

파이썬 과제 계획 및 결과

학과 : 컴퓨터공학과
학년 : 3학년
학번 : 2012722041
이름 : 김승훈
연락처 : 010-7192-3502

1. 제목

광운대 학생들을 위한 U-Campus 알림

2. 필요성 및 동작 개요

광운대 학생이라면, U-campus에 등록되는 모든 자료들을 간편하게 알고 싶어할 것이다. 예를 들어, 공지사항이 올라왔을 경우, 바로 알 수 있는 방법이 없고 직접 노트북이나 스마트폰을 이용해 직접 U-campus에 접속해서 일일이 확인하는 방법 밖에 없다. 그로 인해, 바쁜 일이 있어 제 때 공지사항을 확인하지 못해, 문제가 생기는 경우가 발생할 수 있고, 또한 인터넷 강의가 등록되었는데 확인을 하지 못해 강의시간이 지나가 출석을 하지 못하는 경우가 존재한다. 따라서, 이러한 문제점을 해결하기 위해 U-campus 알림 서비스가 필요하다고 생각했다.

안드로이드 application을 download 받아, 학번과 비밀번호를 입력하여 u-campus에 접속하여 업데이트 되는 내용이 있을 경우, 학번과 비밀번호를 입력한 기기에서 업데이트 내용을 푸시 알림을 받도록 동작.

3. 구현 내용 및 설계

먼저, Android Studio를 이용하여, 학번과 비밀번호를 입력받고, U-campus 내에 업데이트 된 내용이 있을 경우, 푸시 알림을 받을 수 있는 application을 개발

Android application과 PYTHON을 Socket 통신을 이용하여, Android application에서 입력 받은 학번과 비밀번호, 로그인 한 기기로부터 푸시 알림을 넘겨주기 위해 사용하는 FCM Server로 보낼 Token 값들을 PYTHON code로 넘겨준다.

PYTHON code에서는 Android application에서 socket 통신 요청을 기다리다가, 연결 요청이 올 경우, Thread를 생성하여, 학번과 비밀번호, Token 값들을 넘겨받아 decode 한 후, 값들을 .split()을 이용하여 " "단위로 자른 후, 값들을 저장한다. Thread를 사용하는 이유는 사용자가 한명이 아닌 여러 명일 경우, 각각 입력받은 학번에 따라서 듣는 과목과 업데이트 되는 내용, 푸시 알림 받을 기기들이 다르기 때문에 이를 사용하였다.

PYTHON code에서는 urllib.request를 이용하여 U-campus 로그인 페이지에 저장한 학번과 비밀번호, 필요한 header 정보와 로그인에 필요한 값들을 보내 response 값을 받아 온다. response 값에서 Set-cookie에 저장된 값을 로그인 유지를 위한 Cookie 값들을 저장한다. 그 후 저장된 Cookie 값 중에, 'ccmedia' 라는 String이 존재하는지 re.search()를 이용하여 찾은 후, 해당 값이 있을 경우에는, 'login not success' 라는 String을 Android application으로 보내준 후 Socket을 닫고 Thread 또한 종료시켜준다. 만약 'ccmedia' 값이 없을 경우에는 'login success' String을 Android application으로 보내준 후 Socket을 닫는다.

로그인이 성공한 후, push-service를 위한 FCM Server에 있는 api-key 값을 직접 입력하여 저장한다. 그 후, 각 과목들의 강의실 및 강의자료, 공지사항, 과제 제출, 퀴즈 페이지의 내용들의 개수를 저장하기 위한 각각의 count list들을 선언한다. count2의 경우, 한번 페이지를 탐색했는지 안했는지 확인하기 위한 변수로써 선언한다.

다음으로, 페이지들을 계속 탐색하기 위한, While문을 설정한다. While문의 경우, 무한 반복하게끔 True로 설정한다. 그 후, U-campus의 첫 페이지의 url 값을 저장한 후, urllib.request를 이용하여 필요한 header 정보와 로그인 시 받은 Cookie 값을 이용하여, response 받아, U-campus에 존재하는 페이지에 접속하기 위한 또 다른 쿠키 값인 JSESSIONID 와 WMONID 값을 받아온다. 그 후 이 값들을 for 문과 .split()을 이용하여 해당 값들을 이전 쿠키 값의 list에 저장한다.

우선, 과목 정보와 과목에 해당하는 sequence number를 찾기 위한 url을 저장한다. 위

와 마찬가지로, urllib.request를 이용하여 페이지와 연결하기 위한 header 값과 cookie 값을 설정한 후, response 값을 받아 온다. response 값을 받아와 BeautifulSoup을 사용하여 parsing 한다. 그리고 parsing한 값을 이용하여, 사용자가 듣는 과목 정보들과 해당 과목의 sequence number를 각각의 list에 저장한다. 과목 정보와 해당 과목의 sequence number의 경우, soup.find_all()을 이용해, 찾은 후, re.search()를 통해, 해당 내용이 있는 부분만을 가져와 저장한다. 이때, lstrip()을 이용하여 필요한 부분만을 얻은 후 저장한다.

