

[inspect] -> [Network]

XHR

- XML HTTP Request

test postman before requests.get()

postman으로 테스트 하는 이유

- API에 독립으로 접근이 가능한지 알아보기 위해
- API에 독립으로 접근했을 때 필요로 하는 헤더값들을 알아보기 위해

Let's get watcha news list

naver realtime keywords with requests, bs4, pymongo

Again!

Scheduling Web Scraper

1. cron

```
$ crontab file
```

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7)
# | | | | | OR sun,mon,tue,wed,thu,fri,sat
# * * * * * command
```

examples

```
0 * * * * /path/to/scrapper.py
```

```
0,20,40 * * * * /path/to/scrapper.py
```

```
*/1 * * * * /path/to/scrapper.py
```

```
*/10 * * * * /path/to/scrapper.py
```

```
* 9-18 * * * /path/to/scrapper.py
```

```
30 * * * 1-5 /path/to/scrapper.py
```

```
*/10 * * * 1-5 /path/to/scrapper.py
```

Let's scheduling crawler with cloud

Serverless application services

- Google cloud functions
 - only javascript yet..
- AWS Lambda
 - difficulty, durations..
- Microsoft Azure functions
 - easy! with GUI Only!

We'll use Azure functions

Create Account

<https://azure.microsoft.com/ko-kr/>

new functions

Microsoft Azure

홈 > 새로 만들기 > 기능 앱 > 기능 앱 만들기

기능 앱 만들기

* 앱 이름
[앱 이름을 입력하세요.]
.azurewebsites.net

* 구독
무료 체험

* 리소스 그룹 ⓘ
☒ 새로 만들기 ☐ 기존 그룹 사용

* OS
Windows Linux(미리 보기) Docker

* 호스팅 계획 ⓘ
사용 계획

* 위치
미국 중부

* 저장소 ⓘ
☒ 새로 만들기 ☐ 기존 항목 사용
96eb

Application Insights ⓘ 설정 해제

* Application Insights 위치 ⓘ

add storage(for scheduler)

Microsoft Azure

홈 > 새로 만들기 > 기능 앱 > 기능 앱 만들기

기능 앱 만들기

* 앱 이름
nvscraper ✓
.azurewebsites.net

* 구독
무료 체험

* 리소스 그룹 ⓘ
☒ 새로 만들기 ☐ 기존 그룹 사용
nvscraper ✓

* OS
Windows Linux(미리 보기) Docker

* 호스팅 계획 ⓘ
