

<https://console.developers.google.com/apis/>

<https://developers.google.com/youtube/v3/getting-started?hl=ko>

구글 API에 접속하기 위해서는 구글 계정이 필요하며, APIs Console로 이동하여 어플리케이션에서 활용할 서비스/API에 대하여 상태를 On으로 변경해야 함.

→ <https://youtu.be/lm69kzhpR3I>

위 링크는 최상단에 적어놓은 페이지에 포함되어있음.

위 링크를 통해 API에 대한 접근을 활성화 시킬 수 있음.

이번 프로젝트에서 예상될만한/사용될만한 리소스 유형 하단에는 현재 지원되는 작업.

- 1) Channel
 - :: 간단한 YouTube채널에 대한 정보를 포함함.
 - :: list
- 2) Playlist
 - :: 기본적인 YouTube 재생 목록을 표시. 사용자들은 공유 할 수있는 동영상의 모음.
 - :: list, insert, update, delete
- 3) PlaylistItem
 - :: 재생 가능한 동영상이 포함되어 있음. 재생 목록 항목 리소스가 포함 됨.
 - :: list, insert, update, delete
- 4) search result
 - :: Youtube동영상, 채널 또는 재생 목록의 정보를 포함함.
 - ::검색 결과는 동영상과 함께 고유한 콘텐츠를 보여줌.
 - :: list
- 5) thumbnail
 - :: 하나의 리소스에 따라 미리보기 이미지를 선택
 - :: X
- 6) video
 - :: 단순한 YouTube 동영상을 표시함.
 - :: list, insert, update, delete

API의 가장 일반적인 메서드는 아래와 같음

- list :: 0개 이상의 리소스 목록을 검색(GET)
- insert :: 새 리소스를 만듦(POST)
- update :: 요청에 포함된 데이터를 반영하도록 기존 리소스를 수정(PUT)
- delete :: 특정 리소스를 삭제(DELETE)

하지만 일부 리소스는 특정/구체적 기능을 위한 메서드를 제공함.

:: 리소스에 대한 삽입/업데이트/삭제 작업에 대해서는 항상 사용자 인증이 필요함.

list메소드의 경우엔 인증/인증안된 요청 모두 지원하며, 경우에 따라서 사용자에게 대한 정보를 제공하거나, 인증되지 않는 요청을 제외한 공개 데이터만을 검색하거나 함.

//

Youtube API는 어플리케이션에 불필요한 데이터를 전송/파싱/저장하지 않도록 부분 리소스 검색을 허용하며, 실제로 이용 및 필요로 함.
→ 효율적인 컴퓨터 자원의 활용.

API는 2개의 요청 매개변수를 지원함.

- part
매개변수는 리소스에 대해 반환되어야 하는 속성 그룹을 식별
 - 리소스를 검색하거나 반환하는 모든 API요청에 필요한 매개변수.
 - 1개 이상의 최상위 수준 리소스 속성을 식별함.
 - 사용할 리소스 자원에 대한 매개변수 지정을 통해 API할당량 관리 및, 지연시간 감소의 효과를 얻을 수 있으며, 대역폭 사용량 역시 감소됨.
- fields
매개변수는 요청된 리소스 부분 내의 특정 속성만 반환하기 위해 API응답 필터링.
 - 필터링을 통한, 응답이 특정 필드 집합만 포함되도록 함.
 - API응답에서 중첩된 속성의 삭제 → 대역폭 사용량 대폭 절감.
 - 여러 필드선택은 쉼표로 구분
 - *는 와일드 카드를 의미.
 - 중첩된 속성 그룹은 괄호를 사용.
 - 중첩속성 식별을 슬래시를 사용.

ex) 재생목록 항목 ID, 제목, 재생목록 모든 항목의 위치 검색

- `fields=items/id,playlistItems/snippet/title,playlistItems/snippet/position`
- `fields=items(id,snippet/title,snippet/position)`
- `fields=items(id,snippet(title,position))`

→

⇒ 이때, fields매개변수값은 URL로 인코딩 되어 함. 주의.

