**Mobil Programlama Ödevi**

**Grup Üyeleri:**

İbrahim Kaya 1306150028

Doğaç Yıldırım 1306160045

**Özet**

Uygulamamız New Yourk Times Gazatesinin worl, science ve sport rss yayınlarını okumanızı sağlayan, sevmediğiniz haberleri sola kaydırarak o kategorideki haberleri feedinizden çıkardığınız bir yapıya sahip haber okuma uygulaması.

**Kullanılanlar**

Hpptt connection

AsyncTask

RecyclerView

Fragment

Room

SharedPrefrence

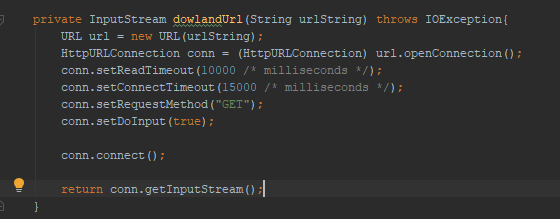
Broadcast Reciver

Notification

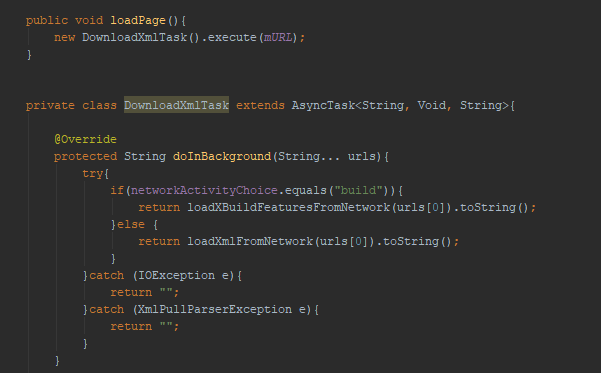
JobScheular

**1.Kısım İnternet üzerinden bağlantıların alınması**

NetworkActivity clası:

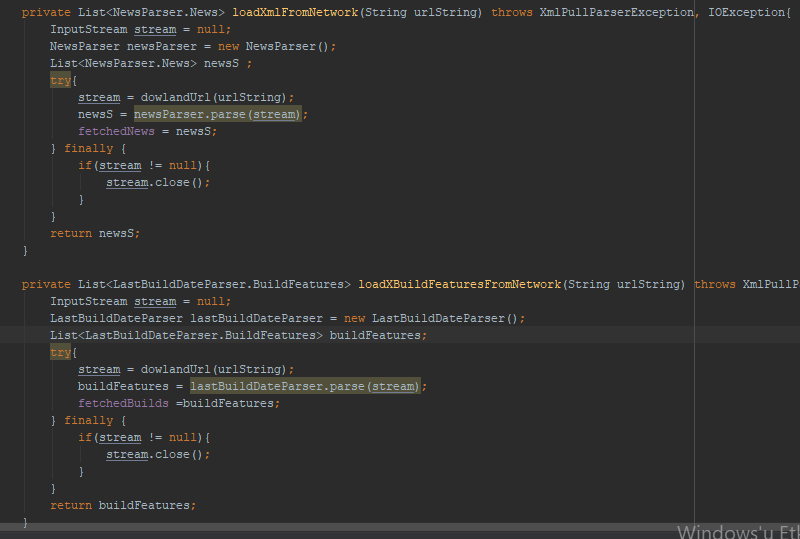


HttpUrlConnection üzerinden bir bağlantı açıp seçilen url’e göre input stream olarak bağlantıyı alıyoruz.



loadPage() metodu ile DowlandXmlTask metodumu dışarı açıyorum.

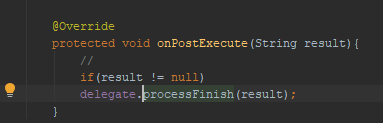
Build ve News kaynakları( yani haber içerikleri ve lastbuild date bilgilerini almak için) iki farklı parser çağıran asyncTask metodlarım var .







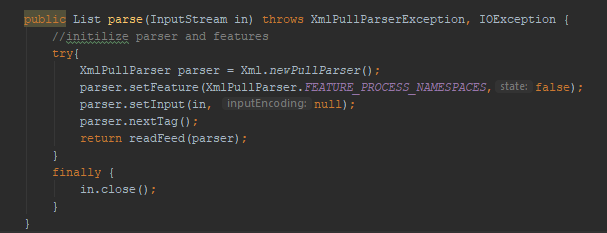
Kendi yazdığım interface sayesinde mainactivitye implement edip onpostExecute metodu tarafından tektiklenebilir bir metod oluşturuyorum.



