

WebData

WebView

- HTML 문서를 출력하는 클래스
 - 웹페이지를 출력하는 뷰

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- 주요 함수
 - **void loadUrl(String url)**
 - 주어진 url의 웹문서를 로딩함
 - 안드로이드 Pie 버전 이후 부터는 "https://"를 통합 접근만 허용됨
 - **void setWebViewClient(WebViewClient client)**
 - WebViewClient 객체와 연결하여, 안드로이드의 기본 웹브라우저 대신 Activity 내에서 WebView가 실행되게 설정함
 - WebViewClient 클래스 : Activity 내에서 WebView가 실행되게 함
 - **WebSettings getSettings()**
 - WebView 설정을 제어하기 위한 WebSettings 객체를 가져옴
 - WebSettings 클래스 : WebView 설정 상태를 관리함

WebView

- WebSettings
 - void setBuiltInZoomControls(boolean enabled)
 - WebView가 줌 기능이 가능하도록 함
 - Synchronized void setJavaScriptEnabled(boolean flag)
 - WebView가 자바 스크립트 실행이 가능하도록 함
- WebViewClient
 - WebViewClient() : WebViewClient 클래스의 생성자

```
AndroidView(  
    modifier = Modifier.fillMaxSize(),  
    factory = { context : Context ->  
        WebView(context).apply { this: WebView  
            webViewClient = WebViewClient()  
            settings.javaScriptEnabled = true  
            settings.builtInZoomControls = true  
            settings.defaultTextEncodingName="utf-8"  
        }  
    },  
    update = { webView : WebView ->  
        webView.loadUrl( url: "https://www.google.com")  
    }  
)
```

AndroidView

- 기존의 Android View를 compose 계층 구조에 통합하는 기능
 - Jetpack compose에서 제공하지 않는 뷰 사용

```
@Composable
fun <T : View> AndroidView(
    factory: (Context) -> T,
    modifier: Modifier = Modifier,
    update: (T) -> Unit = {}
)
```

- factory : Android view를 생성하는 람다 함수
- update : 뷰가 업데이트 될 때 호출되는 람다 함수

AndroidView

- WebView 생성

```
AndroidView(  
    modifier = Modifier.fillMaxSize(),  
    factory = { context : Context ->  
        WebView(context).apply { this: WebView  
            webViewClient = WebViewClient()  
            settings.javaScriptEnabled = true  
            settings.builtInZoomControls = true  
            settings.defaultTextEncodingName="utf-8"  
        }  
    },  
    update = { webView : WebView ->  
        webView.loadUrl( url: "https://www.google.com")  
    }  
)
```

예제. WebView

- WebView 사용 예 확인하기

실습. 다음 뉴스 가져오기

- <https://news.daum.net>의 뉴스 정보 가져오기

- 웹 페이지 소스 가져오기
- html 파싱

尹 "장바구니·외식 물가 잡는 데 정부 역량 총동원"

"조국·이준석도 만날 건가..한동훈과는 결별?" 尹 숨고르더니..[현장영상]

尹 "반도체 규제 풀어 도울 것...금투세 폐지 않으면 자금 이탈 우려"

전자·액상담배는 덜 해로울까..실제 실험해보니[르포]

[단독] 구본성, 회삿돈 부정 수급 혐의 전 임원 아워홈 사내이사 추천

尹 "특검? 지난 정부, 내 가족 봐줬다는 거냐" 불만...명품백은 사과

조선일보 도쿄특파원 "韓 외교부, 日 언론 플레이까지 돕나"

뉴스타파 진상조사위 "김만배 면담녹취는 신학림의 독자적 판단"

비만인은 '○○ 운동'이 최적 ...그럼 일반인은 언제?

의대 증원 핵심 '국립대'의 반란...정부, 당혹감 못 감춰

'업계 1위'골프존, 관리 소홀로 221만명 정보 유출...과징금 75억

Coroutine

- 비동기적으로 실행되는 코드를 간소화하기 위해 사용
 - 여러 서브루틴을 번갈아 실행하는 비동기적인 프로그래밍 기법
 - 멀티 Thread를 운영할 때 발생하는 Context Switching의 부담을 줄임
 - 하나의 Thread로 여러 coroutine을 운영할 수 있음
- CoroutineScope로 코루틴의 범위 관리
 - 디스패처(Dispatchers)를 사용하여 코루틴 실행에 사용되는 스레드 확인
 - Dispatchers.Main
 - 기본(UI, Main) 스레드에서 코루틴을 실행
 - Dispatchers.IO
 - 기본 스레드 외부에서 디스크 또는 네트워크 I/O 실행에 최적화
 - Dispatchers.Default
 - CPU를 많이 사용하는 작업을 기본 스레드 외부에서 실행

Coroutine

- Coroutine 실행

- launch : 메인 스레드를 블록 상태로 만들지 않음, Job 반환
- async : 메인 스레드를 블록 상태로 만듦, Deferred 반환

```
val scope = CoroutineScope(Dispatchers.Main)
```

```
scope.launch {
```

```
}
```

→ withContext(Dispatchers.Main)

```
scope.async {
```

```
}.await()
```

→ withContext(Dispatchers.IO)

- withContext 함수

- CoroutineContext를 바꿀때 사용
- Suspend 함수로 선언되어 있어, CoroutineScope에서 실행

웹페이지 소스 가져오기

- 네트워크 연결

```
fun loadNetwork(url: URL): String {  
    return try {  
        val connect = (url.openConnection() as HttpURLConnection).apply {  
            connectTimeout = 4000  
            readTimeout = 1000  
            requestMethod = "GET"  
        }  
        if (connect.responseCode == HttpURLConnection.HTTP_OK) {  
            streamToString(connect.inputStream)  
        } else {  
            ""  
        }  
    } catch (e: Exception) {  
        Log.e("error", "네트워크 요청 실패", e)  
        ""  
    }  
}
```

