

# Entity and Sentiment Analysis with the Natural Language API

허성실

# Overview

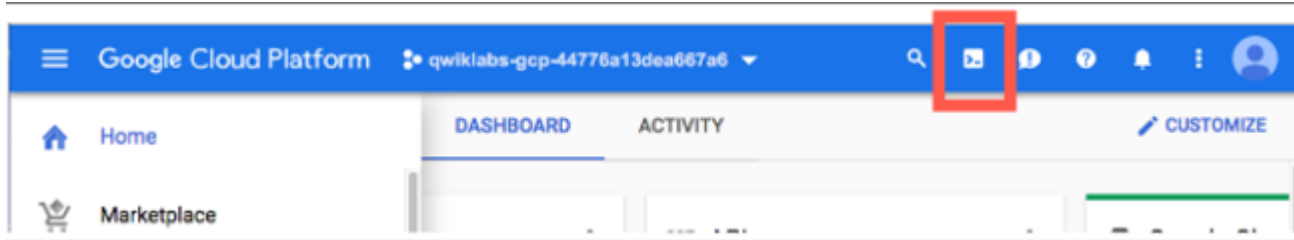
- 퀵랩 2와 유사. The Cloud Natural Language API를 이용해 텍스트로부터 엔터티를 추출하고, 정서를 파악하거나 텍스트를 카테고리화시키는 등의 일을 수행한다.
- 무엇을 배우나?
  - NL API 요청을 생성하고 curl로 API를 불러온다.
  - 엔터티를 추출하고 NL API로 정서 분석을 수행한다.
  - NL API를 사용하여 텍스트의 언어 분석을 수행한다.
  - 다른 언어로 된 NL API 요청을 만든다.
- 무엇이 필요?
  - Google Cloud Platform Project
  - A Browser such as Chrome or Firefox

# Setup and Requirements

- 1. Start lab 버튼 클릭
- 2. 왼쪽에 username 복사해서 Open Google Console을 클릭
- 3. 사용자 계정에 Use Another Account 선택
- 4. 복사한 username 입력, password도 복사 후 입력
- 5. Next 버튼 클릭
- 6. terms and conditions accepts 하기, recovery option은 temporary 계정이니 건너뛰고 free trials을 sign up 하지 않기

# The Google Cloud Shell

- 8. 빨간 박스 클릭하여 Google Cloud Shell 프로그램 실행

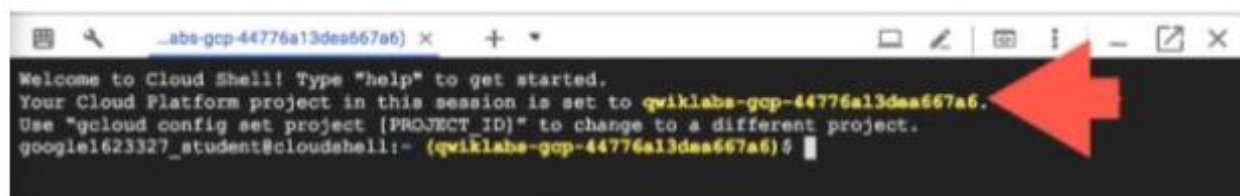


- 9. STRAT CLOUD SHELL 클릭



# The Google Cloud Shell

- 10. 본인의 PROJECT\_ID로 프로젝트가 생성될 것이다.

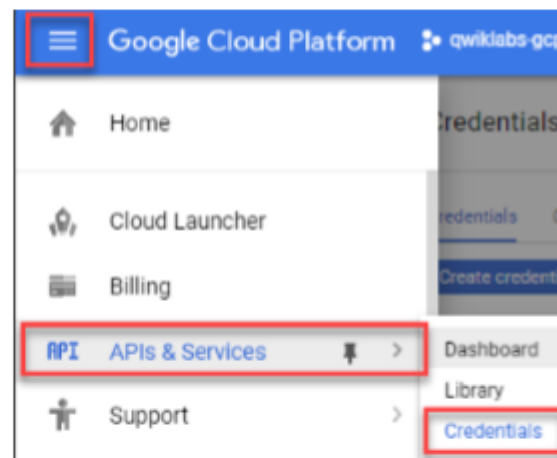


The screenshot shows a terminal window titled "abs-gcp-44776a13dea667a6". The terminal text reads: "Welcome to Cloud Shell! Type 'help' to get started. Your Cloud Platform project in this session is set to **qwiklabs-gcp-44776a13dea667a6**. Use 'gcloud config set project [PROJECT ID]' to change to a different project. google1623327\_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6) \$". A red arrow points to the project ID "qwiklabs-gcp-44776a13dea667a6".