App Service 계획




* App Service 계획/위치
nvscraper(Korea Central) >


* 저장소 ⓘ
☒ 새로 만들기 ☐ 기존 항목 사용
nvscraper96eb ✓


Application Insights ⓘ 설정 해제

* Application Insights 위치 ⓘ
West US 2


deploy success



wychoi0408@hotmail...
기본 디렉터리


알림

해제: [정보 제공](#) [완료됨](#) [모두](#)

 배포 성공오후 12:52

리소스 그룹 'nvscraper'에 대한
'Microsoft.FunctionAppb235f3dd-b6b5' 배포에 성공했습니다.

[리소스로 이동](#) [★ 대시보드에 고정](#)

 ₩224,930 크레딧 남음오후 12:24

'무료 체험' 구독에 ₩224,930의 크레딧이 남아 있음

고급도구(kudu)

개요

플랫폼 기능

기능 검색

일반 설정

함수 앱 설정

응용 프로그램 설정

속성

백업

모든 설정

코드 배포

배포 옵션

배포 자격 증명

개발 도구

논리 앱

콘솔

고급 도구(Kudu)

App Service 편집기

리소스 탐색기

확장

네트워킹

네트워킹

SSL

사용자 지정 도메인

인증/권한 부여

관리 서비스 ID(미리 보기)

푸시 알림

모니터링

진단 로그

로그 스트리밍

프로세스 탐색기

메트릭

API

API 정의

CORS

APP SERVICE 계획

App Service 계획

규모 확대

할당량

리소스 관리

문제 진단 및 해결

활동 로그

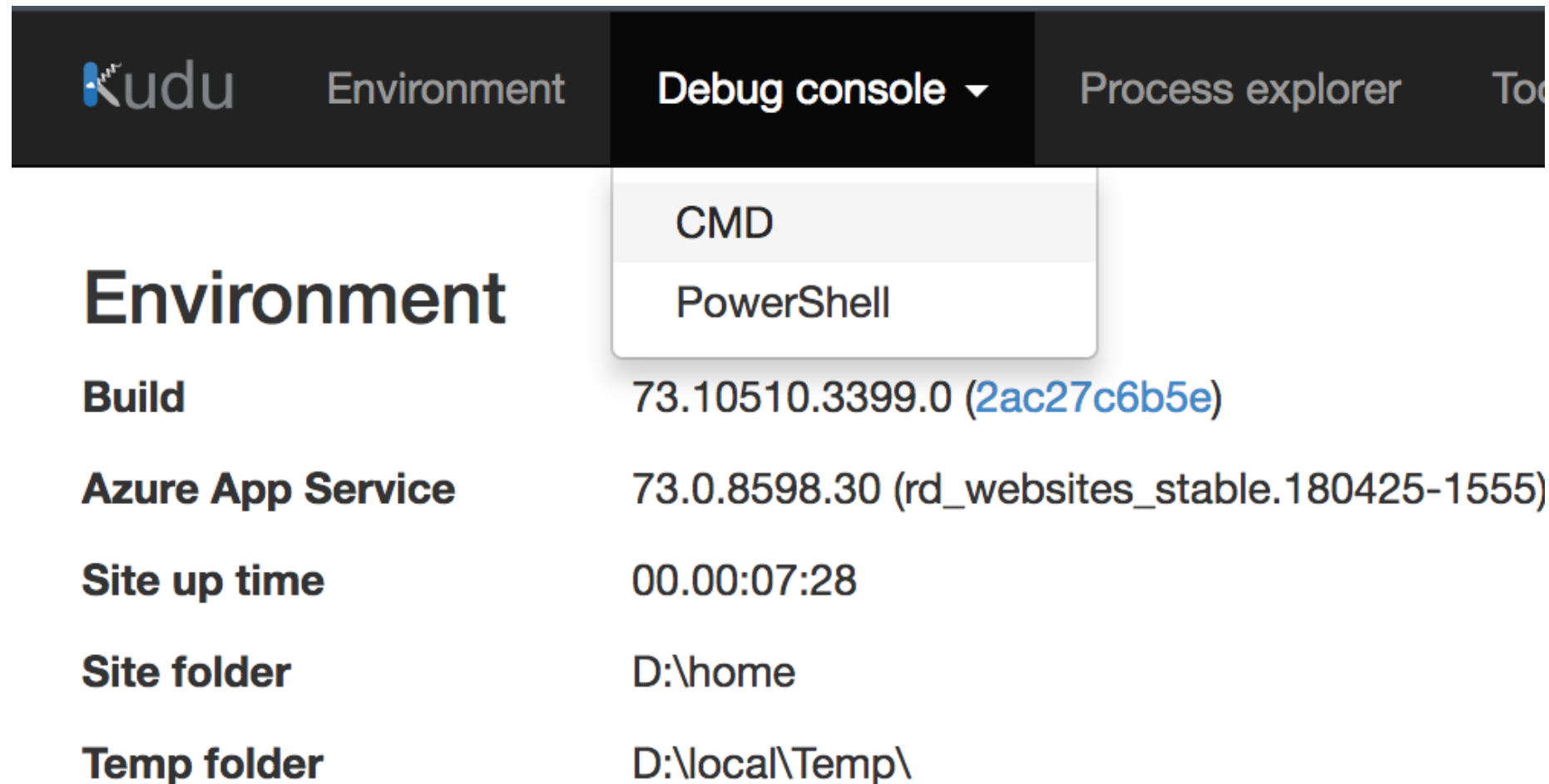
액세스 제어(IAM)

태그

잠금

자동화 스크립트

debug console -> cmd to install requirements



The screenshot shows the Kudu web interface. At the top, there is a navigation bar with the Kudu logo and several tabs: 'Environment', 'Debug console', 'Process explorer', and 'Tools'. The 'Debug console' tab is selected, and a dropdown menu is open, showing 'CMD' (highlighted) and 'PowerShell'. Below the navigation bar, the 'Environment' section is visible, displaying various system and application details.

Environment	
Build	73.10510.3399.0 (2ac27c6b5e)
Azure App Service	73.0.8598.30 (rd_websites_stable.180425-1555)
Site up time	00.00:07:28
Site folder	D:\home
Temp folder	D:\local\Temp\

REST API (works best when using a JSON viewer extension)

install python3

