

Final Project : WSGI를 이용한 서버를 통해 모바일 앱 만들기

20132926 전호현

요 약

Raspberry PI를 활용하여 WSGI를 이용해 서버를 만들고, 이를 Apache 웹 서버가 실행하도록 설정한 후, APP Inventor를 사용해 만든 모바일 클라이언트에서 동작하는 것을 확인한다.

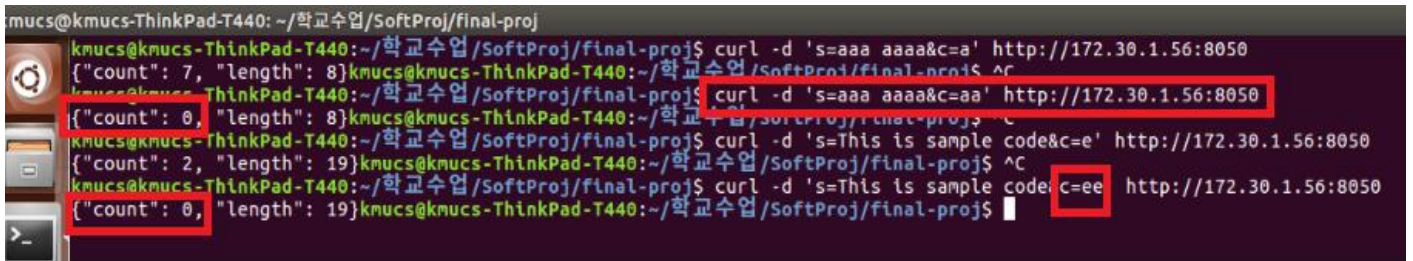
1. WSGI 를 이용한 서버 만들기

- 먼저, 테스트 스크립트를 만들어 실행해 본다

```
from wsgiref.simple_server import make_server
from cgi import parse_qs
import json
def application(environ, start_response):
    try:
        # HTTP POST request body를 얻어온다
        request_body_size = int(environ.get('CONTENT_LENGTH', 0))
    except ValueError:
        request_body_size = 0
    request_body = environ['wsgi.input'].read(request_body_size)
    # Request body에 들어있는 Query String을 분석한다
    d = parse_qs(request_body)
    # Query String 분석 결과로부터 입력 인자를 꺼낸다
    s = d.get('s', [''])[0]
    c = d.get('c', [''])[0]
    # 문자열의 길이를 구하는 연산
    length = len(s)
    # 찾는 character 'c'의 개수를 구하는 연산
    # 'c'가 1이 아닐경우의 예외처리 포함
    if len(c) == 1 :
        count = s.count(c)
    else :
        count = 0
    status = '200 OK'
    # json 을 이용한 출력 데이터 포매팅
    response_body = json.dumps({'length': length, 'count': count})
    response_headers = [
        # Tuple Lists
        ('Content-Type', 'application/json'),
        ('Content-Length', str(len(response_body)))
    ]
    # status 와 response_header로 Response 시작한다.
    start_response(status, response_headers)
    # response_body를 list로 리턴한다.
    return [response_body]

httpd = make_server('172.30.1.56', 8050, application)
httpd.serve_forever()
```

- Curl을 통해 서버 테스트(스크린샷)



```
kmucs@kmucs-ThinkPad-T440: ~/학교수업/SoftProj/final-proj
kmucs@kmucs-ThinkPad-T440:~/학교수업/SoftProj/final-proj$ curl -d 's=aaa aaaa&c=a' http://172.30.1.56:8050
{"count": 7, "length": 8}
kmucs@kmucs-ThinkPad-T440:~/학교수업/SoftProj/final-proj$ curl -d 's=aaa aaaa&c=aa' http://172.30.1.56:8050
{"count": 0, "length": 8}
kmucs@kmucs-ThinkPad-T440:~/학교수업/SoftProj/final-proj$ curl -d 's=This is sample code&c=e' http://172.30.1.56:8050
{"count": 2, "length": 19}
kmucs@kmucs-ThinkPad-T440:~/학교수업/SoftProj/final-proj$ curl -d 's=This is sample code: c=ee' http://172.30.1.56:8050
{"count": 0, "length": 19}
```

첫 번째 예시에서, String이 'aaa aaaa'이고 찾는 문자가 'a'일 경우 'a'의 Count가 7, String 길이가 8로 올바르게 나오는 것을 확인할 수 있다. 세 번째 예시도 마찬가지이다.

두 번째와 네 번째 예시는 예외처리 한 부분의 예시를 보여주는데, 위 스크린샷에서 빨간색으로 표시한 부분이다. 찾고자 하는 Character 'c'의 길이가 두 자 이상일 경우의 Count는 0을 반환한다.

- WSGI 스크립트 배치

인터프리터를 직접 실행하지 않고, Apache 웹 서버가 WSGI 스크립트를 실행하도록 설정하기 위해 Apache 웹 서버 설정파일을 바꾸고 WSGI 스크립트의 디렉토리를 정해서 배치한다.