ETag ::캐시 역할을 하는듯?

리소스의 변경이 없다면 해당 리소스에 연결된 ETag를 반환.

304응답

지연시간 및 대역폭 사용량을 줄임

→ API가 캐시되지만 변경되지 않은 리소스 요청에 더 빨리 응답.

→ 지연시간이 줄어들고 대역폭 사용량은 감소.

→ 실수로 다른 API 클라이언트에서 만들어진 리소스에,
변경사항을 덮어쓰는 일이 발생하지 않음.

<https://code.google.com/p/google-api-javascript-client/wiki/GettingStarted?hl=ko>

이곳에서 JavaScript클라이언트 라이브러리를 통해 If-Match와 If-None-Match를 통해 HTTP요청 헤더를 지원, 정상적인 브라우저 캐싱의 컨텍스트 내에서 ETags가 작동 하도록 설정하는 방법을 알 수 있음.

gzip :: 압축 사용을 설정하여 API응답에 필요한 대역폭을 줄임.

:: 압축을 푸는데 CPU자원을 사용하지만,

:: 적은 네트워크 리소스로 인한 장점이 CPU사용 시간 추가의 비용을 능가함.

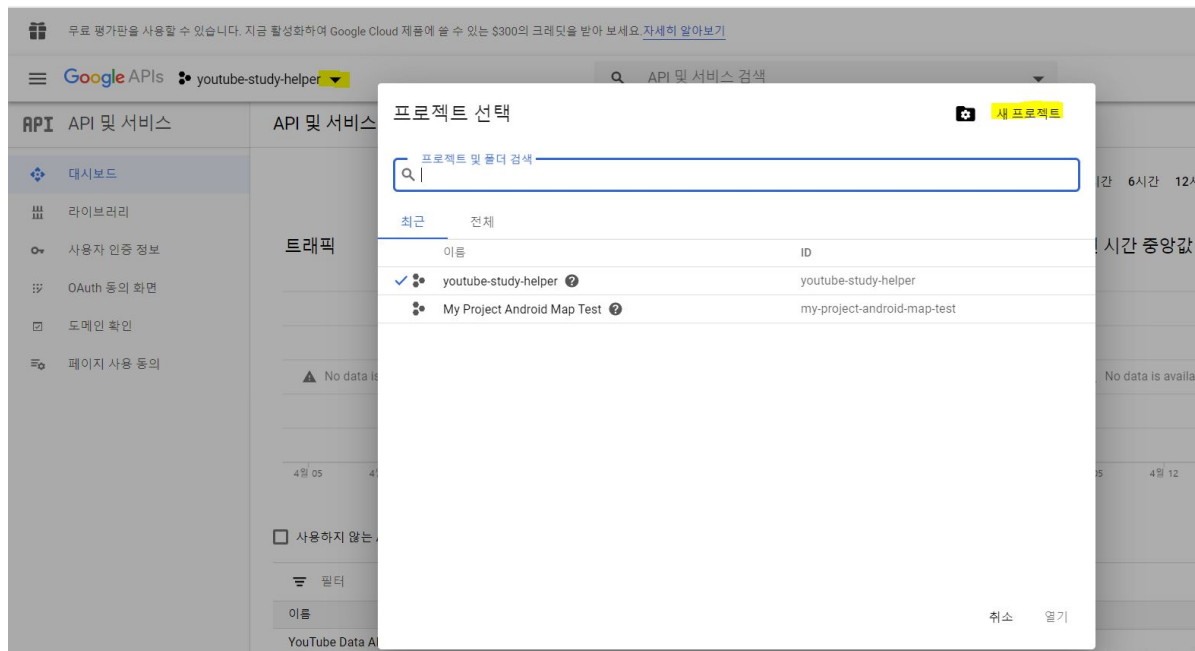
→ gzip에 Accept-Encoding HTTP 요청 헤더를 설정

→ 사용자 에이전트가 gzip 문자열을 포함하도록 수정.

