**[5110135 캡스톤디자인 2020S]**

**Progress Report**

팀 번호: 2팀

진행 기간: 2020. 5. 7. ~ 2020. 5. 13

|  |
| --- |
| 주차별 진행 내용 및 결과를 자유롭게 작성. 분량 제한은 없으나 본 과제도 이번 학기 평가의 대상임에 유의. **주차별 목표**는 3주차 작성 내용과 완전히 일치할 필요는 없으므로 진행 상황에 맞게 지정. **진행 결과**는 주차별 목표 항목과 대응하도록 작성. 자료 수집을 한 경우 자료 수집 내용을 작성하고, 개발을 진행한 경우 Github에 commit한 내역을 캡쳐하여 올릴 것. |

**1. 프로젝트 주제: 비콘을 이용한 의류 매장 서비스**

**2. 주차별 목표 및 달성률**

|  |  |  |
| --- | --- | --- |
| 주차별 목표 | 담당자 | 달성률 |
| (1) 안드로이드 스튜디오 공부, 오픈소스 활용 방안 모색  (2) 안드로이드 공부  (3) Python | 이시은  김소연  박병조 | 100%  100%  100% |

**3. 진행 결과**

**[ 제작 전 안드로이드 스튜디오 구조 공부하기 ]**

**담당자 이시은 Progress Report 진행 현황**

**목적**

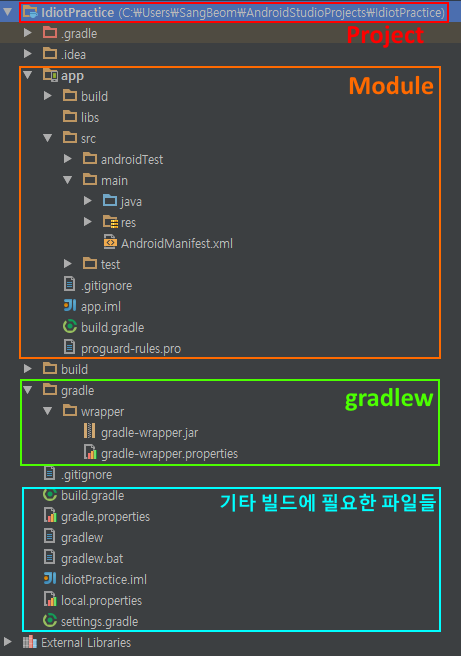
안드로이드 스튜디오의 구조를 공부함으로써 어플의 전반적인 구성을 배울 수 있고

협업 중 원할한 소통을 위해 필수적인 부분이라는 생각이 들어서 학습을 진행해 보았다.

**진행 내용**

프로젝트를 생성했을 때 만들어지는 기본적인 구조를 보고 각 폴더가 의미하는 내용과 기능들을 정리해보며 학습을 진행 해보았다.

1. **프로젝트의 기본적인 구조**



1. **idea폴더의 역할**

어플리케이션 모듈 디렉토리와 파일

1.  assetWizardSettings.xml  - 예를들어 Vector Asset 에서 icon을 추가했다면 마지막으로 추가한 아이콘의 정보를 저장함

2.  build\_file\_checksums.ser  - build.gradle, settings.gradle, gradle.properties, .gradle/ 의 환경설정 관련 파일들이 변경되었는지 여부를 파악하는 파일. 즉, 이 값이 변경되면 sync가 필요하다는 뜻.

3.  codeStyles 폴더  - 프로젝트 코드스타일 정보.

4.  libraries 폴더  - 다운로드 받은 라이브러리의 jar파일이 저장된 위치가 들어있다. 사용자별로 다르므로 반드시 ignore에 추가

5.  gradle.xml  - gradle 버전에 대한 로컬 경로가 들어있음. 사용자별로 다를수있다

6.  misc.xml  - 자바버전, 프로젝트 유형과 같은 프로젝트 정보가 들어있음. 사용자별로 다를수있다

7.  modules.xml  - iml 파일의 경로가 들어있음. 사용자별로 다를수있음

8.  navEditor.xml  - 아키텍처컴포넌트의 네비게이션에 관련된 파일. Element의 위치를 네비게이션 에디터에 저장한다. 네비게이션을 사용한다면 보관해야함

9.  runConfigurations.xml  - 구성편집해서 추가한 구성들이 저장된다. 추가된것이 있다면 보관해야함

10.  vcs.xml  - GUI를 사용하여 버전 관리 작업을 수행할 수 있도록 프로젝트에서 사용하는 VCS에 대한 정보가 들어있음.

11.  workspace.xml  - 열어본 파일의 커서의 마지막 위치 같은 Android Studio의 작업 영역에 대한 정보가 포함되어있음. 사용자마다 다를수 있음  
출처: <https://duzi077.tistory.com/276> [개발하는 두더지]

1. **bulid 폴더 역할**

이 다렉토리는 모든 프로젝트 모듈들의 빌드 결과를 저장함

1. **gradle 폴더 역할**

기본값으로는 그레이들 환경을 모아놓은 wrapper 파일이라고 하는 파일들이 들어 있다.

**gradle이란?**

Gradle은 Groovy를 이용한 [빌드 자동화](https://ko.wikipedia.org/wiki/%EB%B9%8C%EB%93%9C_%EC%9E%90%EB%8F%99%ED%99%94) 시스템이다. Groovy와 유사한 도메인 언어를 채용하였으며, 현재 [안드로이드](https://ko.wikipedia.org/wiki/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C_(%EC%9A%B4%EC%98%81_%EC%B2%B4%EC%A0%9C)) 앱을 만드는데 필요한 안드로이드 스튜디오의 공식 빌드 시스템이기도 하다. Java, [C](https://ko.wikipedia.org/wiki/C_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D_%EC%96%B8%EC%96%B4))/[C++](https://ko.wikipedia.org/wiki/C%2B%2B), [파이썬](https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%B4%EC%8D%AC) 등과 같은 여러 가지 언어를 지원한다.

**Groovy란?**

[공개 소스 소프트웨어](https://terms.naver.com/entry.nhn?docId=864845&ref=y)로 개발되는 최근의 객체 지향 언어로서, 기존 언어와는 달리 변수 타입이 동적(dynamical)이고 유연하며, 소스 파일을 [컴파일](https://terms.naver.com/entry.nhn?docId=818391&ref=y)하지 않고 바로 실행시키는 [스크립팅 언어](https://terms.naver.com/entry.nhn?docId=848219&ref=y).