```
nuget.exe install -Source https://www.siteextensions.net/api/v2/  
-OutputDirectory D:\home\site\tools python364x64
```




mv python3 to azure env


```
mv  
/d/home/site/tools/python364x64.3.6.4.2/content/python364x64/*  
/d/home/site/tools/
```

install requests, bs4, pymongo via pip

```
d:/home/site/tools/python -m pip install requests beautifulsoup4  
pymongo
```


start with custom new function






미리 만들어진 함수로 빠르게 시작


1. 시나리오 선택



Webhook + API



타이머



데이터 처리


2. 언어 선택

☒ CSharp ☐ JavaScript ☐ FSharp ☐ Java

PowerShell, Python 및 Batch의 경우 [사용자 고유의 사용자 지정 함수 만들기](#).

이 함수 만들기

http trigger with python

 HTTP trigger

새 함수

언어:

Python ▼

이름:

nvscraper

HTTP trigger

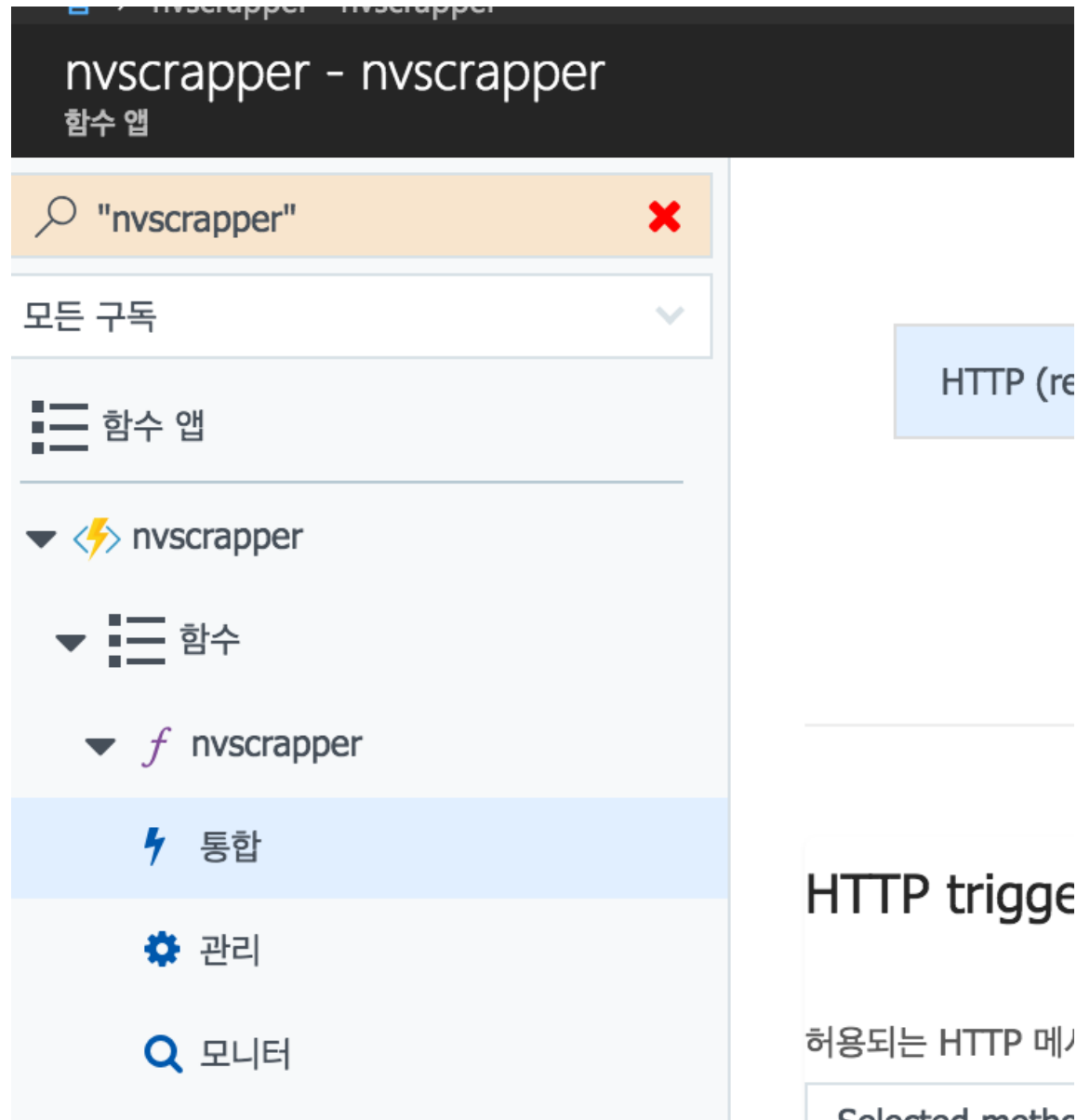
권한 수준 ⓘ

Function ▼

만들기

취소

in integration



delete existed trigger and add new trigger to schedule