웹페이지 소스 가져오기

- 소스 읽어오기

```
fun streamToString(inputStream: InputStream): String {  
    return try {  
        BufferedReader(InputStreamReader(inputStream)).useLines { lines ->  
            lines.fold("") { accumulator, line ->  
                accumulator + line  
            }  
        }  
    } catch (e: Exception) {  
        Log.e("error", "읽기 실패")  
        ""  
    }  
}
```

실습. 다음 뉴스 가져오기

- <https://news.daum.net>의 뉴스 정보 가져오기

- 웹 페이지 소스 가져오기
- html 파싱

<https://news.daum.net>

fetch

```
<!DOCTYPE html><html lang="ko"
class="os_android none tablet version_0
"><head><meta property="og:site_name"
content="다음뉴스"><meta
property="og:site_name" content="다음뉴스" /
><meta property="og:title" content="홈" /
><meta property="og:image"
content="https://t1.daumcdn.net/media/img
-media/mobile/meta/news.png" /><meta
property="og:description" content="다음뉴스
모바일" /><meta property="og:url"
content="https://news.daum.net/" /><meta
name="referrer" content="always"><link
rel="shortcut icon"
href="https://m2.daumcdn.net/img-media
/2010ci/Daum_favicon.ico"><title>홈 | 다음뉴
스</title><meta http-equiv="X-UA-Compatible"
content="IE=edge"><meta name="viewport"
content="user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0,
width=device-width,
target-densityDpi=medium-dpi.
```

HTML 파싱 : Jsoup

- HTML을 파싱하기 위한 자바 라이브러리
 - <https://jsoup.org/>
 - URL, 파일, 문자열을 소스로 하여 HTML을 파싱할 수 있음
 - DOM 구조 탐색 및 CSS 선택자를 사용하여 데이터를 추출할 수 있음
 - 문서내의 HTML 요소, 속성, 텍스트를 조작할 수 있음
 - 사용자가 입력한 데이터로 부터 XSS 공격을 방지하기 위하여 안전한 화이트리스트 방식으로 지정된 태그만 남기고 나머지는 제거 가능
 - 깔끔한 형태의 html을 출력할 수 있음
 - Build.gradle(app)에 라이브러리 추가

'org.jsoup:jsoup:1.17.2'

Jsoup

- Document 클래스
 - URL로 부터 받은 HTML 문서

```
val doc: Document = Jsoup.connect(url: "https://en.wikipedia.org/").get()
val result: String! = doc.text()
```

Wikipedia, the free encyclopedia Main Page From Wikipedia, the free encyclopedia Jump to navigation Jump to search Welcome to Wikipedia, the free encyclopedia that anyone can edit. 6,074,222 articles in English Arts Biography Geography History Mathematics Science Society Technology All portals From today's featured article USS Oberrender (DE-344) was a John C. Butler-class destroyer escort built for the United States Navy during World War II. She was named for Lieutenant Commander Thomas Olin Oberrender, Jr., the engineering officer of the light cruiser USS Juneau, killed when that ship was torpedoed and sunk during the Naval Battle of Guadalcanal. Laid down in November 1943, launched in January 1944, and commissioned almost four months later, Oberrender commenced convoy escort duty in the Pacific in late 1944, with an interlude protecting escort carriers during the early stages of the invasion of Leyte. She was heavily damaged by the explosion of the ammunition ship USS Mount Hood at Manus and was repaired there during November. Returning to service in December, Oberrender served on anti-submarine patrol during the Battle of Okinawa, during which she was irreparably damaged by a kamikaze attack in earl May 1945. She was decommissioned and sunk as a target late that year. (Full article...) Recently featured: 1974 White House helicopter incident Macedonia (ancient kingdom) Unknown (magazine) Archive By email More featured articles Did you know ... Murals in Bjäresjö Church ... that some of the early church murals in Sweden (example pictured) may have been commissioned by Danish kings? ... that Charlton Miner Lewis was a lawyer before becoming an English professor at Yale, and the first judge of the Yale Series of Younger Poets? ... that the scale insect Saissetia coffeae attacks many other plants besides coffee? ... that 70 Pine Street used double-deck elevators due to its narrow plot? ... that in 2001, sixteen-year-old Alina Lebedeva struck Prince Charles of the United Kingdom in the face with three red carnations? ... that after the Battle of Prairie Grove,

<https://jsoup.org/cookbook/input/load-document-from-url>

Jsoup

- Elements
 - Element의 모임
- Element
 - Document의 HTML 요소
- **Select 함수**
 - CSS의 selector syntax를 사용한 Elements 찾는 함수
 - tagname : a
 - .classname : .masthead
 - [attribute] : [href]
 - #id : #logo
 - el.class : div.masthead
 - el[attr] : a[href]
 - ancestor child : .body p (body 클래스를 가진 블록 내에 p)
 - Parent > child : div.content > p (div.content의 child p)