1. **.gitignore**

Git이 무시해야 하는 검색되지 않는 파일을 지정한다.

1. **build.gradle**

gradle이라는 빌드 시스템의 설정 파일이다. 여기에 프로젝트의 전체 설정이 기술된다 모듈 안에도 build.gradle파일이 있는데, 모듈안의 build gradle에는 모듈에 대한 설정이 기술된다. 구체적으로는 android gradle plugin 버전과 maven 리포터리 설정 등이 여기 해당된다.

1. **gradle.properties**

프로젝트에서 널리 쓰이는 gradle의 세팅들을 포함한다.

1. **Gradlew/gradlew.bat**

gradle wrapper를 통해 빌드를 진행하기 위한 스크립트이다.

**Gradlelew?**

GradleWrapper이라는 메커니즘이 있어서 gradle의 버전을 지정해서 빌드할 수 있다. 프로젝트 팀이 같은 버전의 그레이들을 이용하고 싶은 경우에 사용할 수 있다.

**Gradle wrapper란?**

빌드에서 사용하는 gradle 툴을 편리하게 사용할 수 있도록 도와주는 일종의 스크립트로 적절한 gradle 바이너리를 자동으로 다운로드 해주며, 별도로 gradle을 설치하지 않아도 빌드를 진행할 수 있게 해준다.

1. **local.properties**

빌드를 진행할 때 필요한 환경 변수 정보를 저장한다. 안드로이드 SDK의 경로가 여기 저장된다.

1. **.iml**

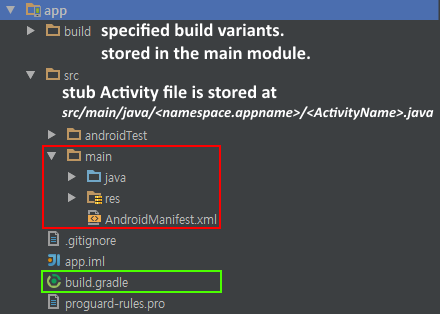
안드로이드에서 사용하는 프로젝트 설정 파일이다.

**(10) settings.gradle**

빌드와 관련된 환경설정 및 같이 빌드 되어야 할 하위 모듈들의 정보를 포함한다.

1. **안드로이드 어플리케이션 모듈**
2. **모듈이란?**

어플리케이션이나 라이브러리를 구성하는 최소 단위로 독립적으로 존재할 수 없고, 항상 프로젝트 내에 포함되어야 한다.



1. build : 빌드 과정에서 생성된 파일 및 산출물을 저장한다
2. src : 소스 및 리소스 파일을 저장한다
3. androidmainfest.xml : 이 파일은 응용프로그램의 특성과 각 구성요소를 설명한다.
4. build.gradle : 모듈의 빌드 방법이 정의된 빌드 스크립트, 빌드에 사용할 SDK 버전, 어플리케이션 버전, 사용하는 라이브러리 등 다양한 항목 설정 가능
5. .iml : 안드로이드 스튜디오에서 사용하는 모듈 설정 정보이다.

**[ 구현해야 할 기능과 오픈소스 활용 방안 ]**

**목적**

기존에 작성한 IA를 바탕으로 앞으로의 설계에 있어서 구현해야 할 요소들에 적용할 수 있는 오픈 소스들을 찾아보며 전반적인 제작 흐름을 유추하며 제작과정에 어려움이 있을 만한 요소들을 미리 생각해볼 수 있다.

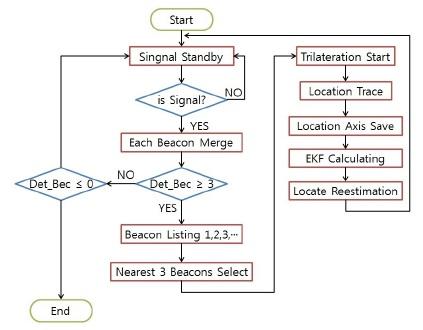
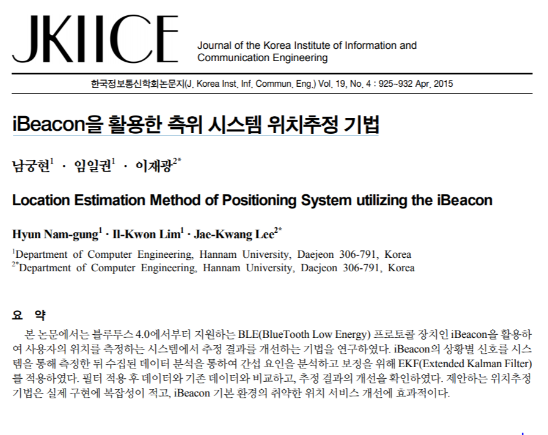
**진행 내용**

IA를 기반으로 구현해야 할 기능들을 작성하고 그와 관련되어 적용할 수 있는 오픈소스를 조사했다. 기존의 오픈소스를 프로젝트에 적용시켰을 때 어떤 점을 수정해야할 지도 고려해보았다.

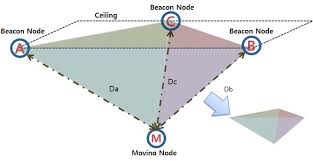


1. **EKF 알고리즘을 이용한 현재 위치 추정**

: EKF가 적용된 위치추정 알고리즘으로 주위에 3개의 비콘이 있으면 위치를 측위하는 알고리즘 사용한다.



< 논문참고 > iBeacon을 활용한 측위 시스템 위치 추정 기법



이때 삼각 측량 기법이 위치 측위에 사용된다.