우선 강의실의 내용 중 업데이트 된 내용이 있는지 확인하기 위해 for문을 설정한다. 해당 for문의 경우, 사용자가 듣는 과목의 개수만큼 돌도록 과목의 subsequence number가 저장된 list의 길이만큼 돌도록 설정한다. 이 때 list의 길이를 구하기 위해, len()을 사용한다. 해당 for문 내에서 subsequence list에 저장된 값들을 과목 number와 년도, class 번호를 .split()을 이용해 나눈 후, .lstrip(), .rstrip()을 이용해 첫 번째와 맨 마지막에 존재하는 ""의 값을 제거하여 저장한다. 이 값을 이용해, 강의실 접속에 필요한 url을 저장한 후 해당 페이지를 받아오기 위한 data 들과, header 값 cookie 값을 이용해 연결한다. 그 다음, BeautifulSoup을 이용하여, 강의실에 존재하는 제목 정보들을 저장하고, 제목 정보들의 개수를 count 하여 저장한다. 그리고 이전에 탐색했는지 안했는지 비교한 후, 탐색했을 경우 해당 count 값과 이전에 저장해놓은 count 값을 비교하여 해당 count 값이 클 경우, message_title 과 message_body(과목 이름, 업데이트 된 내용의 제목)를 설정하고 그 값을 pyfcm module에 존재하는 push_service.notify_single_device()를 이용해 data_message에 저장하여 FCM server로 보내준다. 이 경우, FCM server에서 값들을 받아 해당 안드로이드 기기로 알림 메시지를 보내준다. 그 후, list에 존재하는 이전 count 값을 .pop()를 이용하여 지우고 새로운 count 값을 .append()를 이용하여 저장한다. 만일 한 번도 탐색하지 않았을 경우에는 count 값을 list에 저장한다.

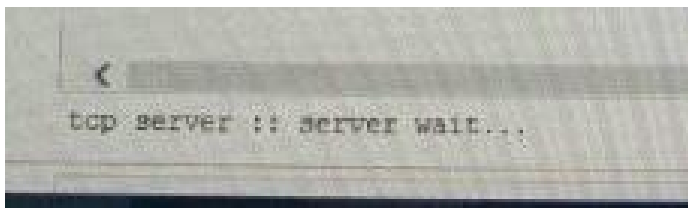
다음으로는 강의 자료실, 공지사항, 과제 제출, 퀴즈, 학생 자료실 페이지를 강의실 페이지와 마찬가지로 탐색한다. 다만 강의 자료실과 공지사항의 경우는 해당 페이지 접속 시, p_pageno의 값이 필요로 한다. p_pageno의 경우 해당 페이지가 여러 페이지로 존재해서, 해당 페이지의 모든 페이지를 탐색하기 위해 while문을 추가하여 구현하였다.

모든 페이지를 탐색한 후에는 thread.sleep()을 이용하여 20초간 해당 thread가 쉬도록 하였고, 한번 탐색이 끝났을 경우, count2 값을 1로 설정하였다.

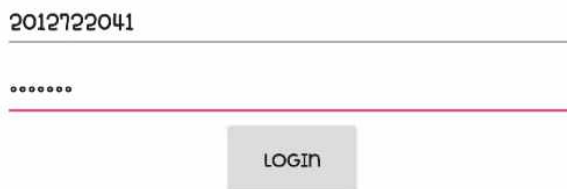
Android application의 경우, 우선 로그인할 수 있는 화면을 제공한다. 로그인 화면에서 학번과 비밀번호를 입력 시, login이 성공했는지 실패했는지에 대한 결과가 화면에 뜬 후, 로그인이 성공할 경우, 메시지가 왔을 경우, 확인할 수 있는 목록으로 넘어간다.

4. 실행 결과

- (1) PYTHON code를 실행 시, server wait이라는 메시지와 함께 Socket 통신을 위해 대기하는 모습을 확인할 수 있다.



- (2) Android application 내, 로그인 페이지에서 학번과 비밀번호를 입력하는 모습을 확인할 수 있다.



- (3) **맞지 않는 학번과 비밀번호를 입력 시, login not success라는 메시지가 뜨는 것을 확인할 수 있다.**



- (4) 알맞은 학번과 비밀번호를 입력 시, login success라는 메시지가 뜨는 것을 확인할 수 있다.