→
Accept-Encoding: gzip
User-Agent: my program (gzip)

=====

<https://console.developers.google.com/apis/>



위의 Console.developers.google.com으로 들어간 뒤 사진의 노란 형광펜으로 칠해놓은부분을 눌러 새로운 프로젝트를 생성해 주세요.

Google APIs API 및 서비스 검색

새 프로젝트

projects 할당량이 9개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)

[MANAGE QUOTAS](#)

프로젝트 이름 *
TempProject ?

프로젝트 ID: tempproject-276006입니다. 나중에 변경할 수 없습니다. [수정](#)

위치 *
 조직 없음 [찾아보기](#)

상위 조직 또는 폴더

[만들기](#) [취소](#)

사용할 프로젝트 이름을 지정합니다.

이름을 지정 후 만들기를 눌러주세요

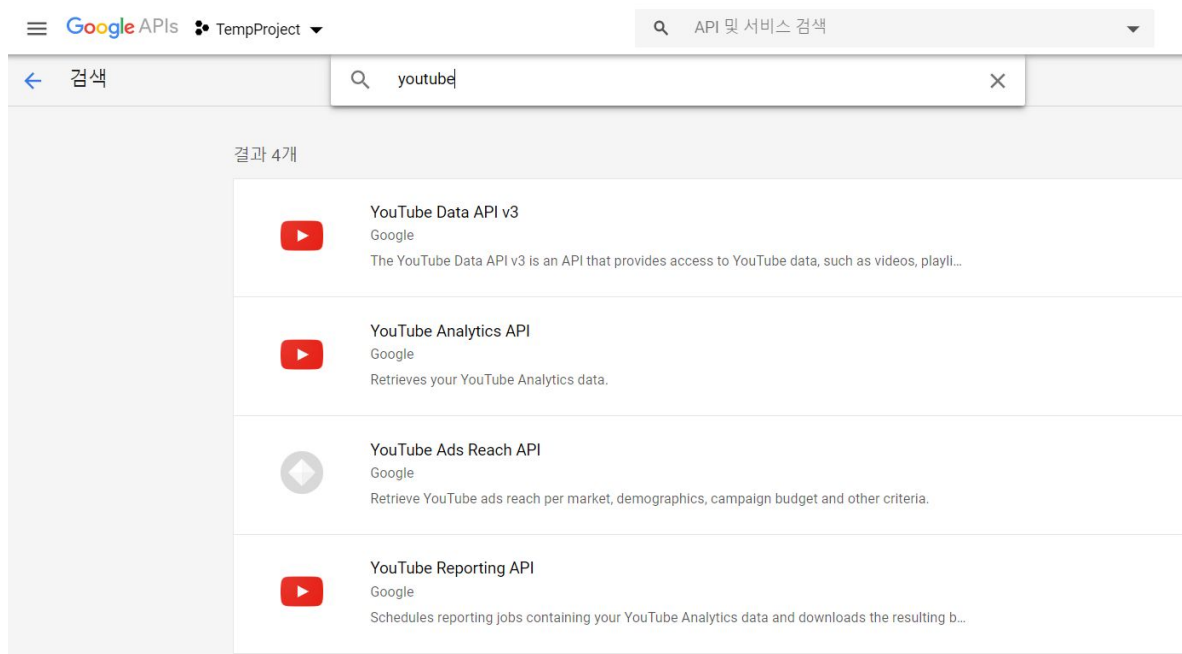
조금 기다리시면 프로젝트가 생성되며, 우측에 프로필부분에 로딩되고 있음을 확인 할 수 있습니다.



새 프로젝트가 생성되었습니다.

어떤 서비스를 사용할 것인지를 추가해야 합니다

API 및 서비스 사용 설정을 눌러주세요



어떤 서비스를 사용할 것인지를 선택하게 됩니다

우리의 경우에는 youtube API를 사용하려고 하기 때문에 YouTube Data API v3를 선택합니다.

선택후 사용설정을 누르면 설정이 완료됩니다.