BLE Beacon 삼각 측량 Git hub 라이브러리 <https://github.com/lemmingapex/Trilateration>

1. **실내 지도에 사용자 위치 표시**

: 실내 지도에 현재위치를 표시하려면, 지도 컴포넌트(TMapView)와 GPS 센서 컴포넌트(TLocationSensor)를 이용해야 한다

1. **지도 컴퍼넌트 참고 자료**

<http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Mobile_Tutorial:_Using_a_Map_Component_to_Work_with_Maps_(iOS_and_Android)>

1. **GPS 센서 컴퍼넌트 참고 자료**

<http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Mobile_Tutorial:_Using_Location_Sensors_(iOS_and_Android)>

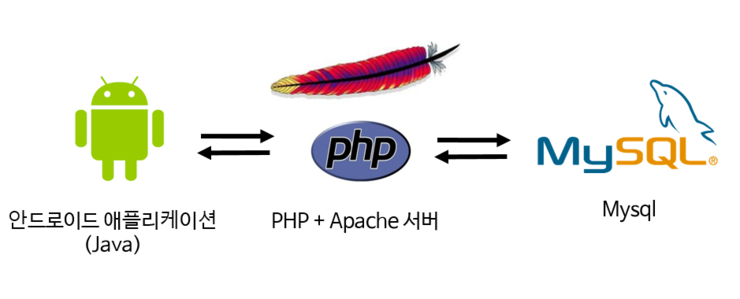
1. **선택한 옷이 걸린 행거의 위치정보 실내 지도에 표시**

: 위와 동일한 방식으로 구현

1. **옷 정보와 옷 재고를 등록을 위한 관리자 모드 데이터 베이스 관리**

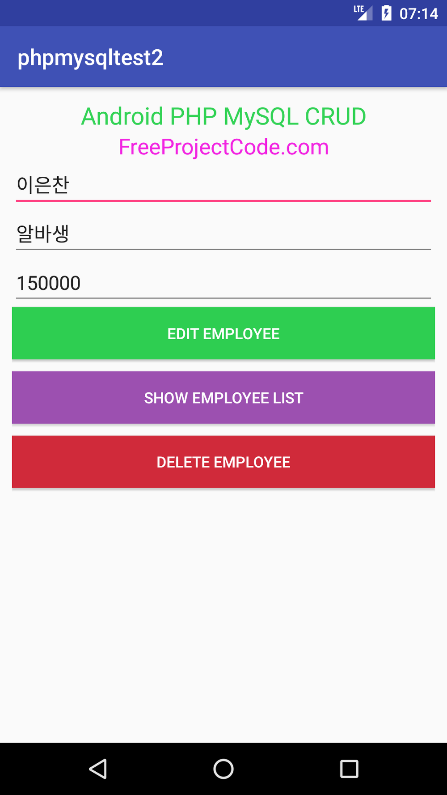
: 안드로이드 DB 읽기, 쓰기, 삭제, 수정 메서드 이용

1. **mysql을 사용한 데이터베이스**



<참고해볼 강의> <https://www.youtube.com/watch?v=cX37pptPBpA>

1. **안드로이드 DB 읽기, 쓰기, 삭제, 수정 메서드**
2. 읽기 : 입력한 데이터를 읽는다. **> 사용자들이 매장에 있는 옷들에 관한 정보를 볼 때 저장된 데이터를 보여줌.**
3. 쓰기 : 입력을 통해 데이터를 입력한다 > **매장직원이 매장에 등록된 옷을 업데이트 할 때 이용**
4. 삭제 : 입력한 데이터를 삭제한다. > **없어지는 옷이나 필요 없는 데이터들에 관한 삭제**
5. 수정 : 입력한 데이터들을 수정한다. > **재고 관리할 때 이용**



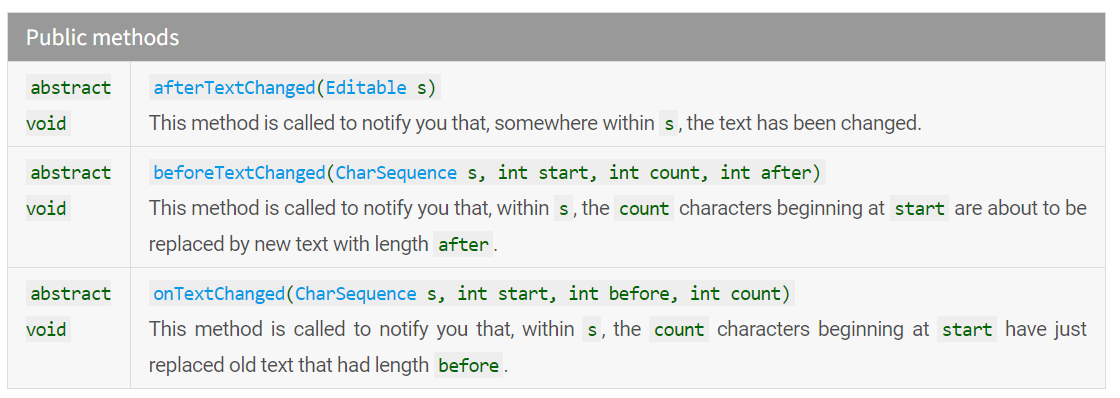
<구현할 때 참고 할 자료>

<https://ilbbang.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-DB-CRUD%EC%93%B0%EA%B8%B0-%EC%9D%BD%EA%B8%B0-%EC%88%98%EC%A0%95-%EC%82%AD%EC%A0%9C-4-Android-%ED%99%94%EB%A9%B4-%EB%B0%8F-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0>

1. **원하는 옷을 검색**

: 매장에 등록된 옷들의 데이터베이스 안에서 내가 원하는 옷을 검색할 수 있어야 함.

커스텀 리스트 뷰에 검색기능을 추가하는 방식



EditText의 **TextChangeListener**를 활용한 방식으로 구현. After인 경우 글자가 바뀌고 난 후이고 before인 경우는 글짜가 바뀌기 전 on인 경우 바뀌는 순간에 실행된다.

<참고 자료> <https://cholol.tistory.com/395>

1. **음성인식 구현하고 음성으로 작동되는 명령**
2. **음성인식 구현**

: 마이크로 목소리 녹음 > 구글한테 전송 > 음성 인식 결과 수신 하는 방식

< 참고 자료 >

<https://m.blog.naver.com/PostView.nhn?blogId=dt3141592&logNo=221184533382&proxyReferer=https:%2F%2Fwww.google.com%2F>

<https://webnautes.tistory.com/1247>

1. **음성액션 사용**

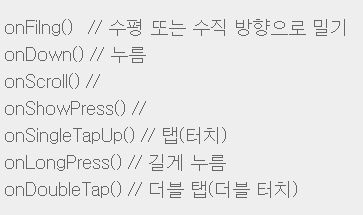
* 시스템 제공 음성 액션 선언
* 앱 제공 음성 액션 선언
* 자유형식 음성 입력 얻기

< 참고자료>

<https://developer.android.com/training/wearables/apps/voice?hl=ko>

1. **제스처 감지**
2. **일반 제스처 감지와 처리하기**

onFling(), onDown(), onScroll(), onShowPress(), onSingleTapUp(), onLongPress() 콜백 메서드를 포함하는 GetstureDetector, OnGetstureListener 인터페이스를 구현하는 클래스를 정의함.