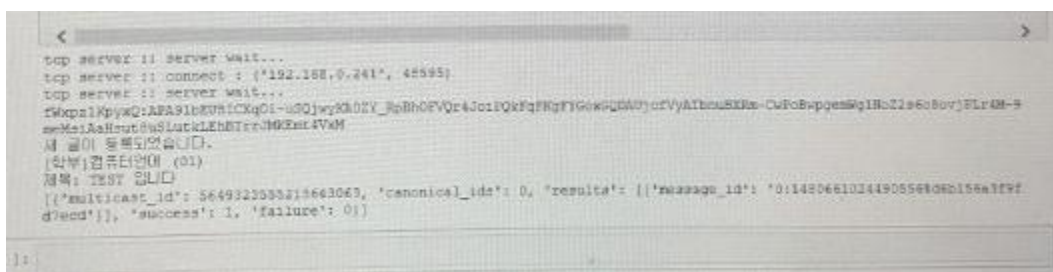
- (5) 로그인 성공 후, 알림 메시지가 올 때마다 message가 저장되어 보여주는 화면으로 넘어가는 모습을 확인할 수 있다.



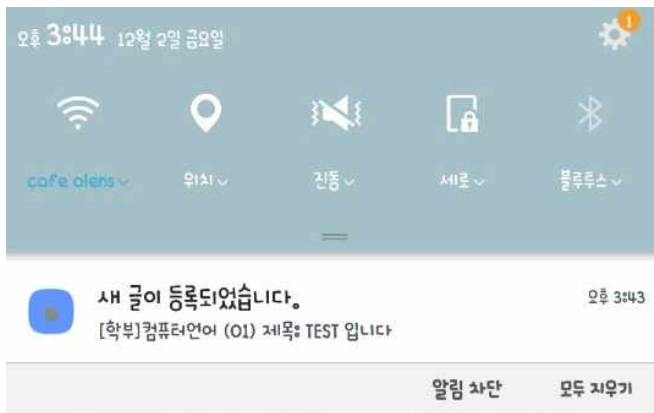
- (6) 아래와 같이, 학생 자료실을 이용하여, 테스트 하기 위해 글을 작성한 모습을 확인 할 수 있다.



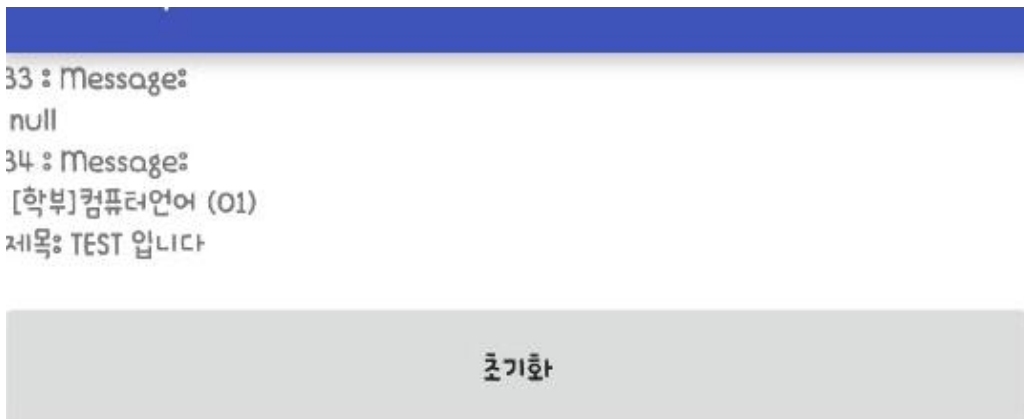
- (7) 서버에서 해당 내용을 catch 하여 FCM Server로 메시지를 보내는 모습을 확인할 수 있다.



- (8) FCM Server에서 안드로이드 기기로 메시지를 보내, 해당 메시지가 안드로이드 기기
에 알리미 뜨는 모습을 확인할 수 있다.



- (9) 알림을 클릭 시, 메시지가 저장되어 보여주는 페이지로 넘어가 저장된 메시지를 보여주는 모습을 확인할 수 있다.