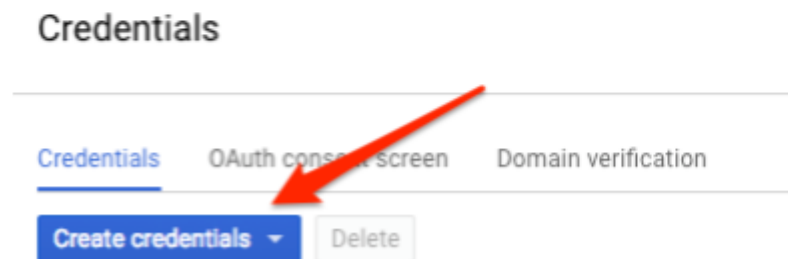
- 11. 다음의 명령어로 활성화된 계정 이름들을 볼 수 있다.  
gcloud auth list
- 12. Project ID들을 볼 수 있다.  
gcloud config list project

# Create an API Key

- 13. Speech API의 요청을 전송하기 위해 curl을 이용할 것이기 때문에, URL 요청을 받기 위해서는 API 키를 생성해야 한다. 본인의 PROJECT\_ID로 프로젝트가 생성될 것이다.
- 14. Navigation menu -> APIs & Services -> Credentials

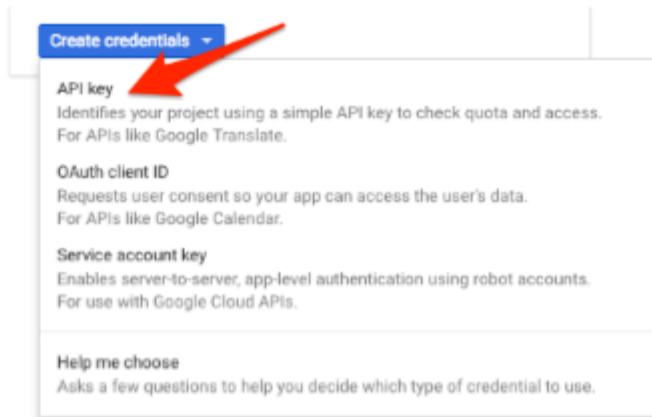


- 15. Create credentials 클릭



# Create an API Key

- 16. 드롭다운 메뉴에서 API key 선택 -> 생성된 키 복사



- 17. 환경변수 설정해주기, 복사한기를 <YOUR\_API\_KEY> 부분에 붙여넣기

```
export API_KEY=<YOUR_API_KEY>
```

# Make an Entity Analysis Request

- 18. 우리가 사용할 NL API의 첫번째 메소드는 analyzeEntities이다. 이 메소드를 사용하면 API가 텍스트에서 인물, 장소 및 이벤트와 같은 엔터티를 추출할 수 있다. “request.json” 파일을 생성하여 NL API의 요청을 작성하자. 연필 아이콘을 클릭하면 기본 명령 편집기를 사용할 수 있다.



```
{
  "document":{
    "type":"PLAIN_TEXT",
    "content":"Joanne Rowling, who writes under the pen names J. K.
Rowling and Robert Galbraith, is a British novelist and screenwriter who
wrote the Harry Potter fantasy series."
  },
  "encodingType":"UTF8"
}
```

Type은 “PLAIN\_TEXT”와  
“HTML”두가지가 있다.