< 참고 자료>

<https://developer.android.com/training/gestures/scroll?hl=ko>

<https://insurang.tistory.com/141>

1. **제스처를 감지하고 직원에게 알람이 가도록 이벤트 처리**
2. **푸시 알람 보내기**
3. **FCM을 이용한 푸시 알람보내기**

< 참고 자료> <http://blog.naver.com/PostView.nhn?blogId=ndb796&logNo=221553341369>

1. **서버없이 푸시 알람 보내기**

: NotificationManager 클래스 사용

**Notification 메서드**

 - setSmallIcon : 푸시 알림 왼쪽 그림

 - setTicker : 알람 발생시 잠깐 나오는 텍스트

 - setWhen : 푸시 알림 시간 miliSecond 단위 설정

- setNumber : 확인하지 않은 알림 개수 표시 설정

 - setContetnTitle : 푸시 알림 상단 텍스트(제목)

 - setContentText : 푸시 알림 내용

 - setDefaults : 푸시 알림 발생시 진동, 사운드 등 설정

 - setContentIntent : 푸시 알림 터치시 실행할 작업 인텐트 설정

 - setAutoCancel : 푸시 알림 터치시 자동 삭제 설정

 - setOngoing : 푸시 알림을 지속적으로 띄울 것인지 설정

**[ 안드로이드 개발 공부 ]**

**담당자 김소연 Progress Report 진행 현황**

**[ Activity 수명 주기 ]**

사용자가 앱을 사용하다가 나가고, 다시 앱으로 돌아오는 등의 활동을 할 때 앱의 Activity 인스턴스는 수명 주기 안에서 서로 다른 상태로 전환된다. 따라서 이를 위해 Activity 클래스는 Activity가 상태변화를 알아차릴 수 있는 여러 콜백을 제공한다.

Activity 수명 주기 상태 간의 전환을 위해서 Activity 클래스는 6가지 콜백을 제공한다. 시스템은Activity가 새로운 상태로 들어가면 각 콜백을 호출한다. 아래의 <그림 1>은 Activity의 수명 주기를 간략히 표현한 그림이다.



<그림 1> Activity 수명 주기

사용자가 Activity를 떠나면 시스템은 Activity를 해체할 메서드를 호출한다. 이때 Activity는 여전히 메모리 안에 남아있으며 포그라운드로 다시 돌아올 수 있다. 사용자가 해당 Activity로 돌아오는 경우 Activity는 사용자가 종료한 지점에서 다시 시작된다.

시스템은 그 시점의 Activity 상태에 따라 특정 프로세스와 그 안의 Activity를 함께 종료할지 여부를 결정한다.

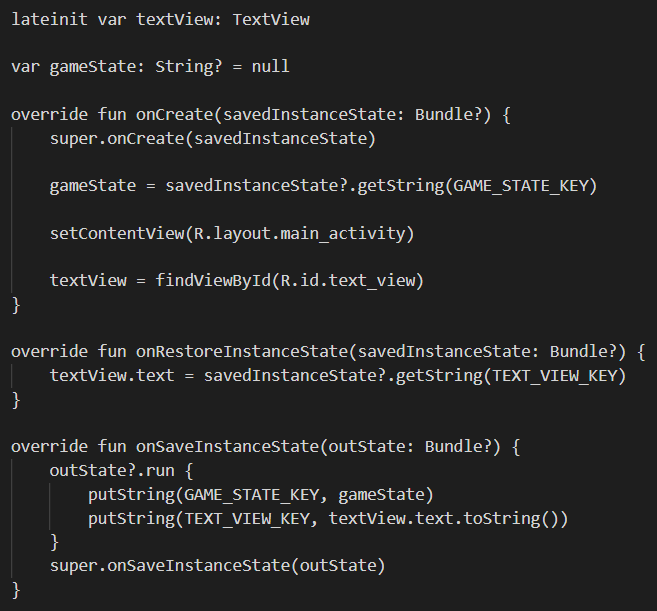
1. onCreate()

* 이 콜백은 시스템이 Activity를 생성할 때 실행되는 것으로 필수적으로 구현되어야 한다.
* Activity가 생성되면 Activity는 “생성”상태가 된다.
* onCreate() 메서드에서는 Activity의 전체 수명 주기 동안 한 번만 실행 되어야하는 기본 애플리케이션 시작 로직을 수행한다.

예시)

* + 데이터를 목록에 바인딩하고 Activity를 viewModel과 연결
  + 일부 클래스 범위 변수를 인스턴스화
  + Activity의 수명 주기와 연결된 수명 주기를 인식하는 구성요소가 있다면 이 구성요소는 ON\_CREATE 이벤트를 수신

아래의 코드는 onCreate() 메서드에 대한 예시로 사용자 인터페이스 선언, 멤버 변수 정의, 일부 UI 구성 등의 Activity에 대한 기본 설정을 보여준다.

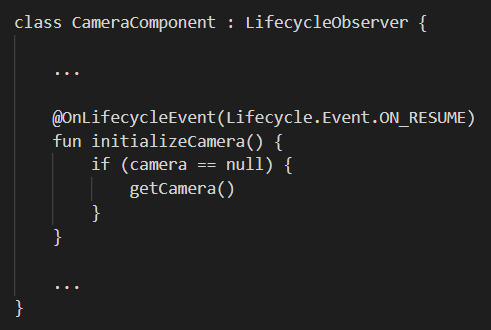


1. onStart()

* Activity가 “시작”상태에 들어가면 시스템은 이 콜백을 호출한다.
* onStart()가 호출되면 Activity가 사용자에게 보이게 되고, 앱은 Activity를 포그라운드에 보내 상호작용할 수 있도록 준비한다.
* Activity가 시작됨 상태로 전환되면 이 Activity의 수명 주기와 연결된 모든 수명 주기 인식 구성요소는 ON\_START 이벤트를 수신한다.
* onStart() 메서드는 매우 빠르게 완료되고, “생성됨”상태와 마찬가지로 Activity가 “시작됨” 상태에 머무르지 않는다.
* 이 콜백이 완료되면 Activity가 “재개됨”상태에 들어가고, 시스템이 onResume() 메서드를 호출한다.