<https://jsoup.org/cookbook/extracting-data/selector-syntax>

Jsoup

- Elements 예

```
val newsHeadlines: Elements = doc.select(cssQuery: "#mp-itn b a")
for (headline: Element! in newsHeadlines) {
    Log.i(tag: "jsoup : ", msg: headline.attr(attributeKey: "title")+"\n"+headline.absUrl(attributeKey: "href"))
}
```

```
<div id="mp-itn" style="padding:0.1em 0.6em;"><div style="margin: 3px 1px 4px 1px; padding: 5px; text-align:center; border-style: solid; background:#fafcfe; border:1px solid #a3b0bf;"><b><a href="/wiki/COVID-19_pandemic"
title="COVID-19 pandemic">COVID-19 pandemic</a></b><br /><div class="hlist" style="font-size: 85%;">
<ul><li><a href="/wiki/Coronavirus_disease_2019" title="Coronavirus disease 2019">Disease</a></li>
<li><a href="/wiki/Severe_acute_respiratory_syndrome_coronavirus_2" title="Severe acute respiratory syndrome coronavirus 2">Virus</a></li>
<li><a href="/wiki/COVID-19_testing" title="COVID-19 testing">Testing</a></li>
<li><a href="/wiki/Timeline_of_the_COVID-19_pandemic" title="Timeline of the COVID-19 pandemic">Timeline</a>
<ul><li><a href="/wiki/Timeline_of_the_COVID-19_pandemic_in_May_2020" title="Timeline of the COVID-19 pandemic in May 2020">May</a></li></ul></li>
<li><a href="/wiki/COVID-19_pandemic_by_country_and_territory" title="COVID-19 pandemic by country and territory">By location</a></li>
<li><a href="/wiki/Socio-economic_impact_of_the_COVID-19_pandemic" title="Socio-economic impact of the COVID-19 pandemic">Impact</a></li>
<li><a href="/wiki/List_of_deaths_due_to_coronavirus_disease_2019" title="List of deaths due to coronavirus disease 2019">Deaths</a></li>
<li><a href="/wiki/Portal:Coronavirus_disease_2019" title="Portal:Coronavirus disease 2019">Portal</a></li></ul>
</div></div>
```

```
COVID-19 pandemic
https://en.wikipedia.org/wiki/COVID-19_pandemic
Visakhapatnam gas leak
https://en.wikipedia.org/wiki/Visakhapatnam_gas_leak
HR 6819
https://en.wikipedia.org/wiki/HR_6819
ABS-CBN franchise renewal controversy
https://en.wikipedia.org/wiki/ABS-CBN_franchise_renewal
_controversy
Deaths in 2020
https://en.wikipedia.org/wiki/Deaths_in_2020
Portal:Current events
https://en.wikipedia.org/wiki/Portal:Current_events
Wikipedia:In the news/Candidates
https://en.wikipedia.org/wiki/Wikipedia:In_the_news/Candidates
```

<https://jsoup.org/cookbook/extracting-data/selector-syntax>

Jsoup

- Elements 예

```
val newsHeadlines: Elements = doc.select(cssQuery: "#mp-itn b a")
for (headline: Element! in newsHeadlines) {
    Log.i(tag: "jsoup : ", msg: headline.attr(attributeKey: "title")+"\n"+headline.absUrl(attributeKey: "href"))
}
```

```
<div id="mp-itn" style="padding:0.1em 0.6em;"><div style="margin: 3px 1px 4px 1px; padding: 5px; text-align:center; border-style: solid; background:#fafcfe; border:1px solid #a3b0bf;"><b><a href="/wiki/COVID-19_pandemic"
title="COVID-19 pandemic">COVID-19 pandemic</a></b><br /><div class="hlist" style="font-size: 85%;">
<ul><li><a href="/wiki/Coronavirus_disease_2019" title="Coronavirus disease 2019">Disease</a></li>
<li><a href="/wiki/Severe_acute_respiratory_syndrome_coronavirus_2" title="Severe acute respiratory syndrome coronavirus 2">Virus</a></li>
<li><a href="/wiki/COVID-19_testing" title="COVID-19 testing">Testing</a></li>
<li><a href="/wiki/Timeline_of_the_COVID-19_pandemic" title="Timeline of the COVID-19 pandemic">Timeline</a>
<ul><li><a href="/wiki/Timeline_of_the_COVID-19_pandemic_in_May_2020" title="Timeline of the COVID-19 pandemic in May 2020">May</a></li></ul></li>
<li><a href="/wiki/COVID-19_pandemic_by_country_and_territory" title="COVID-19 pandemic by country and territory">By location</a></li>
<li><a href="/wiki/Socio-economic_impact_of_the_COVID-19_pandemic" title="Socio-economic impact of the COVID-19 pandemic">Impact</a></li>
<li><a href="/wiki/List_of_deaths_due_to_coronavirus_disease_2019" title="List of deaths due to coronavirus disease 2019">Deaths</a></li>
<li><a href="/wiki/Portal:Coronavirus_disease_2019" title="Portal:Coronavirus disease 2019">Portal</a></li></ul>
</div></div>
```

```
COVID-19 pandemic
https://en.wikipedia.org/wiki/COVID-19_pandemic
Visakhapatnam gas leak
https://en.wikipedia.org/wiki/Visakhapatnam_gas_leak
HR 6819
https://en.wikipedia.org/wiki/HR_6819
ABS-CBN franchise renewal controversy
https://en.wikipedia.org/wiki/ABS-CBN_franchise_renewal
_controversy
Deaths in 2020
https://en.wikipedia.org/wiki/Deaths_in_2020
Portal:Current events
https://en.wikipedia.org/wiki/Portal:Current_events
Wikipedia:In the news/Candidates
https://en.wikipedia.org/wiki/Wikipedia:In_the_news/Candidates
```

<https://jsoup.org/cookbook/extracting-data/selector-syntax>

예제. 다음 뉴스 가져오기

- <https://news.daum.net>의 뉴스 정보 가져오기

- 웹 페이지 소스 가져오기
- **html 파싱**

尹 "장바구니·외식 물가 잡는 데 정부 역량 총동원"

"조국·이준석도 만날 건가..한동훈과는 결별?" 尹 숨고르더니..[현장영상]

尹 "반도체 규제 풀어 도울 것...금투세 폐지 않으면 자금 이탈 우려"

전자·액상담배는 덜 해로울까..실제 실험해 보니[르포]

[단독] 구본성, 회삿돈 부정 수급 혐의 전 임원 아워홈 사내이사 추천

尹 "특검? 지난 정부, 내 가족 봐줬다는 거냐" 불만...명품백은 사과

조선일보 도쿄특파원 "韓 외교부, 日 언론 플레이까지 뚝나"

뉴스타파 진상조사위 "김만배 면담녹취는 신학림의 독자적 판단"

비만인은 '○○ 운동'이 최적 ...그럼 일반인은 언제?