OAuth 동의 화면

대상 사용자를 비롯해 앱을 구성하고 등록하려는 방식을 선택하세요. 프로젝트에는 하나의 앱만 연결할 수 있습니다.

User Type

☐ 내부 ?

G Suite 사용자가 아니기 때문에 앱을 외부(일반 잠재고객) 사용자에게 제공하는 것만 가능합니다.

외부 앱을 제출할 필요는 없

☒ 외부 ?

Google 계정이 있는 모든 사용자가 사용할 수 있습니다.

만들기

외부를 선택합니다. 내부를 선택하기 위해서는 G Suite 사용자여야만 합니다.

OAuth 동의 화면

사용자는 인증 전에 이 동의 화면을 통해 비공개 데이터에 대한 액세스 권한을 부여할지 선택할 수 있으며 서비스 약관 및 개인정보처리방침 링크도 제공됩니다. 이 페이지에서 프로젝트에 속한 모든 애플리케이션의 동의 화면을 구성합니다.

확인 상태
게시되지 않음

애플리케이션 이름 ?
동의를 요청하는 앱의 이름

youtube-study-helper

애플리케이션 로고 ?
사용자가 앱을 알아보는 데 도움이 되는 동의 화면의 이미지입니다.

업로드용 로컬 파일

탐색



지원 이메일 ?
사용자 지원을 위해 동의 화면에 표시됩니다.

vrimd1017@gmail.com

Google API의 범위
범위를 통해 애플리케이션에서 사용자의 비공개 데이터에 액세스할 수 있습니다. [자세히 알아보기](#)

캘린더 또는 드라이브에 대한 전체 액세스 권한을 부여하는 민감한 범위를 추가하는 경우에

어플리케이션의 이름을 입력하고 제일 아래의 등록을 눌러줍니다.
중간에 API의 범위를 설정하거나
도메인을 등록하는 란이 있는데, 이부분은 확실치 않아..

동의 화면 정보

동의 화면에서 사용자에게 데이터 하는 당사자 및 액세스 권한이 필요를 안내합니다.

OAuth 확인

나와 내 사용자를 보호하기 위해 G면과 애플리케이션을 확인해야 할 공개로 표시되어 있고 다음 중 하나우 확인이 필요합니다.

- 앱에서 민감하거나 제한된 범
- 앱의 OAuth 동의 화면에 아이
- 앱에 승인된 도메인이 많이 있
- 이전에 확인된 OAuth 동의 회

확인 절차에 최대 몇 주 정도 걸릴 : 이메일 업데이트가 전송됩니다. [확인하기](#)

Google에서 동의 화면과 애플리케이션이라도 제한사항이 적용된 상태에, 계속 테스트할 수 있습니다. 앱을 사용하는 방식 [자세히 알아보기](#)

OAuth 경험에 대한 의견 보내기

OAuth 동의 화면

← OAuth 클라이언트 ID 만들기

OAuth 2.0 프로토콜을 사용하여 Google API를 호출하는 애플리케이션의 경우 OAuth 2.0 클라이언트 ID로 액세스 토큰을 생성할 수 있습니다. 토큰에 고유 식별자가 포함됩니다. 자세한 내용은 [OAuth 2.0 설정](#)을 참조하세요.

애플리케이션 유형

☒ 웹 애플리케이션

☐ Android [자세히 알아보기](#)

앱의 표시 이름이 아닌 OAuth 클라이언트 ID의 이름입니다.

이름

youtube-study-helper

제한사항

자바스크립트 출처와 리디렉션 URI 중 1개를 입력하거나 둘 다 입력하세요. [자세히 알아보기](#)

OAuth 등의 설정의 승인된 도메인 목록에 출처와 리디렉션 도메인을 추가해야 합니다.