1. onResume()

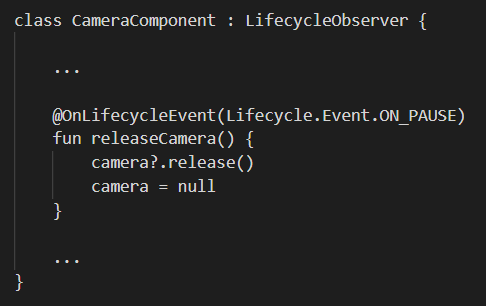
* Activity가 “재개”상태에 들어가면 포그라운드에 표시되고 시스템이 onResume() 콜백을 호출한다.
* 이 상태에 있을 때는 사용자와 앱이 상호작용을 한다.
* 어떤 이벤트가 발생하여 앱에서 포커스가 떠날 때까지 앱이 이 상태에 머무른다.  
  (사용자가 다른 Activity로 이동하거나, 기기 화면이 꺼지는 등의 이벤트)
* Activity가 재개됨 상태로 전환되면 이 Activity의 수명 주기와 연결된 모든 수명 주기 인식 구성요소는 ON\_RESUME 이벤트를 수신한다.
* 이 상태에서 수명 주기 구성요소가 포그라운드에서 사용자에게 보이는 동안 실행해야 하는 모든 기능을 활성화할 수 있다. (카메라 미리보기 시작 등)
* 방해되는 이벤트가 발생하면 Activity는 “일시정지”상태에 들어가고, 시스템이 onPause() 콜백을 호출한다.
* Activity가 “일시정지”상태에서 “재개”상태로 돌아오면 시스템이 onResume() 메서드를 다시 한번 호출한다. 따라서 onResume()을 구현하여 onPause() 중에 해제하는 구성요소를 초기화하고, Activity가 “재개”상태로 전환될 때마다 필요한 다른 초기화 작업도 수행해야 한다.



위는 LifecycleObserver가 ON\_RESUME 이벤트를 수신하면 카메라를 초기화하는 코드의 예시이다.

1. onPause()

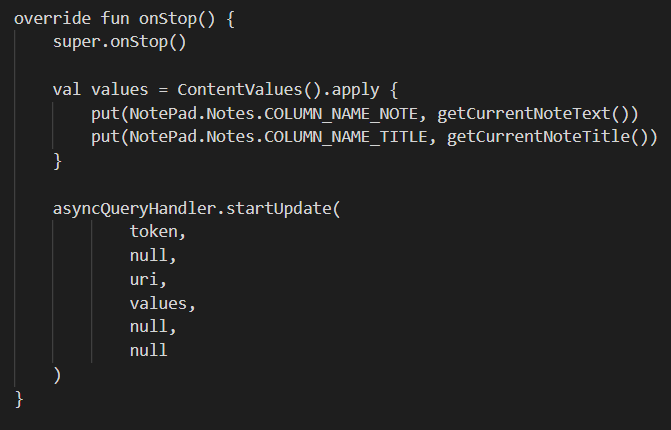
* 시스템은 사용자가 Activity를 떠나면 이 메서드를 호출한다.
* 이때 해당 Activity가 항상 소멸되는 것은 아니고 Activity가 포그라운드에 있지 않게 되었다는 것을 나타낸다.
* Activity가 이 상태에 들어가는 이유들은 다음과 같다.
  + 일부 이벤트가 앱 실행을 방해할 때
  + Android 7.0 (API 레벨 24) 이상에서는 여러 앱이 멀티 윈도우 모드에서 실행된다. 그 중 하나의 앱만 포커스를 가질 수 있기 때문에 시스템이 그 외의 모든 다른 앱을 일시정지시킨다.
  + 대화상자와 같은 새로운 반투명 Activity가 열릴 때
* Activity가 “일시정지”상태로 전환되면 이 Activity의 수명 주기와 연결된 모든 수명 주기 인식 구성요소는 ON\_PAUSE 이벤트를 수신한다.
* 이때 수명 주기 구성요소가 포그라운드에 있지 않을 때 실행할 필요가 없는 기능을 모두 정지할 수 있다.
* onPause() 메서드를 사용하여 시스템 리소스, GPS와 같은 센서 핸들 또는 Activity가 일시정지 중이고 사용자가 필요로 하지 않을 때 배터리 수명에 영향을 미칠 수 있는 모든 리소스를 해제할 수 있다.
* 일시정지된 Activity는 멀티 윈도우 모드에서 계속 화면에 보이는 상태일 수 있으므로 멀티 윈도우 모드를 더 잘 지원하기 위해서는 UI 관련 리소스와 작업을 완전히 해제하거나 조정할 때는 onPause() 대신 onStop()을 사용하는 것이 좋다.



위의 코드는 LifecycleObserver가 ON\_PAUSE 이벤트를 수신한 이후에 카메라 해제하는 코드의 예시이다.

1. onStop()

* Activiry가 사용자에게 더 이상 표시되지 않으면 “중단”상태에 들어가고, 시스템은 onStop() 콜백을 호출한다.
* 이는 새로 시작된 Activity가 화면 전체를 차지할 경우 적용된다.
* Activity의 실행이 완료되어 종료될 시점에 onStop()을 호출할 수도 있다.
* Activity가 “중단”상태로 전환되면 이 Activity의 수명 주기와 연결된 모든 수명 주기 인식 구성요소는 ON\_STOP 이벤트를 수신한다.
* onStop() 메서드에서는 앱이 사용자에게 보이지 않는 동안 앱에 필요하지 않은 리소스를 해제하거나 조정해야 한다.  
  (애니메이션을 정지 등)
* 사용자가 멀티 윈도우 모드에서 Activity를 보고 있더라도 UI 관련 작업이 계속 진행된다.
* onStop()을 사용하여 CPU를 비교적 많이 소모하는 종료 작업을 실행해야 한다.



위의 코드는 onStop() 메서드에서 초안 내용을 SQLite를 직접 사용하여 영구 저장소에 저장하는 코드의 예시이다.