+ 새 트리거

+



타이머

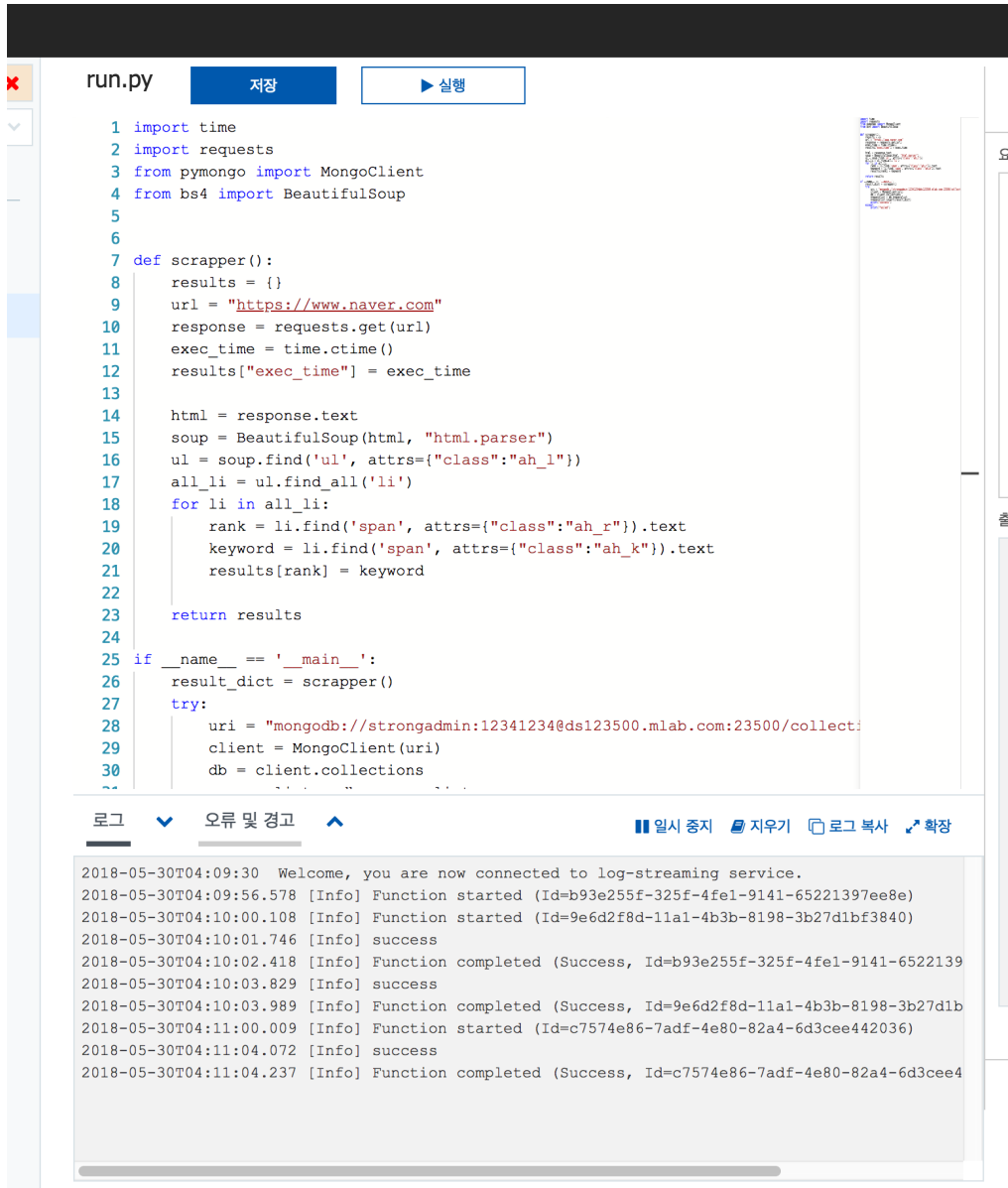


Azure Event Hubs



Azure

paste python script



The screenshot shows a web-based Python script editor. At the top, there's a dark header bar. Below it, the filename 'run.py' is displayed next to '저장' (Save) and '▶ 실행' (Run) buttons. The main area contains a Python script that scrapes data from naver.com and stores it in a MongoDB database. The script includes imports for time, requests, pymongo, and bs4, a scraper function, and a main execution block. To the right of the code editor, there's a small preview window showing the rendered HTML of the script. Below the code editor, there's a '로그' (Log) section with a dropdown menu for '오류 및 경고' (Errors and Warnings). The log area shows a series of messages indicating successful function execution and database connection.