의대 증원 핵심 '국립대'의 반란...정부, 당혹감 못 감춰

'업계 1위'골프존, 관리 소홀로 221만명 정보 유출...과징금 75억

PullRefreshIndicator

- 주로 사용자가 리스트를 아래로 당겨서 새로 고침
 - 새로운 데이터를 로드하거나 최신 데이터를 가져오는데 사용
 - 라이브러리 : [androidx.compose.material:material](https://developer.android.com/reference/kotlin/androidx/compose/material/package-summary) 추가

```
var isLoading by remember { mutableStateOf(false) }
```

```
val pullRefreshState = rememberPullRefreshState(  
    refreshing = isLoading,  
    onRefresh = { } // refresh할 때 수행할 일  
)
```

```
PullRefreshIndicator(  
    refreshing = isLoading,  
    state = pullRefreshState,  
    modifier = Modifier.align(Alignment.TopCenter)  
)
```

* Refresh 기능 modifier에 연결

```
Modifier.pullRefresh(pullRefreshState)
```

예제. 다음 뉴스 가져오기

- <https://news.daum.net>의 뉴스 정보 가져오기

- 웹 페이지 소스 가져오기

- html 파싱

- PullRefreshIndicator 추가하기

尹 "장바구니·외식 물가 잡는 데 정부 역량 총동원"

"조국·이준석도 만날 건가..한동훈과는 결 별?" 尹 숨고르더니..[현장영상]

尹 "반도체 규제 풀어 도울 것...금투세 폐지 않으면 자금 이탈 우려"

전자·액상담배는 덜 해로울까..실제 실험해보니[르포]

[단독] 구본성, 회삿돈 부정 수급 혐의 전 임원 아워홈 사내이사 추천

尹 "특검? 지난 정부, 내 가족 봐줬다는 거냐" 불만...명품백은 사과

조선일보 도쿄특파원 "韓 외교부, 日 언론 플레이까지 돕나"

뉴스타파 진상조사위 "김만배 면담녹취는 신학림의 독자적 판단"

비만인은 '○○ 운동'이 최적 ...그럼 일반인은 언제?

의대 증원 핵심 '국립대'의 반란...정부, 당혹감 못 감춰

'업계 1위'골프존, 관리 소홀로 221만명 정보 유출...과징금 75억

실습. html 파싱

- 사이트 예
 - LazyColumn을 이용하여 노래와 가수 정보 가져오기
- <https://www.melon.com/chart/index.htm>**

XML 데이터 파싱

XML

- XML(eXtensible Markup Language)
 - 데이터 교환에 대한 표준
 - 데이터 구조를 사람이 읽을 수 있는 텍스트 문서로 쉽게 인코딩해 정확하게 해석할 수 있음

```
<?xml version="1.0" encoding="utf-8"?>
<customers>
  <customer>
    <name>홍길동</name>
    <address>서울 강남구</address>
  </customer>
  <customer>
    <name>고길동</name>
    <address>서울 강북구</address>
  </customer>
  <customer>
    <name>김길동</name>
    <address>서울 서초구</address>
  </customer>
</customers>
```

```
val doc : Document! = Jsoup.connect(data)
    .parser(Parser.xmlParser()).get()
val customers: Elements = doc.select(cssQuery: "customer")
for(customer: Element! in customers){
  Log.i( tag: "xml", msg: customer.select(cssQuery: "name").text()+
    customer.select(cssQuery: "address").text())
}
```

RSS 서비스

- **RSS (Really Simple Syndication)**

- Rich Site Summary라고도 함
- 뉴스나 블로그와 같이 콘텐츠 업데이트가 자주 일어나는 웹사이트에서, 업데이트된 정보를 자동으로 쉽게 사용자에게 제공하기 위한 서비스
- XML 데이터 형식

- 사이트 예

<https://fs.jtbc.co.kr/RSS/culture.xml>

실습. JTBC 뉴스 가져오기

- <https://fs.jtbc.co.kr/RSS/culture.xml>

JTBC 다큐멘터리 '딥 크리미널' 휴스턴국
제영화제 대상 수상

"감정 없네?" "점프해 봐!" 멧 갈라서 한국
가수 인종차별 논란 [소셜픽]

'무빙'과 '서울의 봄' 백상대상 쾌거...60년
역사 빛났던 말말말

7년 전 'BTS 사재기 의혹' 재조명...조사 움
직임 전해지자

"한국, BTS에 사과하라"...음원 사재기 조
사? 아미들 '발끈'

[인터뷰] '종이접기' 김영만 "코딱지 친구들
과 계속 함께하고파"

"루머 방치 말라"...뿔난 아미들, 하이브 사
옥 앞에 '근조화환'

'뉴진스 계약 해지권' 폭로전...민희진 배임
인정 땀 1000억 → 28억으로

가수들 라이브 들으며 "꿀잠 주무세요"...12
시간 '수면' 콘서트 [소셜픽]

일본 거리 버려진 아이돌 앨범...민희진의
'일침' 다시 봤더니 [소셜픽]

단월드 "절대 종교 아니다, BTS와 무관"...
공식입장 예고 [소셜픽]

JSON 데이터 파싱

JSON

- 데이터 교환 포맷 형식
- XML 보다 더욱 단순한 객체 표기법

```
{  
  "customers" : [  
    {  
      "name": "홍길동",  
      "address": "서울 강남구"  
    },  
    {  
      "name": "고길동",  
      "address": "서울 강북구"  
    },  
    {  
      "name": "김길동",  
      "address": "서울 서초구"  
    }  
  ]  
}
```

```
{ } ← JSONObject  
[ ] ← JSONArray  
"name ", "address" ← Name(key)  
"홍길동", "서울 강남구" ← Value
```