1. onDestroy()

* onDestroy()는 Activity가 소멸되기 전에 호출된다.
* 시스템은 아래와 같은 상황에서 onDestroy() 콜백을 호출한다.
  + 사용자가 Activity를 완전히 닫거나 Activity에서 finish()가 호출되어 Activity가 종료되는 경우
  + 기기 회전 또는 멀티 윈도우 모드 등의 구성 변경으로 인해 시스템이 일시적으로 Activity를 소멸시키는 경우
* Activity가 “소멸”상태로 전환되면 이 Activity의 수명 주기와 연결된 모든 수명 주기 인식 구성요소는 ON\_DESTROY 이벤트를 수신한다.
* Activity가 종료되는 경우 onDestry()는 Activity가 수신하는 마지막 수명 주기 콜백이 된다.
* 구성 변경으로 인해 onDestroy()가 호출되는 경우 시스템이 즉시 새 Activity 인스턴스를 생성한 다음, 새로운 구성에서 새로운 인스턴스에 대해 onCreate()를 호출한다.
* onDestroy() 콜백은 이전의 콜백에서 아직 해제되지 않은 모든 리소스를 해제해야 한다.

출처 : <https://developer.android.com/guide/components/activities/activity-lifecycle#asem>

**[ Python ]**

**담당자 박병조 Progress Report 진행 현황**

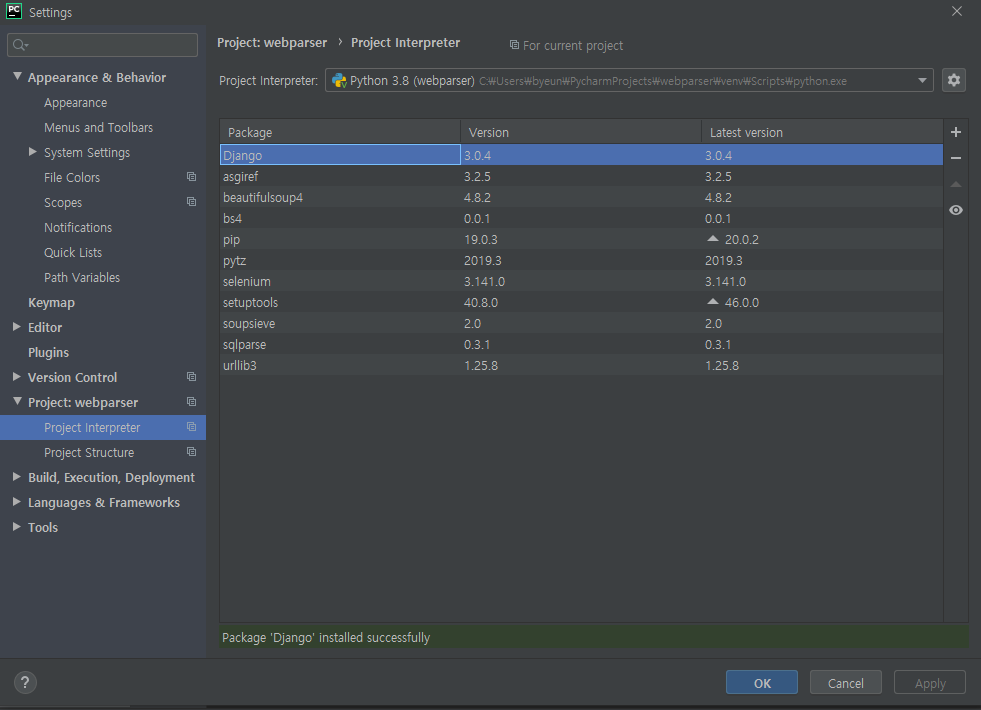
Python을 통한 서버 개설 실습

비콘과 스마트폰을 통해 받아온 정보를 통해 필요한 데이터를 전달해주기 위한 데이터 저장공간을 위한 서버를 생성을 해야하는데 가장 사용하기 편한 서버를 결정하기 위한 서버 실제 생성 및 사용

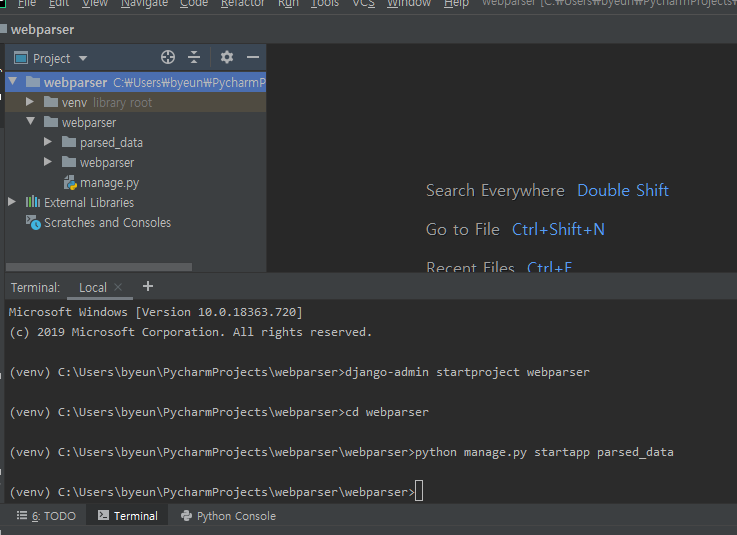
아직 프로젝트 디자인이 안나왔기 때문에 예제로 Django-Crawling Project를 통해 실습

이후 사용할 서버 변경 예정

1) 사Python의 Djang 패키지를 이용하여 서버를 개설

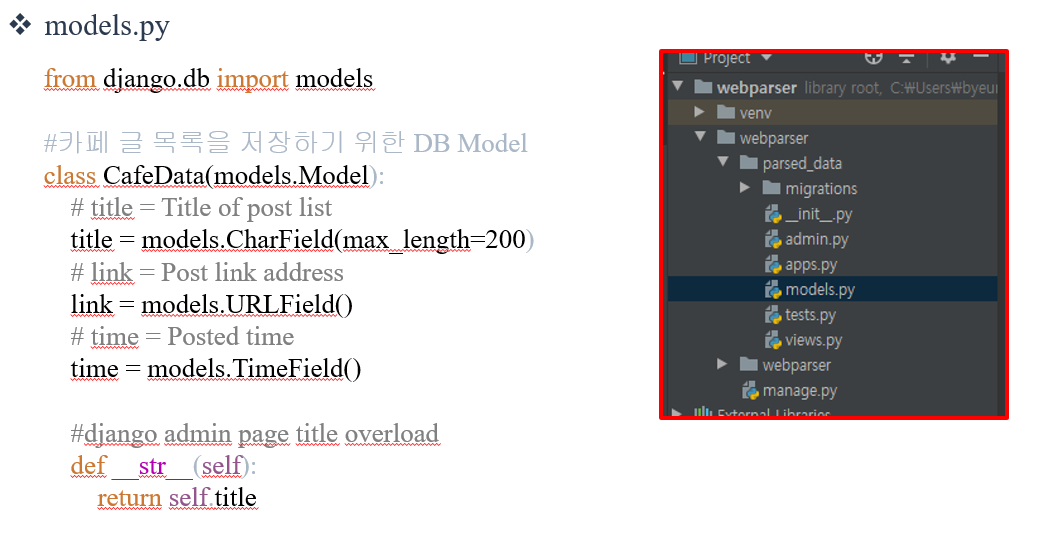
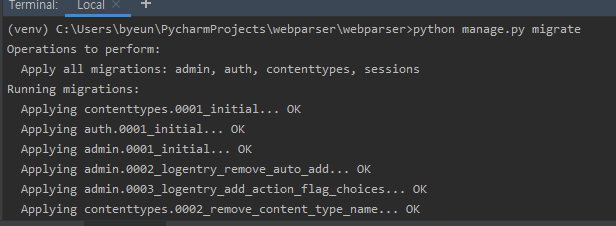