먼저, /etc/apache2/conf-enabled 위치로 이동하여 Apache 웹 서버 설정파일을 추가한다. 설정파일의 이름은 strlen.conf라고 하고 아래와 같은 내용을 추가한다.

```
WSGIScriptAlias /strlen /var/www/wsgi/strlen.py
```

다음으로 원래 만들었던 스크립트를 /var/www/wsgi/ 경로로 이름은 strlen.py로 바꾸어서 복사해 저장한다. 그리고 strlen.py 파일에서 아래 세 줄을 삭제한다.

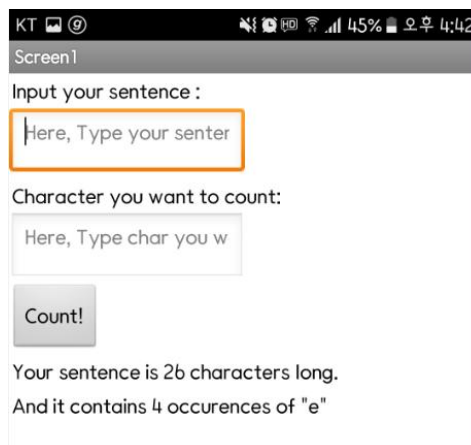
```
from wsgiref.simple_server import make_server
httpd = make_server('172.30.1.56', 8050, application)
httpd.serve_forever()
```

서버를 재시작 한 후에 위에서 curl을 이용한 것과 같이 테스트를 수행한다. 다만, URL은 이제 <http://172.30.1.56:8050>이 아니라, <http://172.30.1.56/strlen> 으로 표현해 주어야 한다. (Apache를 통해 실행하기 때문)

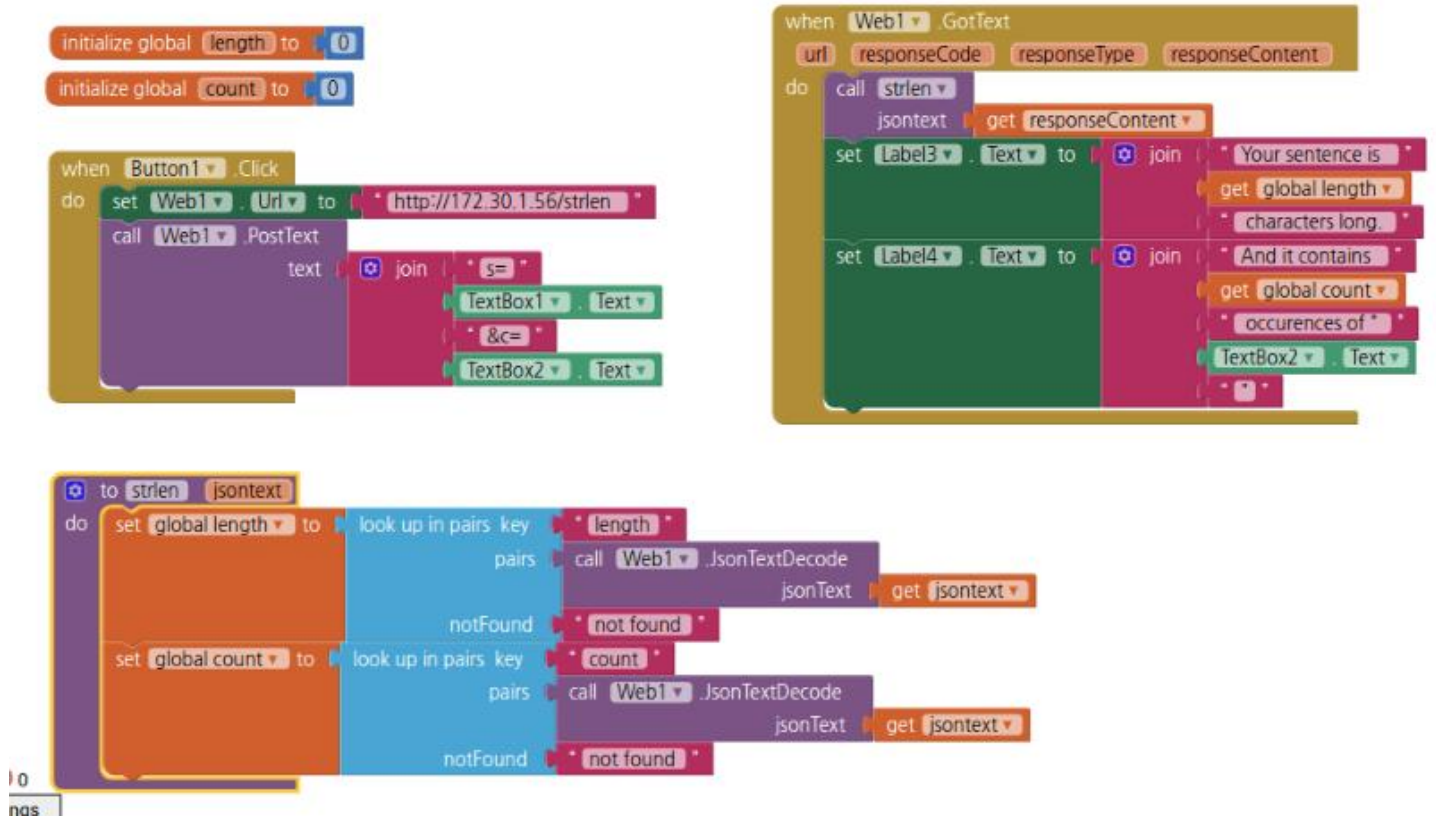
2. 모바일 클라이언트 - UI 디자인

- App Inventor를 활용하여 UI 컴포넌트 배치(스크린샷)

아래 그림과 같이 UI 컴포넌트들을 배치한다.