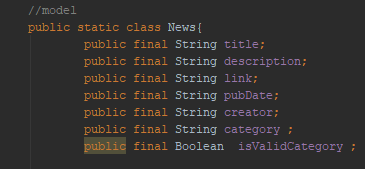
**2.Kısım inputstream’in parser tarafından işlemesi:**

NOT: dökümanda yer almayan 2 ve 3 kısımlar ,Build özelliklerinin parse edilmesi ve parse edilmesi, benzerdir.

NewsParser.class

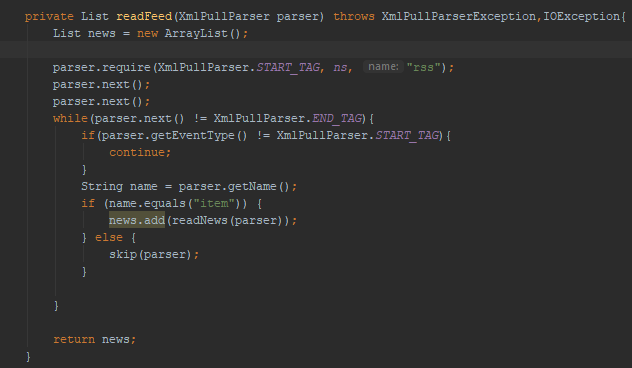


Parser’in dışarı açılan metodu ve özelliklerinin initiliaze edilmesi



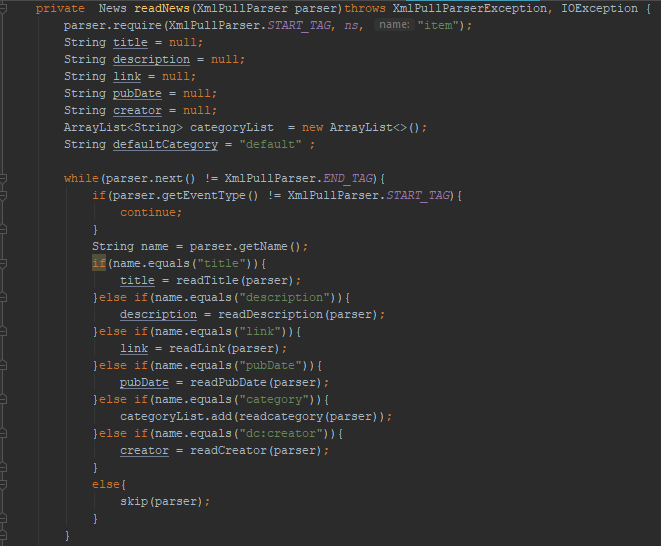
Okunan imputların saklanacağı veri modelim

isValidCategory kullanıcı müdehalesi olana kadar true olarak tutuluyor.

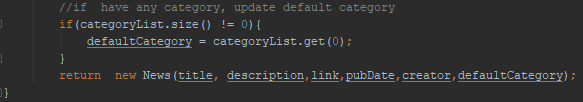


‘rss’ tagı ile okumaya başlıyorum rss yapısından dolayı benim işime yaramayan iki alt tag kısmına geçiş için .next() metodunu kullanıuyırum.

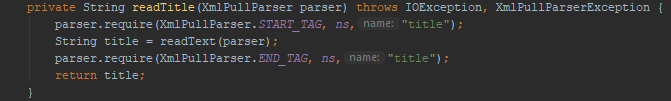
END\_TAG olmayana kadar her <item> nesnesini readNews() metodu ile pars ediyorum, START\_TAG olduğu zamanlarda bir sonraki itemi okumaya başlayabilirim.



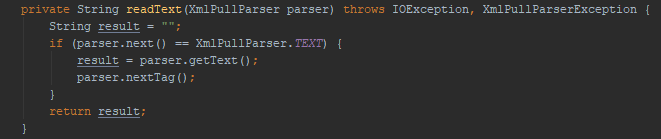
Her item nesnesinin kendi özelliklerine göre parse edileceği metodlarına yönlendirilmesi.