JSON Parser

클래스	설명
JSONArray	색인화된 일련의 값들, 값들은 JSONObject, JSONArray, String, Booleans, Integers, Longs, Doubles, null 등의 값들로 구성됨
JSONObject	Name/value 매핑으로 변경 가능한 집합. Name은 유일해야 하며, 값들은 JSONObject, JSONArray, String, Booleans, Integers, Longs, Doubles, null 등의 값들로 구성됨

클래스	메소드	설명
JSONObject	JSONArray getJSONArray(String name)	JSONArray가 있으면 name에 해당하는 값을 반환함
	String getString(String name)	name에 해당하는 값을 반환함
JSONArray	JSONObject getJSONObject(int index)	JSONObject가 있으면 index에 있는 값을 반환함

JSON Parser

```
{
  "customers" : [
    {
      "name": "홍길동",
      "address": "서울 강남구"
    },
    {
      "name": "고길동",
      "address": "서울 강북구"
    },
    {
      "name": "김길동",
      "address": "서울 서초구"
    }
  ]
}
```

```
val json = JSONObject(data);
```

```
val array = json.getJSONArray("customers");
```

```
val name = array.getJSONObject(0).getString("name");
val addr = array.getJSONObject(0).getString("address");
```

```
val name = array.getJSONObject(1).getString("name");
val addr = array.getJSONObject(1).getString("address");
```

```
val name = array.getJSONObject(2).getString("name");
val addr = array.getJSONObject(2).getString("address");
```

JSON Parser

```
fun parsingJSON(result: String) {  
    try {  
        val json = JSONObject(result)  
        val array = json.getJSONArray("customers")  
        for (i in 0 until array.length()) {  
            val name = array.getJSONObject(i).getString("name")  
            val addr = array.getJSONObject(i).getString("address")  
            textView.append(name + "Wt" + addr + "Wn")  
        }  
    } catch (e: JSONException) {  
        e.printStackTrace()  
    }  
}
```

예제. ChatGpt

- ChatGpt는 Json 형태로 질문하고, Json 형태로 응답

what is gps?

질문

<<Response>>

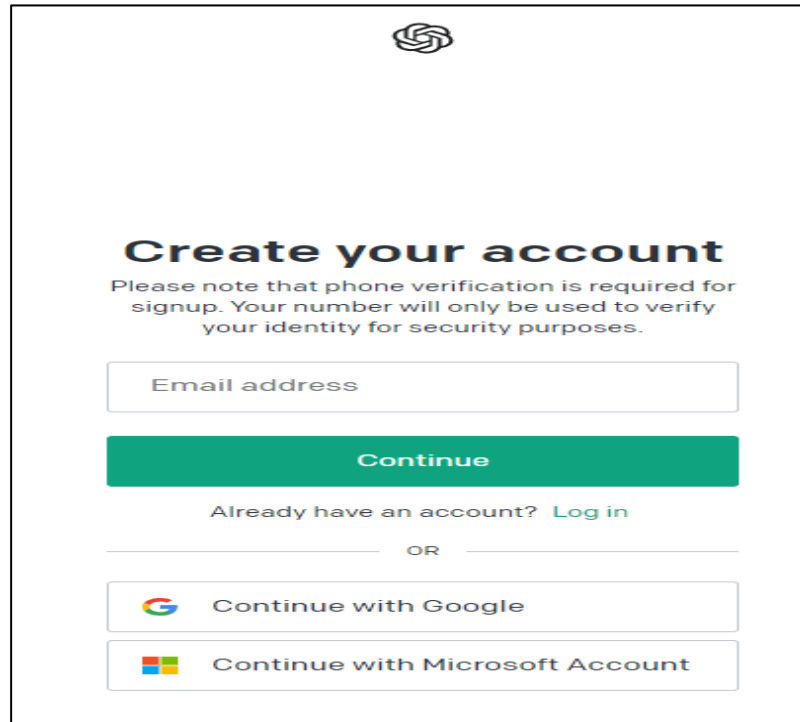
GPS (Global Positioning System) is a satellite-based navigation system that provides location and time information anywhere on Earth. It uses a network of satellites orbiting the Earth to transmit signals to GPS receivers, which can then determine their exact location and time.

GPS is commonly used for navigation, mapping, and tracking purposes in various industries, including transportation, aviation, and military.

ChatGpt 연동하기

- Platform.openai.com 가입하기
 - 이메일 인증과 휴대폰 인증 요구

<https://platform.openai.com/>



The screenshot shows the OpenAI account creation interface. At the top is the OpenAI logo. Below it, the heading "Create your account" is displayed in bold. A note states: "Please note that phone verification is required for signup. Your number will only be used to verify your identity for security purposes." There is a text input field labeled "Email address". Below the field is a green "Continue" button. Underneath the button, it says "Already have an account? Log in". A horizontal line with "OR" in the center separates this from the social login options. There are two buttons: "Continue with Google" (with the Google logo) and "Continue with Microsoft Account" (with the Microsoft logo).

가격 정책

- <https://openai.com/api/pricing/>

Multiple models, each with different capabilities and price points. Prices can be viewed in units of either per 1M or 1K tokens. You can think of tokens as pieces of words, where 1,000 tokens is about 750 words. This paragraph is 35 tokens.

GPT-4 Turbo

With 128k context, fresher knowledge and the broadest set of capabilities, GPT-4 Turbo is more powerful than GPT-4 and offered at a lower price.

[Learn about GPT-4 Turbo ↗](#)

Model	Input	Output
gpt-4-turbo-2024-04-09	US\$10.00 / 1M tokens	US\$30.00 / 1M tokens

Vision pricing calculator

Set width

Set height

150

px

by

150

px

=

US\$0.00255

[i](#)

☐ Low resolution

GPT-3.5 Turbo

GPT-3.5 Turbo models are capable and cost-effective.

