



Nobody cares how it works as long as it works

About Newest Archive Search

Django와 Express의 차이점

올해 8월은 꽤 재미있었다. 갓 스타트업을 시작하신 두 형님이 일하시는 곳이 옥수동에 있는데, 한 달 동안 거의 매일 그 곳에 놀러 갔다. 형들이 일하는 동안 나는 그 옆에서 Node를 공부했다. 그동안 외롭게 개발 공부를 했던 나에게는 가장 가까이에 물어볼 수 있는 사람이 둘이나 있었고, 무엇보다 두 형과 코드 (그 코드가 아니다...)가 잘 맞았기 때문에 더할 나위 없이 좋았다.

난 꽤 오래 Django를 썼다. 하지만 잘 쓰지는 못했다. Django에게는 Python만큼 큰 흥미를 느끼지 못하고 있었기에 열심히 하지 않았다. 초보 개발자의 눈으로 보기엔 Django는 이미 너무 거대해져서 진입장벽이 높아 보이는 프레임워크였다.

Django는 Python의 “batteries included” 철학을 그대로 차용했다. 웹 개발시 필요한 기능들은 대부분 이미 내장되어 있으므로, 사용하고자 하는 기능이 있다면 문서를 보고 Django에 내장된 라이브러리를 import해서 사용하면 된다. 돌이켜보면 이러한 Django의 편의성이 내 개발능력 발전에 벽이 된 것 같다.

많은 기능을 담았기에 문서의 양 또한 방대했고, 필요한 기능을 찾아서 써도 내 것이 되지 않았다. 당장 필요하지 않은 기능들임에도 불구하고, 모르고 있는 것이 너무 많다는 것이 부담되었다. 어떻게 구현해야 좋을지 몰라서 구글링하면, 내장 라이브러리를 import하는 것만으로도 끝났다. 열심히 직접 구현했더니 이미 내장된 라이브러리가 있는 경우도 꽤 있었다. 제공된 기능을 커스터마이징하려면 여기에 얹혀있는 다른 Django의 기능들도 함께 학습해야만 했다. 스크롤의 압박이 있는 영어문서를 읽는 것이 고역이었다. 분명 문서를 보면 해답을 찾을 수 있음에도 문서는 팽개쳐둔 채 바로 보고 이해할 수 ~~써먹을 수~~ 있는 코드를 찾아 구글링을 계속했다.

아무튼 그렇게 Django에 대한 흥미가 떨어져 가고 있을 때, Javascript를 해보기로 했다. 선택에 대한 이유가 있는 건 아니었다. 여기저기서 ‘올해의 대세는 Javascript다’라고 어렴풋이 들었다. 두 형님의 스타트업 기술스택도 Javascript 기반이었다.

Javascript를 익히는 건 어렵지 않았다. 예전에 jQuery와 묶어서 공부해본 적이 있었고, es6부터 대부분의 문법이 다른 언어에서도 익숙한 것들이었기 때문에 한 번 훑어보기만 해도 바로 사용할 수 있을 정도였다.

3년 전 출판된 책을 보면서 Express를 공부하는데, 코드가 작동을 안 했다. 기존에 Express 안에 있던 기능들을 모듈화해서 **밖으로 빼냈기 때문에**, 필요하면 따로 모듈을 설치해주어야 한단다. **Django와는 전혀 다른 방향이다.** Django에서는 내장된 코드만 정답 같아 보였지만, 여기서 정답 대신 다양한 선택지가 있었다. 필요한 기능이 있으면 다른 사람들은 어떤 걸 많이 쓰나 찾아보고 설치했다.

이때부터 문서를 읽기 시작했다. 마음먹고 읽으면 잘 읽을 수 있음에도 영어 울렁증 때문에 읽으려고 하지 않았던 게 극복되었다. 문서를 읽으면서 처음보는 이름들을 검색해서 찾아 보니 babel이나 webpack 등 Javascript 개발시에 자주 쓰이는 툴들도 쉽게 적용할 수 있었고, SPA나 JWT처럼 웹 개발과 관련된 개념들도 알게 되었다.

지금까지가 올해 8월 한 달 동안 있었던 이야기다. Django를 붙잡고 몇 달동안 공부하던 것을 Node+Express로는 겨우 한 달 만에 해냈다.