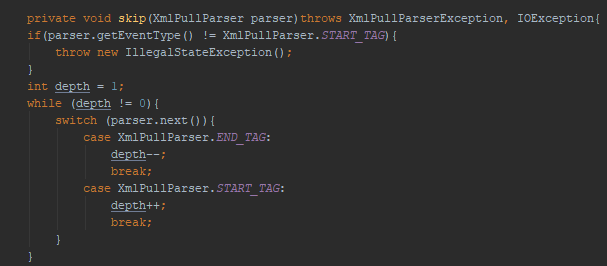
Rss den gelen birden fazla kategori olması halinde ilk kategoriyi tutuyorum.



Örnek title’ın parse edilmesi.



Text içerikli metinlerin okunması.



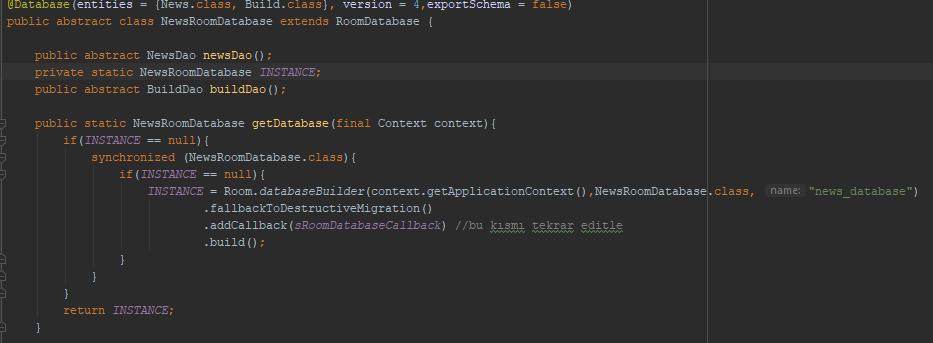
İstediğim tagler haricindeki kısımları okumamak için skip() metodu

Depth 1 den başlıyıp her START\_TAG gördüğünde bir arttırılıp END\_TAG durumda bir azaltılarak istemediğim içerikleri okumadan input streamin içindeki bilgiyi tüketmek(atlamak)

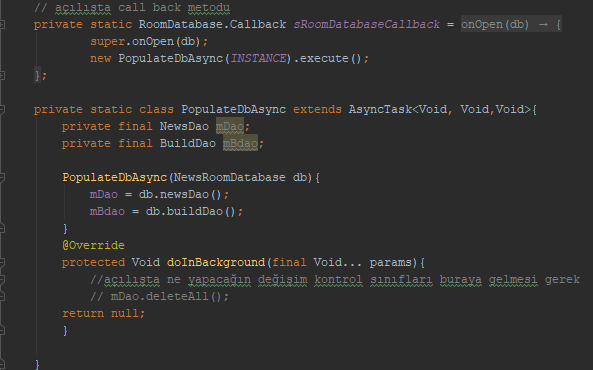
**3.Kısım gelen bilgilerin Room ile telefonda saklanması:**

**3.1 Database oluşturulması:**

NewsRoomDatabase clası, singletion bir yapıdadır,



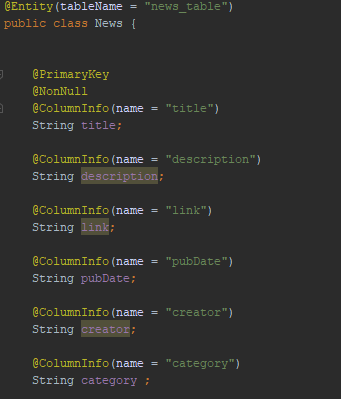
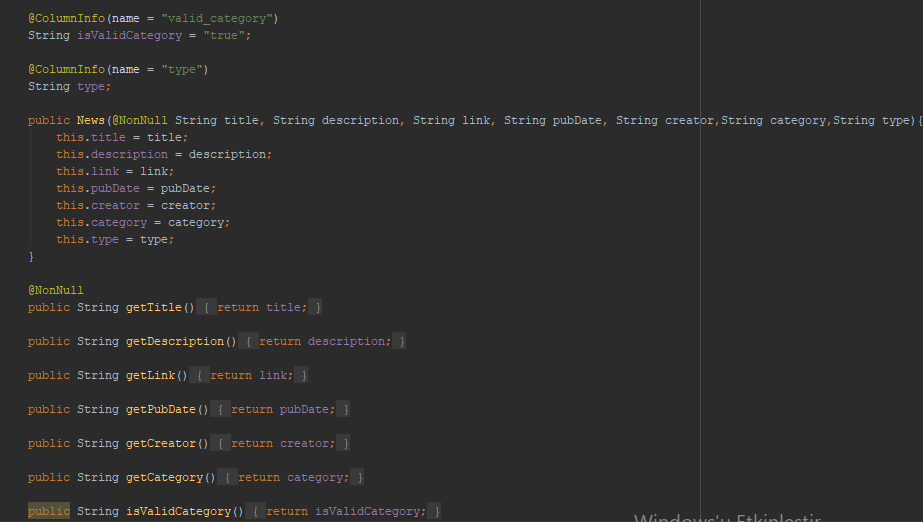
Database’in oluşturulması ve sigletion tasarıma uygun birden fazla örnek oluşturulmaması .