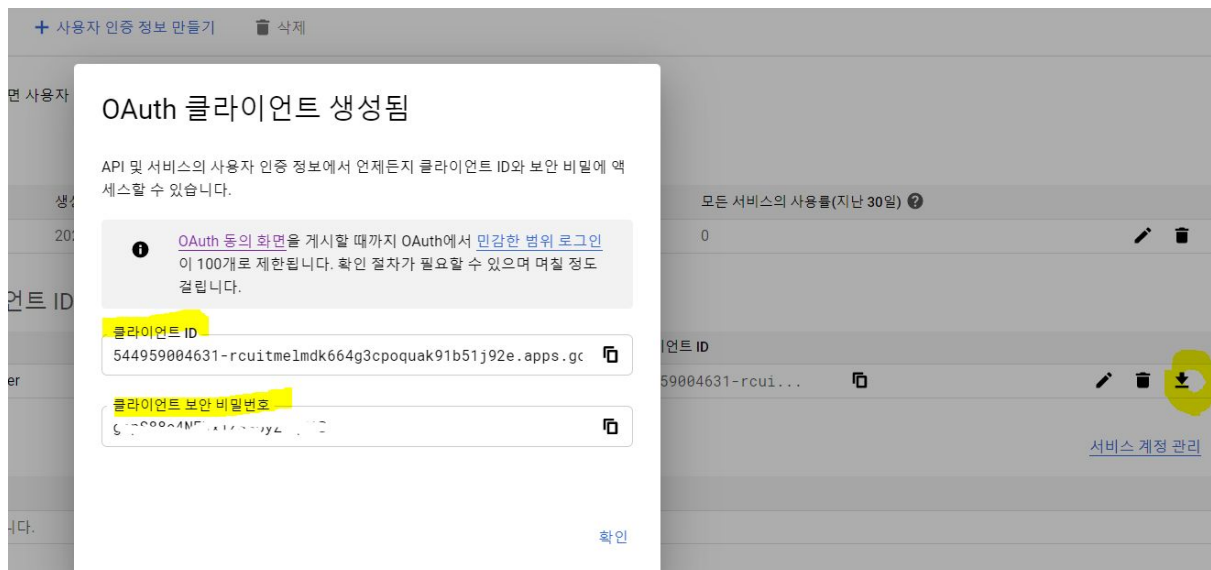
승인된 자바스크립트 원본

브라우저의 요청에 따라 사용되며 클라이언트 애플리케이션의 원본 URI입니다. 와일드 카드(https://*.example.com)나 경로(<https://example.com/subdir>)를 포함할 수 없습니다. 비표준 포트를 사용하고 있다면 이 포트를 원본 URI에 포함해야 합니다.

<https://www.example.com>

도메인을 입력하고 Enter 키를 눌러 추가하세요.

다시 이전과 같이 OAuth 클라이언트 ID만들기를 선택해주면 위와같은 화면이 등장합니다. 이름은 OAuth클라이언트 ID의 이름을 의미한다고 되어있으며, 제 경우에는 프로젝트의 이름과 동일하게 설정해 주었습니다. 어플리케이션의 유형은 웹 어플리케이션을 설정해 주었으며, 생성을 눌러줍니다.



OAuth클라이언트가 생성되었다는 팝업과 함께 클라이언트 ID와 비밀번호secret이 보여집니다.

이는 잘 보관해야 하며, 사진의 우측에 표시된 버튼을 클릭하여 정보를 다운 받을 수 있습니다.

다운받은 json파일에는 ClientID, ClientSecret 등의 정보가 담겨있으며, 이를 프로젝트 폴더에 넣어놓고 잘 보관합니다.

음....

<https://www.youtube.com/playlist?list=PLaNOFHMrtHWTXwwI5MEehcegEn29ij8XQ>

생활코딩의 강의를 보시게 되면 html의 form과 php코드를 통해서 값을 전송하고 받아들이어서 실제 어떻게 처리되는지를 알려주십니다.

하지만 마지막으로 언급하셨던것 처럼 보안 또는 기타 여러 이유로 SDK를 이용하는것이 좋다고 말 하셨으며, 우리의 프로젝트에서는 Spring의 JAVA 혹은 React/Node.js 내에서 사용될 예정이므로 두가지 방법을 통해서, 일단은 몸으로 부딪혀 보려고 합니다.