`gpt-3.5-turbo-0125` is the flagship model of this family, supports a 16K context window and is optimized for dialog.

`gpt-3.5-turbo-instruct` is an Instruct model and only supports a 4K context window.

[Learn about GPT-3.5 Turbo ↗](#)

Model	Input	Output
gpt-3.5-turbo-0125	US\$0.50 / 1M tokens	US\$1.50 / 1M tokens
gpt-3.5-turbo-instruct	US\$1.50 / 1M tokens	US\$2.00 / 1M tokens

Text Generation

- 사용자의 프롬프트(prompt)에 대한 예상 Completions을 반환

<https://api.openai.com/v1/completions>

- Request

model	모델 (예, gpt-3.5-turbo-instruct)
prompt	프롬프트 (사용자 질문)
temperature	범위 : 0~2 높을수록 "창의성 있는 텍스트 생성"-무작위성
max_tokens	완료시 생성할 최대 토큰 수

- Response

Id / object / model	api 호출에 대한 property
choices (Json 배열타입)	text index logprobs finish_reason

Retrofit2

- 서버와 클라이언트의 HTTP 통신을 위한 네트워크 라이브러리
 - Squareup사의 REST API 통신 라이브러리
 - 어노테이션 사용으로 코드의 가독성이 뛰어남
 - 동기/비동기 작업을 쉽게 구현할 수 있음
 - 라이브러리 추가
 - com.squareup.retrofit2:retrofit
 - com.squareup.retrofit2:converter-gson
 - Json 객체로 변환해 주는 converter
 - 권한 요청

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Retrofit2

- 구성요소

- **DTO (Data Transfer Object)**: 응답 데이터 구조에 맞는 데이터 모델
 - @SerializedName 이 명시될 경우 속성명이 달라도 되지만, 없을 경우 반드시 일치해야 함

```
{
  "id": "cmpl-9Mx",
  "object": "text_completion",
  "created": 1715252571,
  "model": "gpt-3.5-turbo-instruct",
  "choices": [
    {
      "text": "WnWnGPS (Global Positioning System) is a satellite-based navigation system that provides location and time information anywhere on Earth. It uses a network of satellites orbiting the Earth to transmit signals to GPS receivers, which can then determine their exact location and time. GPS is commonly used for navigation, mapping, and tracking purposes in various industries, including transportation, aviation, and military.",
      "index": 0,
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 4,
    "completion_tokens": 75,
    "total_tokens": 79
  }
}
```

```
data class ChoicesResponse(
    @SerializedName("choices")
    val choiceList: List<Choice>
)

data class Choice(
    val text: String,
    val index: Int,
    val logprobs: Any?,
    val finish_reason: String
)
```

Retrofit2

- **Interface 정의**

- 사용할 메소드 선언

`https://api.openai.com/v1/completions`

baseurl

end point

```
interface ChatGptAPIService {
```

```
    @POST("completions")
```

http method : GET/POST/PUT/...

```
    suspend fun getCompletion(
```

```
        @Body request: CompletionRequest
```

@Body : 요청 객체 → Json 변환

```
    ): Response<ChoicesResponse>
```

```
}
```

```
data class CompletionRequest(
```

```
    val model: String,
```

```
    val prompt: String,
```

```
    val temperature: Int,
```

```
    val max_tokens: Int
```

```
)
```

Retrofit2

- Retrofit 인스턴스 생성

```
object RetrofitClient {  
    private const val BASE_URL = "https://api.openai.com/v1/"  
    val apikey = "Your api key"  
  
    private val client = OkHttpClient.Builder()  
        .addInterceptor { chain ->  
            val request = chain.request().newBuilder()  
                .header("Authorization", "Bearer $apikey")  
                .build()  
            chain.proceed(request)  
        }  
        .build()  
  
    private val retrofit: Retrofit = Retrofit.Builder()  
        .baseUrl(BASE_URL)  
        .client(client)  
        .addConverterFactory(GsonConverterFactory.create())  
        .build()  
  
    val apiService: ChatGptAPIService = retrofit.create(ChatGptAPIService::class.java)  
}
```

Retrofit2

- API 서비스 이용하기

```
class ChatGptRepository {  
    suspend fun getResponse(query: String): String {  
        val completionRequest = CompletionRequest(  
            model = "gpt-3.5-turbo-instruct",  
            prompt = query,  
            temperature = 0,  
            max_tokens = 200  
        )  
  
        val response = RetrofitClient.apiService.getCompletion(request = completionRequest)  
  
        if (response.isSuccessful) {  
            return response.body()?.choiceList?.get(0)?.text ?: ""  
        } else {  
            return ""  
        }  
    }  
}
```

```
class ChatGptViewModel(  
    private val repository: ChatGptRepository) : ViewModel() {
```

