Android App 개발 기록

# commit list

**commit 781e4cbb9acabb8ab269581d89857c19e894894b (HEAD -> master, origin/master)**

Author: wjlee <wjg2nius@gmail.com>

Date: Thu May 13 14:41:07 2021 +0900

[Study][Android] MyProject: Modify to crawl any content type

**commit a560420f39a17e212703b83b1af885868502d17d (HEAD -> master, origin/master)**

Author: wjlee <wjg2nius@gmail.com>

Date: Thu May 13 14:11:04 2021 +0900

[Study][Android] MyProject: Add https connection

**commit 0677910f6f46937bea2faf8fa80f27a673bcb6d6 (HEAD -> master, origin/master)**

Author: wjlee <wjg2nius@gmail.com>

Date: Wed May 12 17:59:43 2021 +0900

[Study][Android] MyProject: Fix issue - frame drop

Description:

In the first turn of the Recycler View of MainFragment,

the first crawling in onCreateView takes a long time to crawl,

resulting in frame drop.

This should be changed to crawling in advance at the time of app execution,

and this is resolved by declaring it as lazy init in MainActivity.

(Reference link: https://victorywskim.tistory.com/7)

**commit 379d88f2feab06bec52480cf398f37d73b5394cf**

Author: wjlee <wjg2nius@gmail.com>

Date: Tue May 4 15:42:14 2021 +0900

Test Init

# commit 설명

1. 379d88f2feab06bec52480cf398f37d73b5394cf

**초기 개발 상태**

Jsoup.connect("<https://dhlottery.co.kr/common.do?method=main>").get()

Jsoup.connect("<https://m.dhlottery.co.kr/common.do?method=main>").get()

viewpager 첫 전환 시 이렇게 직접 크롤링해서 가져오는 상태

정보는 잘 뽑아오고, 첫 화면 전환 시 버벅거림 발생함

1. 0677910f6f46937bea2faf8fa80f27a673bcb6d6 - 버벅거림 해결!

Kotlin 에서는 전역변수로 하면 초기화가 무조건 포함되어야 하고,

초기화를 포함시키면 App 처음 올라올 때 Network 연결이 들어가서 에러 띄운다.

(onCreate 관련해서 처음 앱을 띄울때 시간이 오래걸리는? 작업은 넣지 않도록 하는것이 안드로이드 정책임, 따라서 크롤링 넣으면 실행안됨.)

하지만 한번만 크롤링하기 위해서는 전역변수를 써야 여러곳에서 접근 가능할것.

lazy init 을 통해서 해결함

|  |
| --- |
| --- a/EmptyApp/app/src/main/java/com/example/emptyapp/MainActivity.kt  +++ b/EmptyApp/app/src/main/java/com/example/emptyapp/MainActivity.kt  @@ -15,6 +15,12 @@ import org.jsoup.Jsoup  import org.jsoup.nodes.Document  class MainActivity() : AppCompatActivity() {  + val crawling by lazy {  + Crawling(  + Jsoup.connect("https://dhlottery.co.kr/common.do?method=main").get(),  + Jsoup.connect("https://m.dhlottery.co.kr/common.do?method=main").get()  + )  + }  --- a/EmptyApp/app/src/main/java/com/example/emptyapp/MainFragment/MainFragment.kt  +++ b/EmptyApp/app/src/main/java/com/example/emptyapp/MainFragment/MainFragment.kt  + inner class crawlingThread: Thread() {  + override fun run() {  + Log.d("Time 0-1 ", System.currentTimeMillis().toString())  + (activity as MainActivity).crawling.doc  + (activity as MainActivity).crawling.doc\_mobile  + Log.d("Time 0-2 ", System.currentTimeMillis().toString())  + }  + }  +  } |

lazy init 으로 미리 크롤링 코드를 전역변수로 넣어서 초기화 구문은 넣고,

실제 init 되는 시점은 MainFragment 에서 crawlingThread.start() ~ crawlingThread.join() 시점에 init 되도록 의도적으로 접근하는 코드를 삽입.

그러면 이 시점에 접근하면서 init 이 되므로 이 시점에 crawling 된다.

이렇게 하면 FirstFragment, SecondFragment 의 onViewCreated 시점에 크롤링을 새로 할 필요가 없어서 버벅거림이 사라진다.

1. a560420f39a17e212703b83b1af885868502d17d - Add https connection

Oracle Cloud VM에 Spring 으로 만든 WebServer 올리고 여기에 접근하여 정보를 가져오도록 변경하고자 했음

1. http 로 접근하면 보안상 접근 안되도록 Android에서 막음

Server를 https 주소로 바꿔서 다시 구동함.

1. https 를 접근해도 문제 발생

문제: Trust anchor for certification path not found

인증서가 없어서 문제 발생.