<https://developers.google.com/youtube/v3/libraries>

SDK, youtube api를 사용하는데 도움을 줄 library들을 모아놓은 곳 입니다.
이곳에 들어가서 웹 어플리케이션에서 사용 할 언어에 따라 알맞은 SDK를 선택합니다.
우선은 Spring/JAVA로...

https://developers.google.com/youtube/v3/code_samples/java?hl=ko

이것저것 뒤져보고 생활코딩 영상에서 나왔던 API Client Library라던가 다 봐봤는데 아무래도 저걸 보고 참고해서 해야될것 같습니다.
youtube에도 youtube API v3 in JAVA 와 같은 키워드로 검색해도 정확히 나오지는 않고..

설치

Java 용 Google API 클라이언트 라이브러리는 설치가되었습니다.

pom.xml 파일에 다음 행을 추가하십시오.

```
<project>
  <dependencies>
    <dependency>
      <groupId>com.google.api-client</groupId>
      <artifactId>google-api-client</artifactId>
      <version>1.30.9</version>
    </dependency>
  </dependencies>
</project>
```

그렇습니다 Maven은 빛과도 같았습니다...

위 홈페이지에서 예제가 담긴 github repository를 링크로 걸어져있었습니다.

하단의 프로젝트 readme를 보니 설치 해놓고

<dependency>가 표기되있는데...

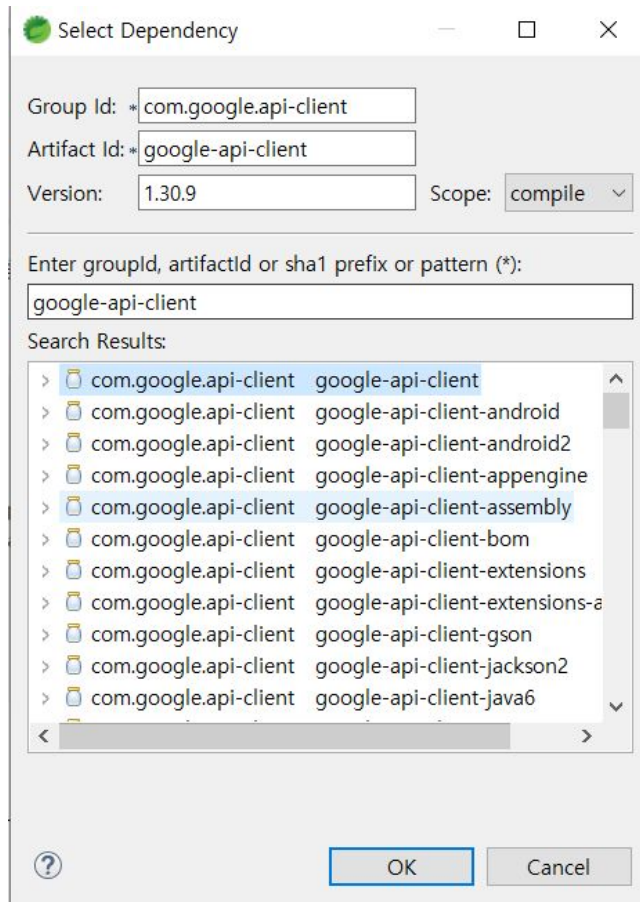
아무래도 이를 통대로 스프링에서 사용 할 수 있으며,

이전에 다운받아놓은 ClientID와 ClientSecret을 사용하여 어떻게 해주면 될거 같습니다.

스프링을 킬게요...

STS로 spring legacy project를 하나 생성하고 Spring MVC Project로 프로젝트를 생성합니다.

프로젝트의 생성이 완료되면 pom.xml을 통해 위의 dependency를 추가해줍니다.



추가해주시고 저장해주시면 로딩이 되면서 의존전이 파팍 아마 관련 파일이 다 들어갈겁니다.