# Call the Natural Language API

- 19. 다음의 명령어를 수행하라

```
curl "https://language.googleapis.com/v1/documents:analyzeEntities?  
key=${API_KEY}" \  
-s -X POST -H "Content-Type: application/json" --data-binary  
@request.json
```

- 20. 응답결과는 다음과 같다.(퀵랩 2와 같음)

Name: 개체 이름  
Type: 말그대로 타입  
Metadata: 연관된 위키피디아 정보  
나타냄  
Salience: 전체 텍스트에서 얼마나  
중요한 비중을 차지하나  
Mentions: 전체 텍스트에서, 이  
개체와 동일한 개체?

```
{  
  "entities": [  
    {  
      "name": "Robert Galbraith",  
      "type": "PERSON",  
      "metadata": {  
        "mid": "/m/042xh",  
        "wikipedia_url": "https://en.wikipedia.org/wiki/J._K._Rowling"  
      },  
      "salience": 0.7980405,  
      "mentions": [  
        {  
          "text": {  
            "content": "Joanne Rowling",  
            "beginOffset": 0  
          },  
          "type": "PROPER"  
        },  
        {  
          "text": {  
            "content": "Rowling",  
            "beginOffset": 10  
          },  
          "type": "PROPER"  
        }  
      ]  
    }  
  ]  
}
```

# Sentiment analysis with the Natural Language API

- 21. “request.json” 파일을 다음과 같이 수정하라.

```
{
  "document":{
    "type":"PLAIN_TEXT",
    "content":"Harry Potter is the best book. I think everyone should
read it."
  },
  "encodingType": "UTF8"
}
```

- 22. 다음의 명령어를 수행하라(analyzeSentiment)

```
curl "https://language.googleapis.com/v1/documents:analyzeSentiment?
key=${API_KEY}" \
-s -X POST -H "Content-Type: application/json" --data-binary
@request.json
```

# Sentiment analysis with the Natural Language API

- 23. 다음의 결과가 나타난다.

```
{
  "documentSentiment": {
    "magnitude": 0.8,
    "score": 0.4
  },
  "language": "en",
  "sentences": [
    {
      "text": {
        "content": "Harry Potter is the best book.",
        "beginOffset": 0
      },
      "sentiment": {
        "magnitude": 0.7,
        "score": 0.7
      }
    },
    {
      "text": {
        "content": "I think everyone should read it.",
        "beginOffset": 31
      },
      "sentiment": {
        "magnitude": 0.1,
        "score": 0.1
      }
    }
  ]
}
```

Score : -1.0 부터 1.0까지 나타내며  
부정적인 것부터 긍정적인 것까지  
나타낸다.

Magnitude: 0 ~무한대의 범위를 가지며,  
각 감정에 대한 강도를 나타낸다.

Sentences: 문장단위로 끊어서 감정  
분석을 해준다.

# Analyzing entity sentiment

- 24. 정서를 분석해보자. 다음 문장을 보자 “나는 스시를 좋아했지만 서비스는 끔찍했다.” 이 경우, score을 얻는 것은 별로 유용하지 않을 것이다. 하지만, 엔터티에 대한 감정을 얻을 수 있는 메서드를 쓰게 되면 유용한 정보가 될 것이다. “request.json”을 다음과 같이 업데이트 하자.

```
{
  "document":{
    "type":"PLAIN_TEXT",
    "content":"I liked the sushi but the service was terrible."
  },
  "encodingType": "UTF8"
}
```

- 25. 그리고 다음의 명령어를 입력하자.

```
curl
"https://language.googleapis.com/v1/documents:analyzeEntitySentiment?
key=${API_KEY}" \
-s -X POST -H "Content-Type: application/json" --data-binary
@request.json
```

# Analyzing entity sentiment

- 26. sushi는 0.9점을 얻었으나, service는 -0.9점을 얻었음을 알 수 있다.

```
{
  "entities": [
    {
      "name": "sushi",
      "type": "CONSUMER_GOOD",
      "metadata": {},
      "salience": 0.52716845,
      "mentions": [
        {
          "text": {
            "content": "sushi",
            "beginOffset": 12
          },
          "type": "COMMON",
          "sentiment": {
            "magnitude": 0.9,
            "score": 0.9
          }
        }
      ],
      "sentiment": {
        "magnitude": 0.9,
        "score": 0.9
      }
    }
  ],
  "language": "en"
}
```

```
{
  "name": "service",
  "type": "OTHER",
  "metadata": {},
  "salience": 0.47283158,
  "mentions": [
    {
      "text": {
        "content": "service",
        "beginOffset": 26
      },
      "type": "COMMON",
      "sentiment": {
        "magnitude": 0.9,
        "score": -0.9
      }
    }
  ],
  "sentiment": {
    "magnitude": 0.9,
    "score": -0.9
  }
},
  "language": "en"
}
```