Açılışta çağıralacak metodlar, Data acces object (Dao) yapılarının bağlanması, eğer gerekli ise database populate yapan metod.

**3.2 Veri modelinin oluşturulması:**

News clası:

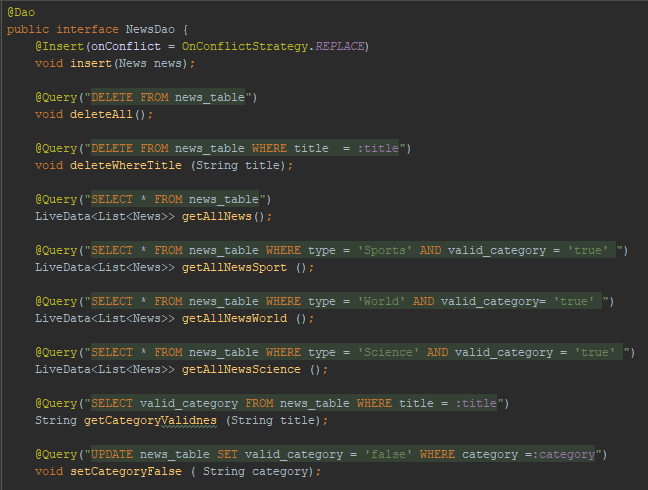
Tablodaki stunların tanımlanması,modelin constructor ve getter metodları

@CloumInfo ile değişkenin bir stun olduğunu belirliyorum.

@Entity bu clasın tablo olduğunu belirltir.

**3.3 Data Access Object oluşturulması:**

NewsDao clası, tablo üstünde yapılacak değişikliklerden sorumludur .

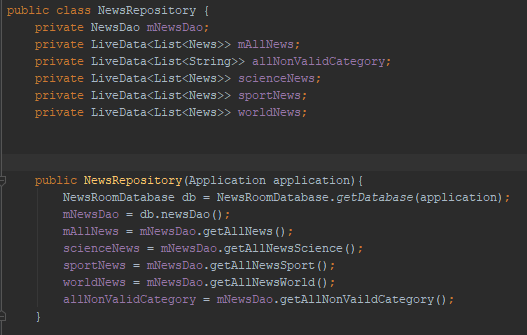


@Dao bu clasın Dao olduğunu tanımlar.

@Insert ve @Query ile metodlarımı belirlenmesi.

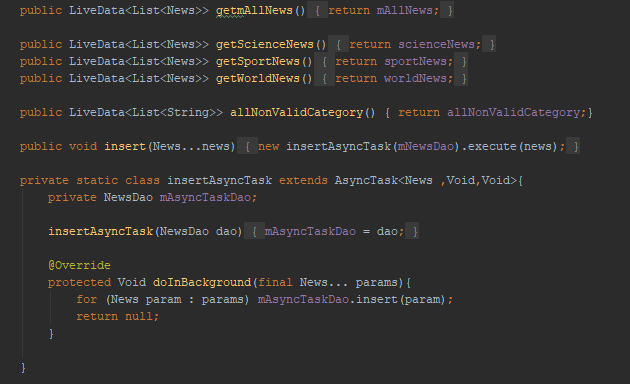
**3.4 Repository:**

NewsRepository clası:



Live dataların oluşturulması (data sourcedaki kaynağın izlenmesini sağlayan yapılar)

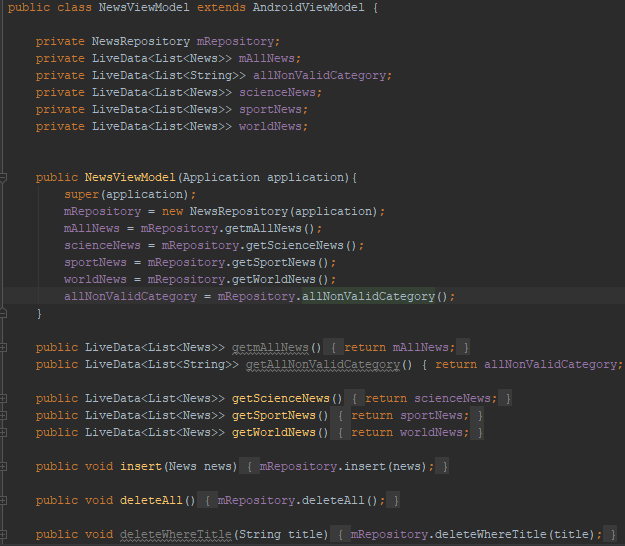
Class constructor’ı: Dao yapılarının atanması ve database bağlantısının oluşturulması.



Asenkron olarak Dao üzerinden bilgilerin database’e kaydedilmesi, NewsDao’nun sahip olduğu tüm metodlar için benzer metodlar yazılır.

**3.5 ViewModel, Verilere UI üzerinden erişilmesi için kullanılır:**

NewsViewModel class’ı, repoda olduğu gibi Dao yapsıının tüm metodlarını soyutlar.



**4 MainActivityde verilerin arayüze eklenmesi:**

**4.1 RecyclerView ve yapılarının oluşturulması:**

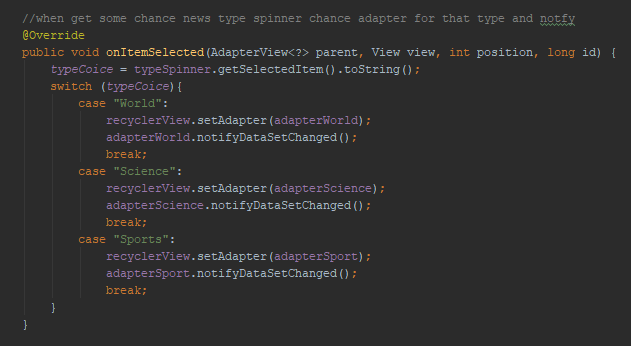
Activity\_main.xml layoutuna view olarak eklenir daha sonra bilgileri tutup viewe bağlaması için custom adapter yazılır, NewsListAdapter classı.

NewsListAdapter:

ViewHolder’ın bindViewHolder yapısı ile RcyclerView view yapısının sahip olduğu viewlere (satırlar) gösterilecek haberleri bağlar, ViewHolder classı bilgilerin tam olarak nasıl bağlanacağına karar verir.

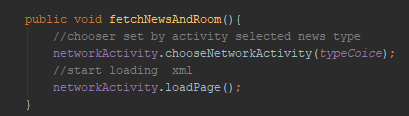
**4.2 Hangi haber tipinin yükleneceğinin edileceğinin seçilmesi**





UI da olan spinner yapısının seçimine göre type değiştirmek.

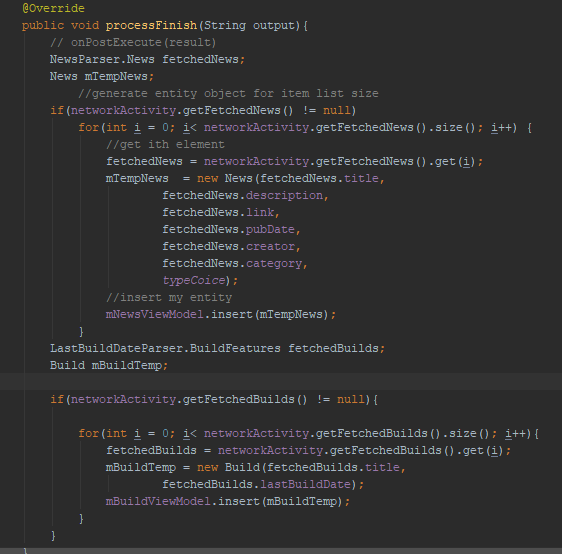
**4.3 Network activtiy den gelen bilgileri Room’a geçirmek:**