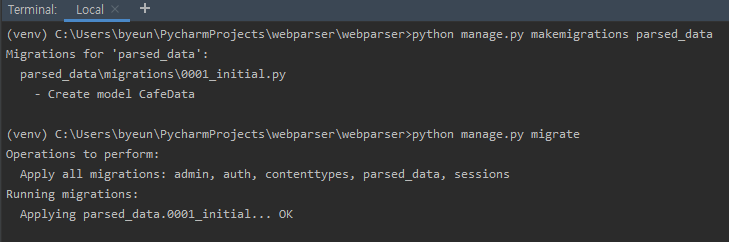
2) Django를 install 후 Project 생성



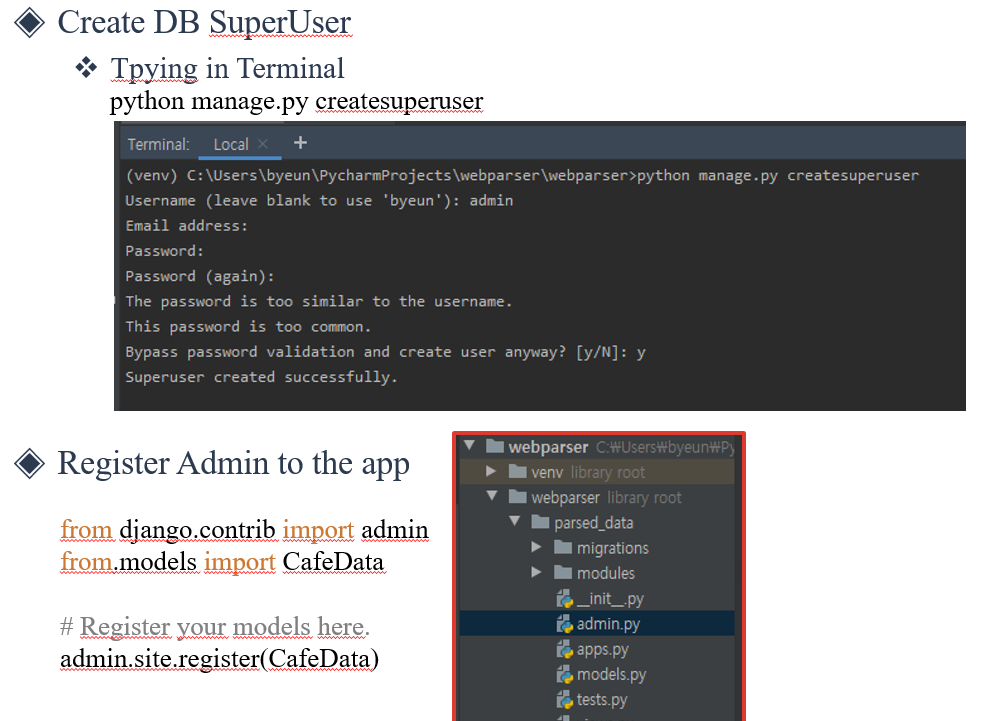
3) Project 내부에 DB 생성

임시 실습을 위한 서버이기 때문에 일단 기본적으로 Django에서 지원해주는 Splite3 DB를 이용한다.