```
1 import time
2 import requests
3 from pymongo import MongoClient
4 from bs4 import BeautifulSoup
5
6
7 def scrapper():
8     results = {}
9     url = "https://www.naver.com"
10    response = requests.get(url)
11    exec_time = time.ctime()
12    results["exec_time"] = exec_time
13
14    html = response.text
15    soup = BeautifulSoup(html, "html.parser")
16    ul = soup.find('ul', attrs={"class": "ah_l"})
17    all_li = ul.find_all('li')
18    for li in all_li:
19        rank = li.find('span', attrs={"class": "ah_r"}).text
20        keyword = li.find('span', attrs={"class": "ah_k"}).text
21        results[rank] = keyword
22
23    return results
24
25 if __name__ == '__main__':
26     result_dict = scrapper()
27     try:
28         uri = "mongodb://strongadmin:12341234@ds123500.mlab.com:23500/collecti
29         client = MongoClient(uri)
30         db = client.collections
31         ..
```

로그 ▼ 오류 및 경고 ▲ || 일시 중지 지우기 로그 복사 확장

```
2018-05-30T04:09:30 Welcome, you are now connected to log-streaming service.
2018-05-30T04:09:56.578 [Info] Function started (Id=b93e255f-325f-4fe1-9141-65221397ee8e)
2018-05-30T04:10:00.108 [Info] Function started (Id=9e6d2f8d-11a1-4b3b-8198-3b27d1bf3840)
2018-05-30T04:10:01.746 [Info] success
2018-05-30T04:10:02.418 [Info] Function completed (Success, Id=b93e255f-325f-4fe1-9141-6522139
2018-05-30T04:10:03.829 [Info] success
2018-05-30T04:10:03.989 [Info] Function completed (Success, Id=9e6d2f8d-11a1-4b3b-8198-3b27d1b
2018-05-30T04:11:00.009 [Info] Function started (Id=c7574e86-7adf-4e80-82a4-6d3cee442036)
2018-05-30T04:11:04.072 [Info] success
2018-05-30T04:11:04.237 [Info] Function completed (Success, Id=c7574e86-7adf-4e80-82a4-6d3cee4
```