****

Network ile istediğim tipteki bilgileri alıyorum,

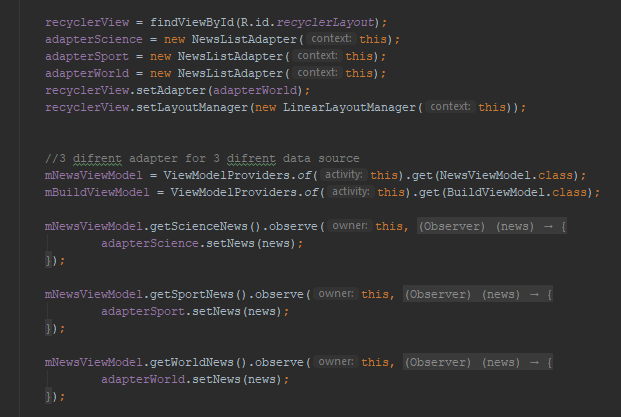


PostExeCute metodunu kullanmak için delegate nesnesi üretiyorum.



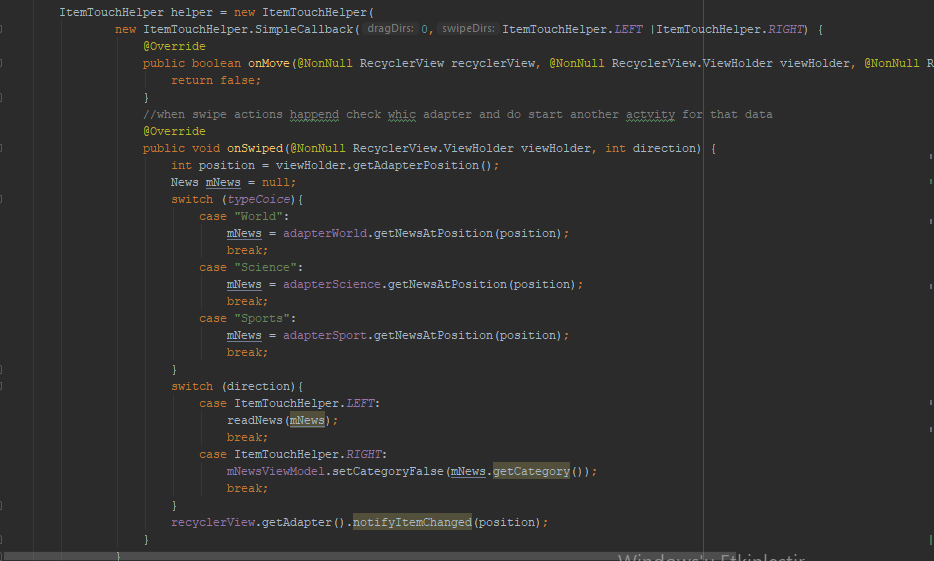
postExecute çalıştığında yani bilgilerim tamamen geldiğinde kontrollerden sonra Rooma kaydediyorum.

**4.4 Kullanıcın istediği kaynağı görüntülemek için LiveData ve Rcycler view bağlantısı**

****

3 farklı adapter oluşturup Live data yapılarına observer bağlarayarak olası bir data değişiminde rcycler view yapsını tekrar güncelliyorum (adapteri tekrar bağlıyorum).