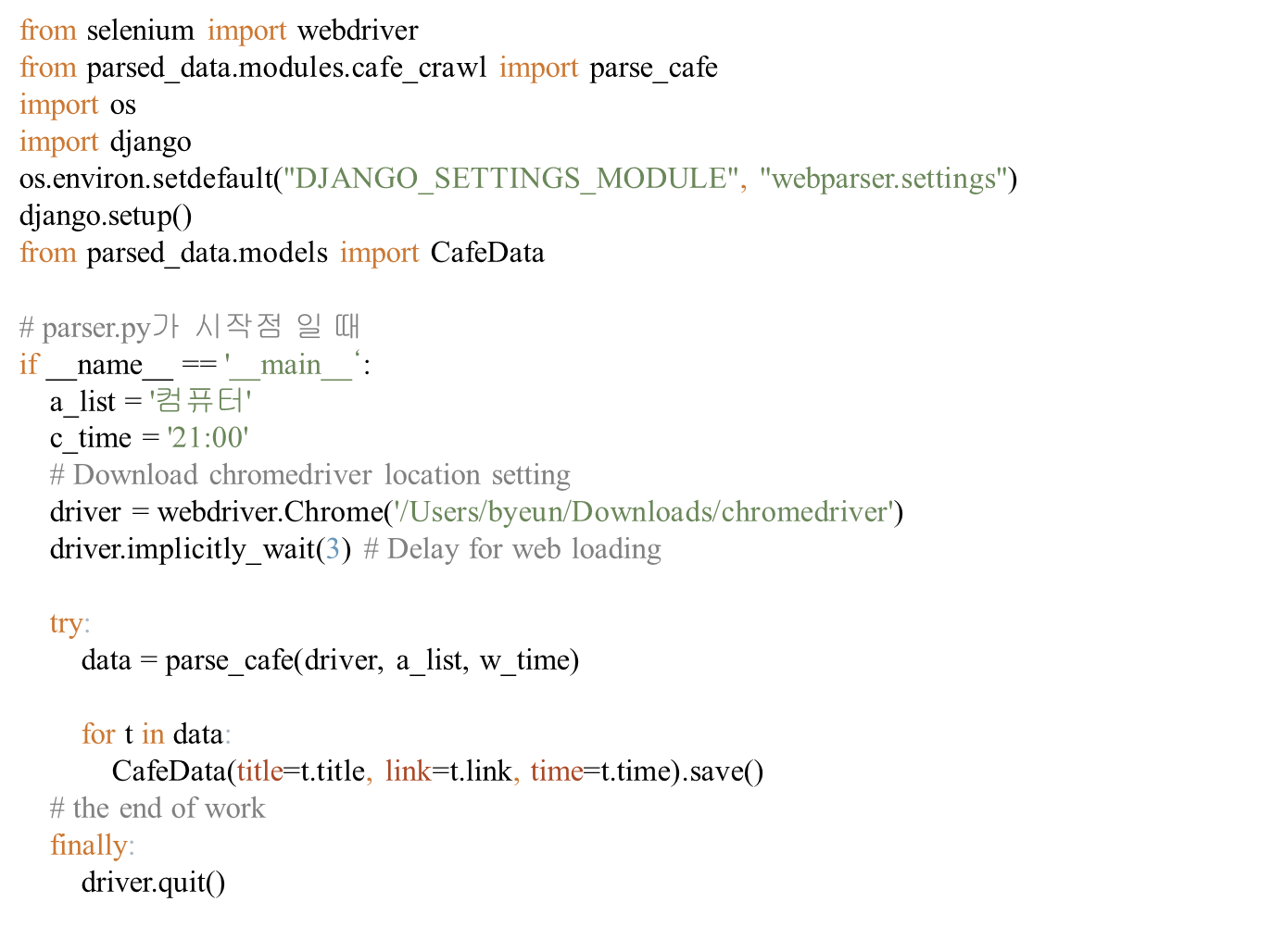




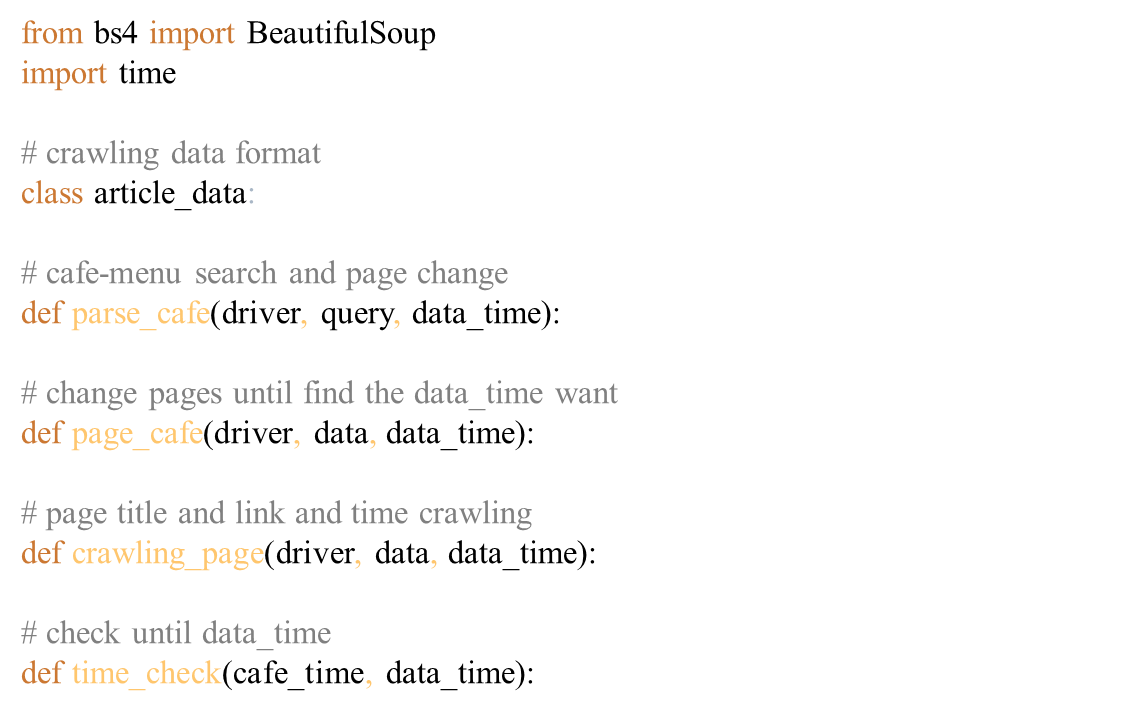
4) DB 접속을 위한 계정 생성 및 웹을 통한 DB 접속 연결



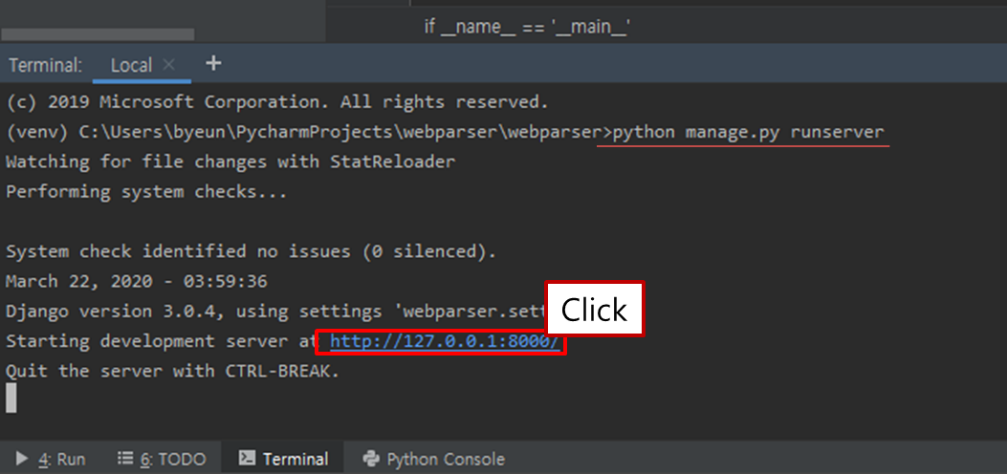
5) Crawling을 이용하여 얻은 데이터를 서버에 저장



6) 데이터 Crawling 소스 (소스가 길어 함수명과 변수만 보고서에 작성)



7) 해당 project를 실행



8) 서버 접속