생각보다 오래 걸리네 이걸...

https://developers.google.com/youtube/v3/code_samples/java?hl=ko

이곳의 키워드별 검색 부분을 예제로 받아 Spring에서 실행 해 보려고 함.

google-api-client : 1.30.9

google-api-services-youtube : v3-rev20200402-1.30.9

google-http-client-jackson2 : 1.34.2

세가지 dependency를 추가해줌.

처음에 pom.xml을 보고도 뭘 추가해야 될지 감이 잘 안잡혀서...

실제 코드를 STS에 붙여놓고 import문에서 오류가 뜨는부분을 dependency로 잡아줬음.

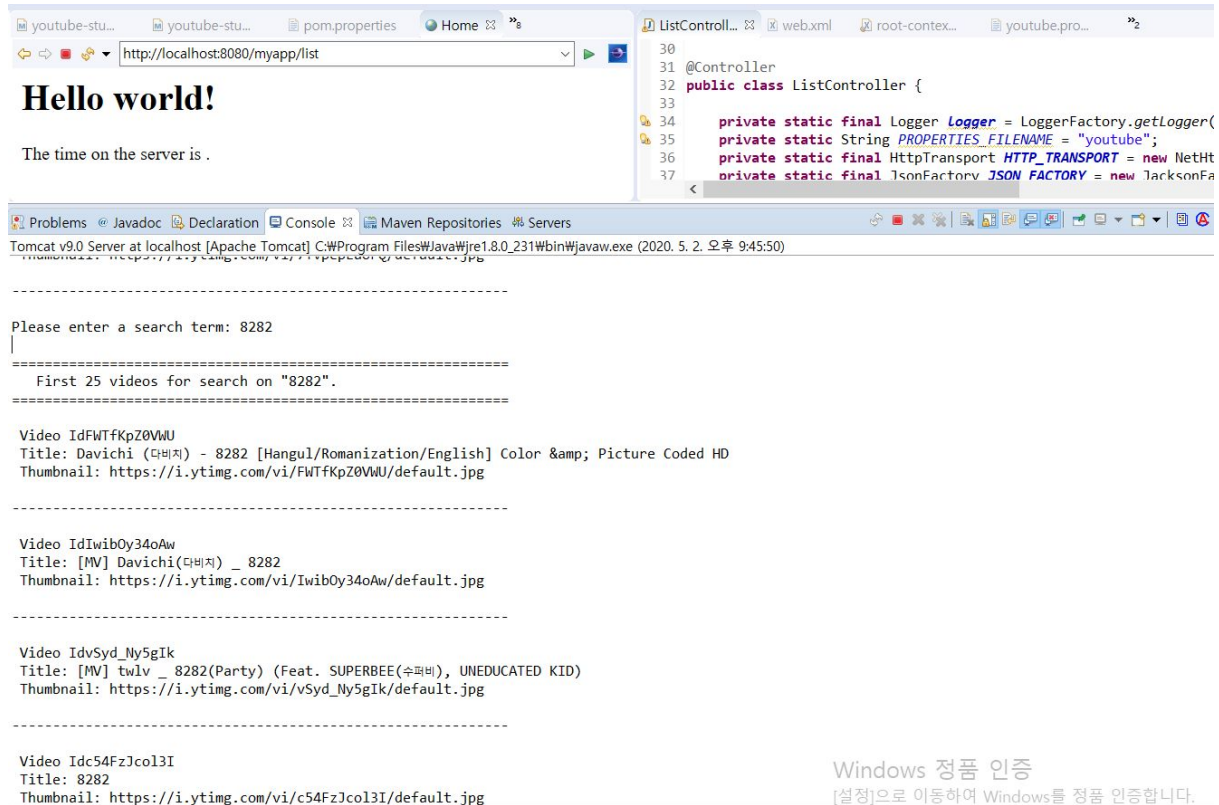
예제 코드가 java라 main문이 있는거때문에 멍을 좀 때리다가 그냥 JAVA파일로 만들었더니 의존전이란관련해서 파일간 의존문제가...