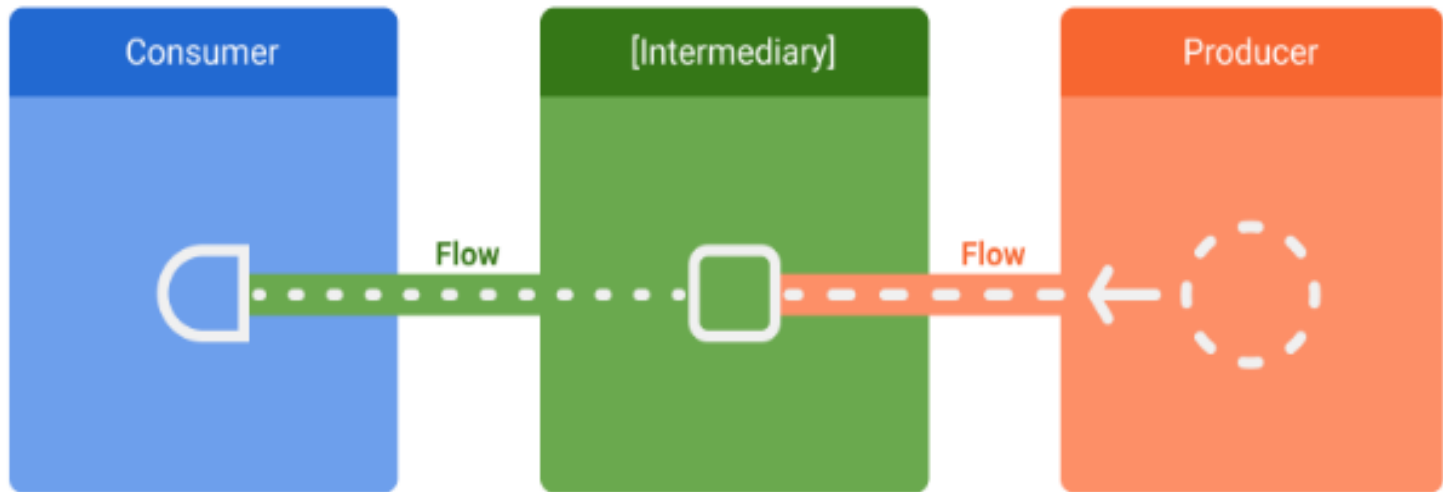
```
data class ChoicesResponse(  
    @SerializedName("choices")  
    val choiceList: List<Choice>  
)  
  
data class Choice(  
    val text: String,  
    val index: Int,  
    val logprobs: Any?,  
    val finish_reason: String  
)
```


Kotlin Flow

- Kotlin에서 비동기적인 데이터 스트림을 처리하기 위한 라이브러리
 - developer.android.com/kotlin/flow
 - Coroutines과 함께 사용되며, 비동기 코드를 작성하고 실행하기 위한 효과적인 방법을 제공
 - 기존의 suspend function이 단일 값만 반환하는 것과 달리 여러 값을 순차적으로 내보내는 기능이 가능
 - 네트워크 연결을 통해 실시간 도착하는 데이터 스트림 수신 가능

Kotlin Flow

- 구성 항목
 - 생산자 : 스트림에 추가되는 데이터 생산
 - (옵션) 중개자 : 스트림에 내보내는 각각의 값이나 스트림 자체를 수정
 - 소비자 : 스트림 값을 사용



Kotlin Flow

```
class DemoViewModel : ViewModel(){
    val myFlow : Flow<Int> = flow{ this: FlowCollector<Int>
        for(i : Int in 1..10){
            emit(i)
            delay( timeMillis: 2000)
        }
    }
}
```

```
@Composable
fun ScreenSetup(viewmodel: DemoViewModel = viewModel()) {
    MainScreen(viewmodel.myFlow)
}

@Composable
fun MainScreen(myFlow: Flow<Int>) {
    val count : Int by myFlow.collectAsState(initial = 0)
    Column(modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally) { this: ColumnScope
        Text(text="$count", style = TextStyle(fontSize = 40.sp))
    }
}
```

Kotlin Flow

```
class DemoViewModel : ViewModel(){  
    val myFlow : Flow<Int> = flow{ this: FlowCollector<Int>  
        for(i : Int in 1..10){  
            emit(i)  
            delay( timeMillis: 2000)  
        }  
    }  
  
    val newFlow : Flow<String> = myFlow  
        .filter { it % 2 == 0 }  
        .map { it: Int  
            "current value = $it"  
        }  
}
```

```
@Composable  
fun ScreenSetup(viewmodel: DemoViewModel = viewModel()) {  
    MainScreen(viewmodel.newFlow)  
}
```

Cold Flow / Hot Flow

- Cold Flow
 - 소비자가 값을 수집하기 시작해야만 생산자가 값을 방출
 - 예) Flow
- Hot Flow
 - 소비자가 값을 수집하는지에 관계없이 생산자는 값을 방출
 - 소비자는 가장 최근 값부터 순차적으로 값을 받게 됨
 - 예) StateFlow, SharedFlow

StateFlow

- 시간에 따라 변경되는 단일 값을 저장하고, 변경이 발생했음을 모든 소비자에게 알림
- 상태 변경에 반응하는 코드 작성 가능
 - 상태값이 변경되었는지 지속적으로 확인하는 코드 필요 없음
 - 생성

```
private val _response = MutableStateFlow<String>("")  
val response = _response.asStateFlow()
```
 - 상태 변경 `_response.value = result`
 - StateFlow 수집

```
val responseState by newsViewModel.response.collectAsState()
```

StateFlow

```
class DemoViewModel : ViewModel(){
    private val _stateFlow : MutableStateFlow<Int> = MutableStateFlow( value: 0)
    val stateFlow : StateFlow<Int> = _stateFlow.asStateFlow()

    fun increaseValue(){
        _stateFlow.value +=1
    }
}

@Composable
fun MainScreen(viewmodel: DemoViewModel = viewModel()) {
    val count : Int by viewmodel.stateFlow.collectAsState()

    Column(modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally) { this: ColumnScope
        Text(text="$count", style = TextStyle(fontSize = 40.sp))
        Button(onClick = {viewmodel.increaseValue()}){ this: RowScope
            Text( text: "Click me")
        }
    }
}
```

SharedFlow

- StateFlow보다 좀 더 일반적인 목적의 스트리밍 옵션 제공
- StateFlow와 차이점
 - 소비자를 구독자라 부름
 - 인스턴스를 생성할 때 초기값 제공하지 않음
 - 수집 이전에 방출된 값을 구독자가 수신하게 할 수 있음
 - Value 프로퍼티를 사용하는 대신 값을 방출(emit)
 - Emit통해 데이터를 방출하므로 별도의 coroutine 블록에서 수행