어떻게 보면 결과론적인 이야기다. Django는 문서화가 잘 되어있기로 손꼽히는 프레임워크 중 하나라, 만약 지금 Django를 다시 쓴다면 이전보다는 더 잘 쓸 수 있을 것 같다. Django 대신 flask를 주로 썼거나, Python보다 Javascript를 먼저 썼더라면, 이 글은 위에 쓴 내용과 정반대가 되었을지도 모른다. 사실 난 지속적으로 성장하고 있었을 뿐, 다른 언어나 다른 프레임워크를 사용하는 게 내 성장에 큰 영향을 미치지 않았을 수도 있다. 하지만 난 지금 Javascript를 사용하게 된 것에 대해 만족하고 있기에 이렇게 글을 남긴다.

Oct 8, 2016

vimium

나는 웹서핑을 하면서 마우스를 거의 사용하지 않는다.

대부분 키보드 위에 두 손을 가지런히 올려놓고, 손가락만 움직이면서 웹서핑을 한다. 마우스를 사용하는 것보다 동선이 작아서 편하다. Chrome의 확장 프로그램인 **Vimium** 덕분이다. Vimium은 크롬에서 vim의 키보드 단축키를 사용하게 해준다.

내가 자주 쓰는 단축키는 아래와 같다. 대문자는 **Shift** 키와 조합해서 누른다. 모든 단축키는 **?** 을 입력하면 확인할 수 있다.

스크롤

Key	설명
d	스크롤 빠르게 내리기
u	스크롤 빠르게 올리기
j	스크롤 천천히 내리기
k	스크롤 천천히 올리기
gg	스크롤 맨 위로 이동
G	스크롤 맨 아래로 이동

링크

Key	설명
f	링크 열기
F	새 탭에서 링크 열기

- **f** 를 누르면 각 링크마다 노란색으로 말풍선이 뜬다. 이동을 원하는 링크의 말풍선 키워드를 키보드로 입력하면 링크로 이동한다.
- **F** 를 누르고 링크 키워드를 입력해도 되고, **f** 를 누르고 링크 키워드를 **Shift** 와 입력해도 된다.

탭 이동

Key	설명
J	왼쪽 탭으로 이동
K	오른쪽 탭으로 이동

히스토리

Key	설명
H	뒤로 가기
L	앞으로 가기
r	새로고침

탭 관리

Key	설명
x	현재 탭 닫기
X	이전에 닫은 탭 열기
t	새 탭 열기
T	탭 목록

주소창

Key	설명
o	주소창
O	새 탭 주소창

검색

Key	설명
/	검색
n	검색결과 다음 항목
N	검색결과 이전 항목

- **/** 을 누르면 우측 하단에 작은 입력창이 하나 뜬다. 여기에 검색하고 싶은 내용을 입력하고 **Enter** 를 누르자. 그러면 첫번째 결과값이 표시된다. 이 상태에서 **n** 이나 **N** 를 누르면 다음

nodemon

nodemon은 실행중인 node 코드에 변경사항이 있으면 자동으로 해당 어플리케이션을 재시작해주는 어플리케이션이다. 웹 개발을 하는 경우, 코드를 수정하고 서버를 켜다가 다시 켜곤 하는데, nodemon을 사용할 경우 코드를 변경하고 저장하면 알아서 서버가 재시작된다. 사용법은 간단하게 `node` 대신 `nodemon` 을 써주면 된다. 실행 중에 `rs` 를 타이핑하면 수동으로 재시작해준다.

```
$ nodemon app.js
```

`--watch` 옵션으로 변경사항을 감지할 파일이나 폴더를 추가해줄 수도 있고, `--ignore` 로 제외할 목록을 설정할 수도 있다.

```
$ nodemon app.js --watch src
$ nodemon app.js --ignore build
```

node 외의 다른 프로그램을 실행할 때는 `--exec` 옵션을 이용한다. node로 개발하는 경우 es6, es7 코드를 **babel-node**로 실행하는데 nodemon과 함께 쓰려면 `--exec` 을 쓰면 된다.

```
$ nodemon --exec babel-node app.js
$ nodemon --exec python script.py
```

자세한 사용법은 문서! 문서를 보자!