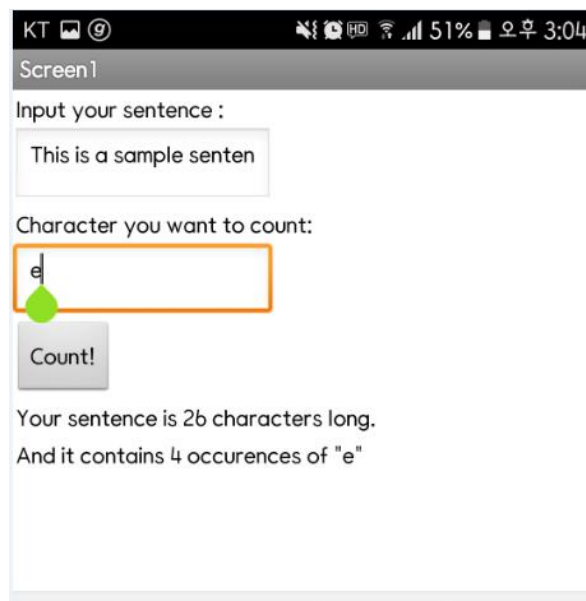
- 클라이언트 프로그램 블록 코딩(스크린샷)



3. 서버와 클라이언트 동작 설명

서버는 클라이언트로부터 S(String) 값과 C(Character) 값을 받아온다. 클라이언트에서는 텍스트 박스 내부에 S와 C를 각각 사용자로부터 입력받아 Query String 형태로 웹서버에 보내주는 역할을 담당한다. 서버에서는 이렇게 받아온 Query String을 분석하여 S와 C값을 뽑아낸 후에 S에 대한 길이 length와 S에 있는 C의 개수를 찾은 count 값을 구해낸 후에 json을 활용해 response_body를 포매팅 한 후에 이것을 반환하는 역할을 수행한다. 클라이언트에서는 다시 서버로부터 받은 jsonText 값을 분석해 length와 count라는 변수에 저장한 후에, 정해진 UI형태에 따라 사용자에게 보여질 수 있도록 해당 변수들을 출력한다.

4. 최종 결과물 스크린샷



5. 프로젝트를 마치고

처음 수업시간에 프로젝트 과제를 교수님으로부터 들었을 때는 어떻게 해야할 지 전혀 감이 잡히지 않았다. 수업시간에는 내가 지금 하고 있는 내용에 대한 정확한 이해가 없이 그저 PDF파일을 보며 하라는 대로 따라만 했기 때문이었다. 하지만 과제를 처음부터 차근차근 하나씩 뜯어보니 결국은 수업시간에 했던 내용들과 비슷한 과제물이었다. 다만, 수업시간에는 무엇인지에 대한 정확한 이해가 없이 그저 수동적으로 따라하기만 했다면, 이번 과제를 하면서는 이러한 과정이 왜 필요하며 어떤 기능을 수행하게 될 것인지에 대해 따져보며 능동적으로 차근차근 수행해 나갔다. 수업시간에는 따라가기에 급급해 보이지 않았던 의미와 내용들이 눈에 보였던 것 같아 좋았다. 또한, Git을 활용하는 것도 수업시간에는 많이 와닿지 않았는데, 이번 과제를 하며 실제로 사용해 보니 어떤 식으로 작동하는지 조금은 감을 잡을 수 있었던 듯 하다. 프로젝트를 완료하긴 했지만, 스크립트의 코드 하나하나가 무엇을 의미하며 어떤 식으로 쓰이는 지는 정확히 알지 못한다. 하지만 이렇게 서버를 만드는 작업에 대한 호기심이 생겼고, 방학 동안에 웹 클라이언트 컴퓨팅이라는 과목을 미리 배워보고 싶다는 욕심이 생겼다. 그래서 방학 동안에 어떤 식으로 공부를 할지 계획을 짜는 중이다.

사실 이번 학기에 처음으로 전과를 해서 소프트웨어라는 과목에 대해 많이 어렵다고만 느껴졌었는데, 이 과목을 수강하면서 소프트웨어에 대한 호기심과 재미를 느낄 수 있어서 좋았다. 특히, App inventor와 라즈베리 파이를 이용했던 실습이 한 학기동안 가장 기억에 남고 재미있었던 것 같다. 나는 3학년이긴 하지만, 1학년 신입생들에게 소프트웨어라는 전공에 대한 재미와 호기심을 줄 수 있는 좋은 과목이라고 생각한다.

한 학기동안 어려운 내용들을 쉽게 풀어서 자세히 설명해주신 이경용 교수님께 감사드립니다.