# Analyzing syntax and parts of speech

- 27. “request.json” 파일의 내용을 교체하자.

```
{
  "document":{
    "type":"PLAIN_TEXT",
    "content": "Joanne Rowling is a British novelist, screenwriter and
film producer."
  },
  "encodingType": "UTF8"
}
```

- 28. 다음의 명령어를 입력하자.

```
curl "https://language.googleapis.com/v1/documents:analyzeSyntax?
key=${API_KEY}" \
  -s -X POST -H "Content-Type: application/json" --data-binary
@request.json
```

# Analyzing syntax and parts of speech

- 29. 다음의 응답 메시지가 나타난다.

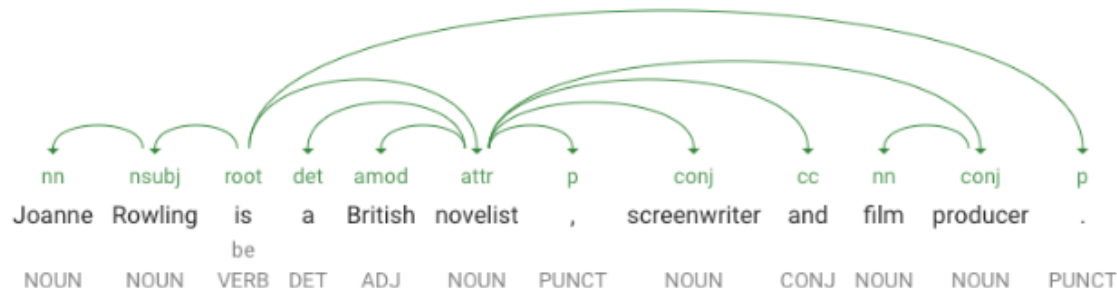
```
{
  "text": {
    "content": "is",
    "beginOffset": 15
  },
  "partOfSpeech": {
    "tag": "VERB",
    "aspect": "ASPECT_UNKNOWN",
    "case": "CASE_UNKNOWN",
    "form": "FORM_UNKNOWN",
    "gender": "GENDER_UNKNOWN",
    "mood": "INDICATIVE",
    "number": "SINGULAR",
    "person": "THIRD",
    "proper": "PROPER_UNKNOWN",
    "reciprocity": "RECIPROCITY_UNKNOWN",
    "tense": "PRESENT",
    "voice": "VOICE_UNKNOWN"
  },
  "dependencyEdge": {
    "headTokenIndex": 2,
    "label": "ROOT"
  },
  "lemma": "be"
},
```

Content : 분석 대상  
토큰 명시

품사 정보 : "is"는 동사임

headTokenIndex : 이 토큰이 의존성 파스 트리에서 Joanne을  
가리키는 인덱스  
Label: 문장에서 해당 토큰의 역할

원형: be



# Multilingual natural language processing

- 30. “request.json”을 다음과 같이 교체한다.

```
{  
  "document": {  
    "type": "PLAIN_TEXT",  
    "content": "日本のグーグルのオフィスは、東京の六本木ヒルズにあります"  
  }  
}
```

- 31. 다음의 명령어를 입력하자.

```
curl "https://language.googleapis.com/v1/documents:analyzeEntities?  
key=${API_KEY}" \  
-s -X POST -H "Content-Type: application/json" --data-binary  
@request.json
```



# Multilingual natural language processing

- 32. 결과

```
{
  "entities": [
    {
      "name": "日本",
      "type": "LOCATION",
      "metadata": {
        "mid": "/m/03_3d",
        "wikipedia_url": "https://en.wikipedia.org/wiki/Japan"
      },
      "salience": 0.23854347,
      "mentions": [
        {
          "text": {
            "content": "日本",
            "beginOffset": 0
          },
          "type": "PROPER"
        }
      ]
    },
    {
      "name": "グーグル",
```

Name: 개체 이름  
Type: 말그대로 타입  
Metadata: 연관된 위키피디아 정보  
나타냄  
Salience: 전체 텍스트에서 얼마나  
중요한 비중을 차지하나  
Mentions: 전체 텍스트에서, 이  
개체와 동일한 개체?