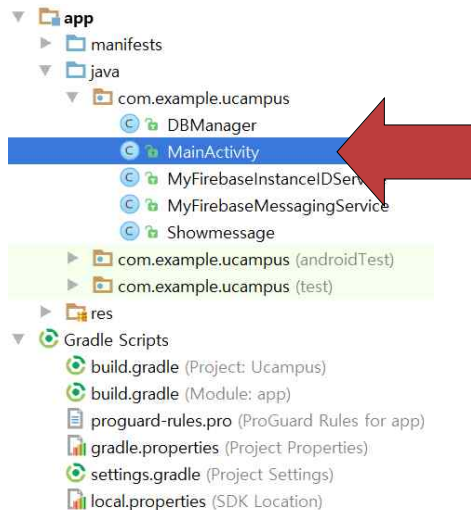


5. 결론

이번 프로젝트를 진행하면서, 처음 무엇을 구현을 할까에 대한 많은 고민을 하였다. 고민을 시작할 때, 학교를 다니면서 불편한 사항이 무엇일까 라는 고민을 하게 되었고, U-Campus 알림기가 있을 경우 편리하겠다는 생각을 하였다. 그래서 주저 없이 이에 대한 자료 조사를 시작하였다. 처음에는 어떤 식으로 이 알림을 알려줄 수 있을까 라는 큰 난제에 부딪혔다. 이메일로 알림기를 해보자 현재 브라우저를 통해 들어가서 확인하는 방식과 큰 차이가 없어 다른 생각을 해보았다. smart 기기에서 직접 보여주는 것은 어떨까라는 고민을 하였다. 처음에는 .py 파일을 안드로이드 기기에서 실행할 수 있도록 도와주는 SL4A라는 application이 존재하는 것을 알게 되었다. 그러나 이럴 경우, 다수의 이용자가 사용하지 못하고 오로지 자기 자신만 사용할 수 있으며, 또한 SL4A 어플을 받아서 이용자가 직접 설정을 해줘야되는 번거로움이 있어 다른 방식을 찾아보았다. 그 중에 구글에서 제공하는 FCM Server가 존재한다는 것을 알게 되었고 PYTHON에서도 사용이 가능한 여부를 찾아보던 중 마침 PYFCM 이라는 Module이 존재해 이것을 이용해서 진행하면 되겠구나 라는 생각을 가졌다. 따라서 PYTHON서버를 만들어 안드로이드 기기와 통신을 통해 PYTHON Server에서 안드로이드 기기에서 받은 값을 이용해 로그인하고, 그 값을 이용해 U-campus 페이지들을 Parsing 하여 Update 되는 값이 있을 때마다 해당 값을 PYFCM을 이용해 FCM Server로 보내는 방식으로 구현하면 되겠구나 라는 생각을 가지고 진행하였다. 처음에 U-campus를 Parsing 하는 과정에서 많은 어려움을 겪었다. 로그인 유지를 위한 Cookie 값 받기, 그리고 U-campus가 하나의 HTML 파일로 되어 있는 것이 아닌 각각의 페이지마다 다른 HTML 파일을 가지고 있어 이러한 주소를 찾는 것부터도 많은 어려움을 겪었다. 또한 각각의 페이지마다 요청하는 data 값들도 달라서 해당 data 값을 찾고 하는데도 많은 어려움을 겪었다. 그리고 가장 큰 문제는 U-campus의 경우, 일정 시간이 지나면 Cookie 값 중 하나인 JSESSIONID 값이 reload 되는 문제가 있어 이것 또한 어떤 방식으로 처리를 해주어야 될까라는 고민을 많이 할 수 있었다. 그러나 이러한 문제를 하나씩 해결하면서 PYTHON을 공부하는데 많은 도움을 얻을 수 있었다. 또한 프로젝트를 진행하면서 다른 컴퓨터 언어들과 다

르게 많은 module 들이 존재하여 처음 컴퓨터 언어를 배우려는 입문자들에게 가장 쉽게 접근할 수 있는 언어가 아닐까 라는 생각을 가지게 되었다.

해당 PYTHON code를 계속해서 돌려줄 Server를 구하지 못하여 올리지 못해, 매번 안드로이드 application의 socket 주소를 바꿔줘야 된다.



Android Studio 내에 존재하는 MainActivity를 Click

```
public class MainActivity extends AppCompatActivity {  
  
    Button btn;  
    TextView tv;  
    EditText UsernameEt, PasswordEt;  
    AlertDialog alertDialog;  
    BackPressCloseHandler backPressCloseHandler = new BackPressCloseHandler(this);  
    // TCP연결 관련  
    private Socket clientSocket;  
    private BufferedReader socketIn;  
    private PrintWriter socketOut;  
    private int port = 7424;  
    private final String ip = "192.168.0.239";  
    private MyHandler myHandler;  
    private MyThread myThread;  
    SharedPreferences mPref;  
    String userId;  
    String pass;  
}
```

MainActivity의 private final String ip 값을 현재 사용 중인 ip 주소로 변환 해야됨. 안드로이드 기기 또한 같은 ip 주소를 사용해야 됨.



Run 버튼을 이용하여 다시 application을 핸드폰과 연결하여 download 해야 됨.

6. 참고문서

<https://github.com/pompobit/PyFCM> -pyfcm 관련

<http://pydjango.tistory.com/entry/CODE-%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EC%86%8C%EC%BC%93-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D>

- socket 통신 관련

<http://hurderella.tistory.com/113> - beautifulsoup 관련

<https://firebase.google.com/docs/cloud-messaging/?hl=ko> -fcm 관련

7. 첨부파일 목록

구동 영상

PYTON code(2012722041_김승훈.py)

Android application project(U-campus)

android apk(app-debug.apk),

결과 보고서