다시 STS로 돌아와서 dependency를 추가하였음.

main을 지워버리고 Controller를 하나 생성하여 해당 컨트롤러를 사용하였음.

중간에 Properties파일이 load가 안되서 이유는 모르겠는데

일단 지워버리고 api키를 String으로 선언하여 프로그램을 실행하였습니다.
일단 동작이 되는지를 봐야되니까요



동작은 성공적입니다
Video의 고유값, 타이틀, 그리고 썸네일을 뽑아내었습니다.

목표.

- 1) OAuth2.0을 이용, 사용자의 권한 승낙
- 2) Client ID, Client Secret, Authentic Code를 이용하여 AccessKey 획득
- 3) 사용자의 재생목록을 리스트 확인
- 4) 재생목록 리스트에서 각각의 동영상 정보 확인.

<https://developers.google.com/youtube/v3/guides/auth/server-side-web-apps>
<https://developers.google.com/youtube/v3/guides/implementation/playlists?hl=ko>
<https://developers.google.com/youtube/v3/guides/implementation/partial?hl=ko>
<https://developers.google.com/youtube/v3/docs/playlistItems?hl=ko>
<https://developers.google.com/youtube/v3/docs/playlists?hl=ko>
<https://developers.google.com/youtube/v3/docs/videos?hl=ko>

첫번째 링크로 가서 OAuth에 관련된 작업을 수행합니다.

```
https://accounts.google.com/o/oauth2/auth?
client_id=1084945748469-eg34imk572gdhu83gj5p0an9fut6urp5.apps.googleusercontent.com&
redirect_uri=http%3A%2F%2Flocalhost%2Foauth2callback&
scope=https://www.googleapis.com/auth/youtube&
response_type=code&
access_type=offline
```

사용자에 인증을 유도하는 Sample url입니다.

몇가지 선택적 추가 파라미터가 존재하며, 지금은 우선 이거로 테스트를 해 보도록 하겠습니다.

client_id 에는 이전에 프로젝트를 등록하면서 다운받았던 클라이언트 ID의 값승르.

redirect_uri에는 인증이 되고나서 Resource Server에서 어디로 다시 리다이렉트 해 줄것인지를,

access_type은 해당 토큰이 refresh token을 발급 받을것인지에 대한 내용입니다.

아마 추가적인 설정사항이 존재할것으로 보이며,

scope와 response_type의 경우에는 문서를 참고 바랍니다.

- + 이때, 링크를 통해 갈것이기때문에 uri를 전부 인코딩 시켜 줘야합니다.
- + <https://www.convertstring.com/ko/EncodeDecode/UrlEncode>
- + 이런사이트를 통해 uri에 직접 넣을경우에는 인코딩해 사용하시면 되며,
- + form등을 사용하여 input으로 보낼경우에는 그냥 그대로 작성하시면됩니다.
- + 또한, redirect_uri는 인증된 페이지로밖에 보낼 수 없기때문에 아래와 같이
- + redirect_uri를 신뢰할만한 페이지로 등록해야합니다.

API Console interface showing the '사용자 인증 정보' (User Consent Information) page for the 'youtube-study-helper' project. The page includes fields for '이름 *' (Name) and '클라이언트 ID'. A note states: 'OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.' Below this, there is a section for '승인된 자바스크립트 출처' (Authorized JavaScript origins) with a '+ URI 추가' button. Another section for '승인된 리디렉션 URI' (Authorized redirect URIs) is circled in yellow, also featuring a '+ URI 추가' button. The left sidebar shows the 'API & Services' menu with '사용자 인증 정보' selected.

API Console로 가셔서 사용자인증정보 - 프로젝트 선택을 하게되면 위와같은 창을 볼 수 있습니다.

승인된 리디렉션 URI에 Owner의 승인 이후 어떤 페이지를 보여줄것인지,

그리고 승인의 결과인 Code를 어디에서 받을것인지를 포함시켜 주어야 합니다.