**5 Sağa ve Sola kaydırma aksiyonları:**

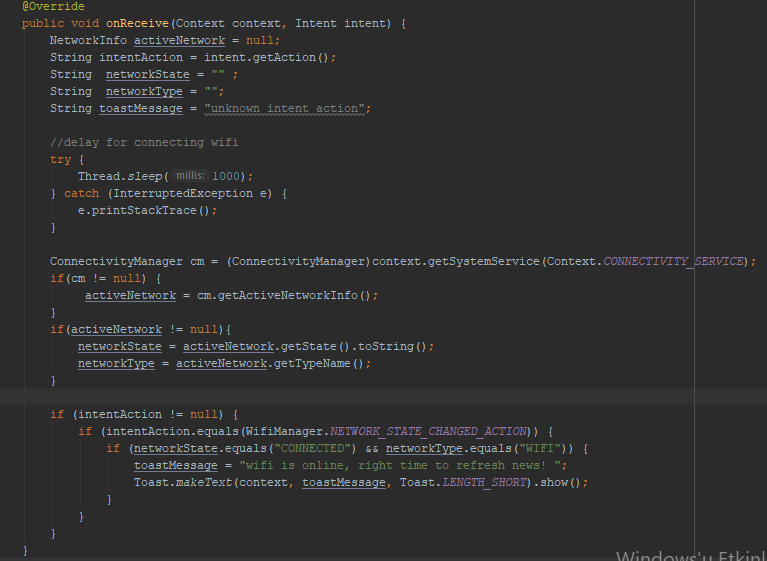
****

İtemTouchHelper yardımı ile sağa kaydırıldığında itemi silme (vaildCategorinin false yapılması ve live data tarafından göz ardı edilmesi), sola kaydırma aksiyonunda ise DescriptionActivity ile haberin ayrıntılı okunmasını sağlıyorum.

**6 BroadcastReciver ile wifi bağlantısını kontrol etmek**

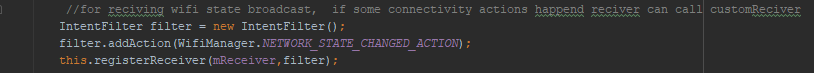
Wifi bağlı olduğu zaman kullanıcıya haberleri güncellemesini hatırlatan bir toast mesajı yayınlar.

CustomReciver classı: app tarafından kayıt edilen (filitreler) broadcast mesajları yakalandığında ne yapılacağını belirler

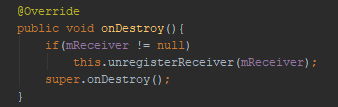


Broadcast mesajı geldiğinde connectivity menager ile aktif bağlantının bilgilerini activeNetwork’e alıyoruz, eğer bağlantı var ise bilgilerini tip ve state olarak alıyoruz, buradaki Thread.sleep kodu wifi bağlantılarının yapısından dolayı bağlantının geç kurulması durumundan kaynaklı bir bekleme süresi ekliyoruz.

Bağlantı ve tip bilgilerimiz uyuşuyor ise toast mesajı yayınlıyoruz.



MainActivity OnCreate metodunda reciveri kaydetmek



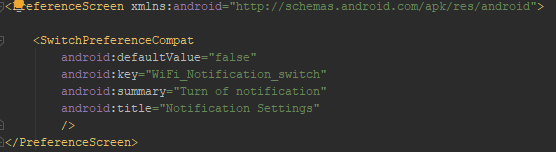
Gereksiz kaynak tüketmemek için onDestroy metodunda reciverin kaydını siliyoruz.

**7 SettingsActivity, fragment ve shared preferences:**

Buradaki amacımız fragment yapısı olarak settings activityi kodlamak, notification alıp almama seçeneklerini sharedpref ile kalıcı olarak saklamak.

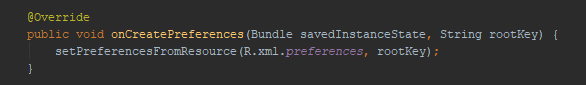
**7.1 Fragment oluşturmak**

Res>xml klasörü altında preferences.xml olarak bir fragment yaratıyoruz

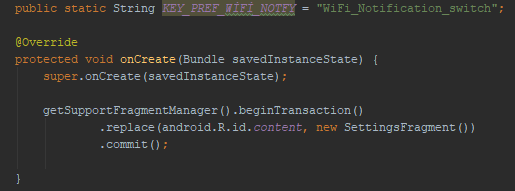


UI elemanı olan bir Switch oluşturuyoruz.