SharedFlow

```
class DemoViewModel : ViewModel(){
    private val _sharedFlow : MutableSharedFlow<Int> = MutableSharedFlow<Int>(
        replay = 10,
        onBufferOverflow = BufferOverflow.DROP_OLDEST
    )
    val sharedFlow : SharedFlow<Int> = _sharedFlow.asSharedFlow()

    fun startSharedFlow(){
        viewModelScope.launch { this: CoroutineScope
            for(i : Int in 1..5){
                _sharedFlow.emit(i)
                delay( timeMillis: 2000)
            }
        }
    }
}
```

```
@Composable
fun MainScreen(viewmodel: DemoViewModel = viewModel()) {
    val count : Int by viewmodel.sharedFlow.collectAsState( initial: 0)
    Column(modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally) { this: ColumnScope
        Text(text="$count", style = TextStyle(fontSize = 40.sp))
        Button(onClick = {viewmodel.startSharedFlow()}){ this: RowScope
            Text( text: "Click me")
        }
    }
}
```

예제) ViewModel 생성

```
class ChatGptViewModel (private val repository: ChatGptRepository)
: ViewModel() {
    private val _response = MutableStateFlow<String>("")
    val response = _response.asStateFlow()
    fun getResponse(query: String) {
        viewModelScope.launch {
            val result = repository.getResponse(query)
            _response.value = result
        }
    }
}
```

뷰모델 생성시 인자 주입하기

- ViewModelProvider.Factory 제공

```
class ChatGptViewModelFactory(private val repository: ChatGptRepository) : ViewModelProvider.Factory {  
    override fun <T : ViewModel> create(modelClass: Class<T>): T {  
        if (modelClass.isAssignableFrom(ChatGptViewModel::class.java)) {  
            return ChatGptViewModel(repository) as T  
        }  
        throw IllegalArgumentException("Unknown ViewModel class")  
    }  
}
```

```
class ChatGptViewModel(private val repository: ChatGptRepository) : ViewModel() {  
    private val _response = MutableStateFlow<String>("")  
    val response: StateFlow<String> = _response  
  
    fun getResponse(query: String) {  
        viewModelScope.launch {  
            val result = repository.getResponse(query)  
            _response.value = result  
        }  
    }  
}
```

```
fun MainScreen(  
    gptViewModel: ChatGptViewModel = viewModel(  
        factory = ChatGptViewModelFactory(ChatGptRepository())  
    )  
) {
```

의존성 주입 (DI : Dependency Injection)

- 객체의 의존관계를 내부에서 결정하는 것이 아니라 객체 외부에서 런타임 시점에 결정하는 방식
 - 코드 재사용 가능 / 리팩터링 편의성 / 테스트 편의성

```
class Car {  
  
    private val engine = Engine()  
  
    fun start() {  
        engine.start()  
    }  
}
```

```
class Car(private val engine: Engine) {  
    fun start() {  
        engine.start()  
    }  
}
```

- 의존성 주입을 자동으로 해주는 라이브러리 Hilt

<https://developer.android.com/training/dependency-injection/hilt-android?hl=ko#setup>

의존성 주입

- 라이브러리 추가
 - com.google.dagger:hilt-android
 - com.google.dagger:hilt-android-compiler
 - androidx.hilt:hilt-navigation-compose
- 프로젝트의 build.gradle 파일

```
plugins { this: PluginDependenciesSpecScope
    alias(libs.plugins.androidApplication) apply false
    alias(libs.plugins.jetbrainsKotlinAndroid) apply false
    id("com.google.dagger.hilt.android") version "2.51.1" apply false
}
```

의존성 주입

- app의 build.gradle 파일

- 플러그인에 추가

```
plugins { this: PluginDependenciesSpecScope
    alias(libs.plugins.androidApplication)
    alias(libs.plugins.jetbrainsKotlinAndroid)
    id("com.google.dagger.hilt.android")
    kotlin("kapt")
}
```

(kapt : kotlin annotation processing tool)

- Dependencies 수정

```
implementation(libs.hilt.android)
kapt(libs.hilt.android.compiler)
implementation(libs.androidx.hilt.navigation.compose)
```

- 코드 생성 허용 구문 추가

```
kapt { this: KaptExtension
    correctErrorTypes = true
}
```

```
plugins {
    id 'kotlin-kapt'
    id("com.google.dagger.hilt.android")
}

android {
    ...
}

dependencies {
    implementation("com.google.dagger:hilt-android:2.44")
    kapt("com.google.dagger:hilt-android-compiler:2.44")
}

// Allow references to generated code
kapt {
    correctErrorTypes = true
}
```

의존성 주입

- Hilt Application 클래스 생성

```
@HiltAndroidApp  
class MainApplication : Application()
```

```
<uses-permission android:name="android.permission.  
<application  
    android:name=".MainApplication"  
    android:allowBackup="true"  
    android:dataExtractionRules="@xml/data_e
```

- Android 클래스에 종속 항목 삽입

- @AndroidEntryPoint 주석이 있는 Android 클래스에 종속 항목을 제공

```
@AndroidEntryPoint  
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?)
```

- Application (@HiltAndroidApp)
- ViewModel (@HiltViewModel)
- Activity
- Fragment
- View
- Service
- BroadcastReceiver

의존성 주입

- 생성자의 매개변수가 종속 항목

```
@Singleton
class ChatGptRepository @Inject constructor() {
    suspend fun getResponse(query: String): String {
        val completionRequest = CompletionRequest(
            model = "gpt-3.5-turbo-instruct",
            prompt = query,
            temperature = 0
        )
    }
}
```

```
@HiltViewModel
class ChatGptViewModel @Inject constructor(private val repository: ChatGptRepository) : ViewModel() {
    private val _response : MutableStateFlow<String> = MutableStateFlow<String>(value: "")
    val response: StateFlow<String> = _response
}
```

```
@Composable
fun MainScreen(newsViewModel: ChatGptViewModel = hiltViewModel()) {
```


수고하셨습니다.