기존에 https 구현할 때 생성했던 인증서를 추가하고(res/raw/certlwj.cer),

MainActivity 내부에

|  |
| --- |
| class MainActivity() : AppCompatActivity() {  **+ lateinit var cf: CertificateFactory**  **+ lateinit var caInput: InputStream**  **+ lateinit var ca: Certificate**  **+ lateinit var keyStoreType: String**  **+ lateinit var keyStore: KeyStore**  **+ lateinit var tmfAlgorithm: String**  **+ lateinit var tmf: TrustManagerFactory**  **+ lateinit var sslContext: SSLContext**  +  + val CustomServerURL = "https://152.70.248.64:8443/"  + val crawling\_custom\_server by lazy {  + CrawlingCustomServer(  + Jsoup.connect(CustomServerURL + "crawl/web")  + **.sslSocketFactory(sslContext.socketFactory)**.get(),  + Jsoup.connect(CustomServerURL + "crawl/mobile")  + **.sslSocketFactory(sslContext.socketFactory)**.get(),  + //Jsoup.connect(CustomServerURL + "update")  + // .sslSocketFactory(sslContext.socketFactory).get(),  + //Jsoup.connect(CustomServerURL + "lastWinning")  + // .sslSocketFactory(sslContext.socketFactory).get(),  + )  + }  …  }  + **httpsServerConnection()**  fragmentChange(FragmentIndex.MainFragment)  …  @@ -67,6 +102,29 @@ class MainActivity() : AppCompatActivity() {  }  }  + fun httpsServerConnection() {  **+ cf = CertificateFactory.getInstance("X.509")**  **+ caInput = applicationContext.getResources().openRawResource(R.raw.certlwj)**  **+ ca = cf.generateCertificate(caInput)**  **+ keyStoreType = KeyStore.getDefaultType()**  **+ keyStore = KeyStore.getInstance(keyStoreType)**  **+ tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm()**  **+ tmf = TrustManagerFactory.getInstance(tmfAlgorithm)**  **+ sslContext = SSLContext.getInstance("TLS")**  **+**  **+ keyStore.load(null, null)**  **+ keyStore.setCertificateEntry("ca", ca)**  **+ tmf.init(keyStore)**  **+ sslContext.init(null, tmf.getTrustManagers(), null)** |

위와 같이 추가함. 하지만 여전히 문제 발생

1. 안드로이드 클라이언트에서도 인증을 검증해야 함.

문제: SSLPeerUnverifiedException trustmanager

아래 함수는 default가 false 이고, 이를 true 로 override 하면 해결됨.

|  |
| --- |
| + fun httpsServerConnection() {  + cf = CertificateFactory.getInstance("X.509")  + caInput = applicationContext.getResources().openRawResource(R.raw.certlwj)  + ca = cf.generateCertificate(caInput)  + keyStoreType = KeyStore.getDefaultType()  + keyStore = KeyStore.getInstance(keyStoreType)  + tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm()  + tmf = TrustManagerFactory.getInstance(tmfAlgorithm)  + sslContext = SSLContext.getInstance("TLS")  +  + keyStore.load(null, null)  + keyStore.setCertificateEntry("ca", ca)  + tmf.init(keyStore)  + sslContext.init(null, tmf.getTrustManagers(), null)  +  **+ val hv: HostnameVerifier = object : HostnameVerifier {**  **+ override fun verify(arg0: String?, arg1: SSLSession?): Boolean {**  **+ return true**  **+ }**  **+ }**  **+ HttpsURLConnection.setDefaultHostnameVerifier(hv)**  + }  + |

굵은 글씨 부분을 추가해줌.

이렇게 하면 "crawl/web", "crawl/mobile" 에 대해서는 성공.

1. 781e4cbb9acabb8ab269581d89857c19e894894b

에러 정보:

unhandled content type. must be text/\*, application/xml, or application/\*+xml. mimetype=application/json

기본 crawling은 가능하지만, “update”, “lastWinning” 은 type이 Long, Winning 이라서 타입이 맞지 않아 발생하는 문제이다.

Jsoup 사용 시 ignoreContentType 을 사용하면 해결 가능하다.

|  |
| --- |
| - //Jsoup.connect(CustomServerURL + "update")  - // .sslSocketFactory(sslContext.socketFactory).get(),  - //Jsoup.connect(CustomServerURL + "lastWinning")  - // .sslSocketFactory(sslContext.socketFactory).get(),  + Jsoup.connect(CustomServerURL + "update")  + .sslSocketFactory(sslContext.socketFactory)  + .ignoreContentType(true).get(),  + Jsoup.connect(CustomServerURL + "lastWinning")  + .sslSocketFactory(sslContext.socketFactory)  + .ignoreContentType(true).get(), |