SettingsFragment classı içerisinde onCreatPreferences metodunda kullanılacak xml’i bağlıyoruz



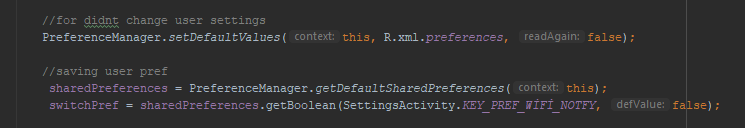
**7.2 SettingsActivity de fragmenti kullanmak**

****

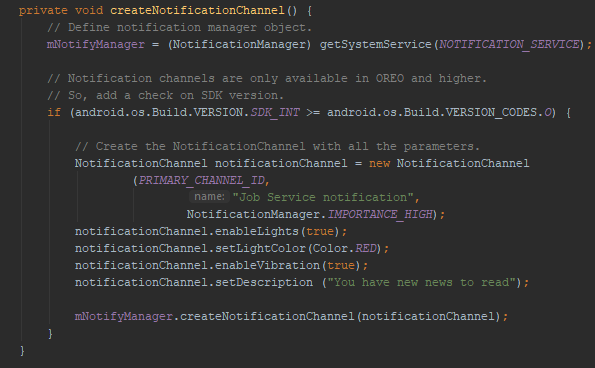
*KEY\_PREF\_WİFİ\_NOTFY*

İle main activity üzerinde seçimin bundle keyini belirmiş oluyoruz

**7.4 Main Activity de seçimin kaydedilmesi:**

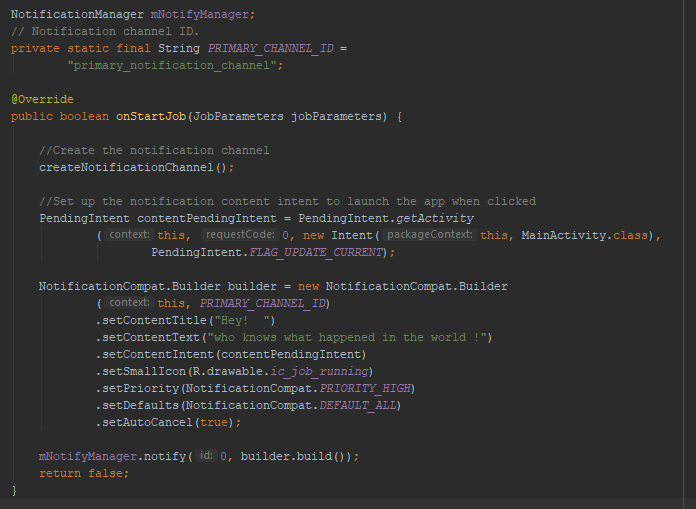
****

**8 JobScheuler ile notification gönderilmesi**

****

Notification channel yaratılması,

SistemService üstünden notyficationManagerin oluşturulması, android sürümü kontrolü sonrasında notification channel oluşturulması.

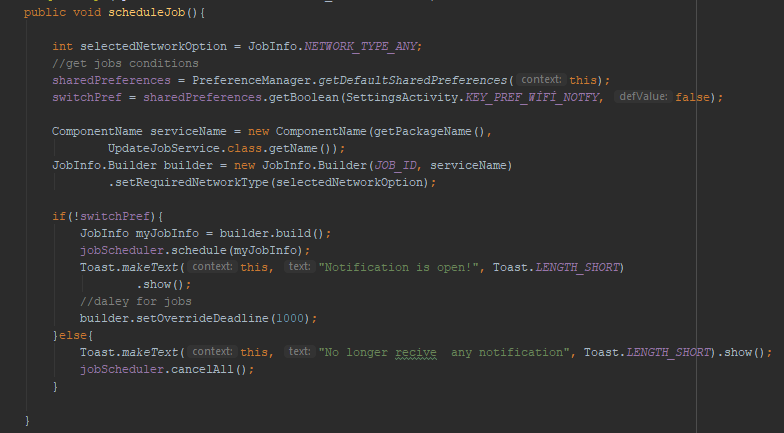


OnStartJob

Notificationchannel oluşturduktan sonra, pending intent oluşturuyoruz, bu sistemin bizi ayağa kaldırmasını sağlıyor.



onStopJob, eğer parametre olarak false gönderir ise jobscheularin görevi tamamlanmış olur ve sistem bir daha jobu tekrar kurmaz.



MainACtivity içerisinde jobu kurmak,

sharedPref üstünden switchPref alıyoruz, kullanıcı tarafından seçilmiş notificationların atılıp atılmadığını kotrol eden settings bilgisi,

Builder ile jobu kuruyoruz burada en az bir condition olması OS tarafında bize zorunlu tutuluyor, internete bağlı olduğumuzu kontrol ediyoruz.