

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO NIÊN LUẬN CƠ SỞ NGÀNH
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài
GOOGLE CLOUD TRANSLATION API

Sinh viên thực hiện: Dương Lữ Điện
Mã số: B1401133
Khoá: 40

Cần Thơ, 11/2017

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO NIÊN LUẬN CƠ SỞ NGÀNH
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài
GOOGLE CLOUD TRANSLATION API

Giảng viên hướng dẫn:
TS. Phạm Thế Phi

Sinh viên thực hiện: Dương Lữ Điện
Mã số: B1401133
Khoá: 40

Cần Thơ, 11/2017

[illegible]

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn, TS. Phạm Thế Phi đã đề xuất cho em đề tài này, em đã tích lũy được rất nhiều kiến thức và kinh nghiệm trong quá trình thực hiện đề tài. Con xin gửi lời cảm ơn đến Bà Cô, ông bà nội ngoại, cha mẹ luôn bên cạnh nhắc nhở, khuyến khích và làm hậu phương vững chắc cho con trên con đường học tập.

Cần Thơ, ngày 10 tháng 11 năm 2017

Người viết

Dương Lữ Điện

Mục lục

1	GIỚI THIỆU VỀ GOOGLE TRANSLATION API	1
1.1	Đăng ký sử dụng Google Cloud Platform	1
1.2	Tạo một project trên Google API Console	1
1.3	Kích hoạt Google Cloud Translation API	3
1.4	API key	3
1.5	Tạo một project trên Eclipse	4
2	DỊCH VĂN BẢN	7
2.1	HTTP request	7
2.2	Kết quả trả về	8
2.3	TranslateTextResponseList	8
2.4	Dịch văn bản	8
2.4.1	Dịch chuỗi văn bản	8
2.4.2	Sử dụng một mô hình cụ thể	9
2.5	Dịch vụ Translate trong Java	10
3	CÁC NGÔN NGỮ ĐƯỢC HỖ TRỢ	11
3.1	HTTP request	11
3.2	Kết quả trả về	11
3.3	GetSupportedLanguagesResponseList	12
3.4	Các ngôn ngữ được hỗ trợ	12
3.4.1	Nhận về danh sách các ngôn ngữ được hỗ trợ	12
3.4.2	Nhận về danh sách các ngôn ngữ được hỗ trợ dưới dạng một ngôn ngữ	13
3.5	Dịch vụ Supported Languages trong Java	14
4	NHẬN DIỆN NGÔN NGỮ	15
4.1	HTTP request	15
4.2	Kết quả trả về	15
4.3	DetectLanguageResponseList	16
4.4	Nhận diện ngôn ngữ	16
4.4.1	Nhận diện ngôn ngữ trên một chuỗi	16
4.4.2	Nhận diện ngôn ngữ trên nhiều chuỗi	17
4.5	Dịch vụ Detecting Language trong Java	17

5 KẾT LUẬN	
VÀ HƯỚNG PHÁT TRIỂN	19
5.1 Kết luận	19
5.2 Hướng phát triển	19

Danh sách hình vẽ

1.1	Điền đầy đủ thông tin Google yêu cầu	2
1.2	Thông báo của Google khi đăng ký thành công	2
1.3	Tạo một project trên Google Cloud Platform	3
1.4	Kích hoạt dịch vụ Translation	3
1.5	Tạo một project Maven trên Eclipse	4
1.6	Thiết lập Archetype Parameters	5
1.7	Thêm Dependencies	6

Chương 1

GIỚI THIỆU VỀ GOOGLE TRANSLATION API

Google Translate là một dịch vụ dịch thuật được phát triển bởi Google, hỗ trợ trên 100 ngôn ngữ khác nhau. Số liệu thống kê vào tháng 05 năm 2013 cho thấy, Google Translate phục vụ dịch thuật cho hơn 200 triệu người mỗi ngày.

Google Cloud Translation API là một API thuộc nhóm Cloud Machine Learning API. Cloud Translation API cung cấp một giao diện lập trình ứng dụng đơn giản để lập trình viên có thể tích hợp tính năng phiên dịch (Translation) của Google vào website hoặc ứng dụng của họ. Với tính tương thích cao, các websites và ứng dụng tích hợp Google Translation hoạt động rất hiệu quả trong việc phiên dịch từ một ngôn ngữ sang một ngôn ngữ khác (Ví dụ: từ tiếng Anh sang tiếng Việt). Ngoài ra, tính năng phát hiện ngôn ngữ cũng hoạt động khá hiệu quả khi ngôn ngữ gốc chưa được xác định.

Trong tài liệu này, chúng tôi sử dụng Google Cloud Translation API version 2 revision 51. Minh họa bằng ngôn ngữ lập trình Java.

1.1 Đăng ký sử dụng Google Cloud Platform

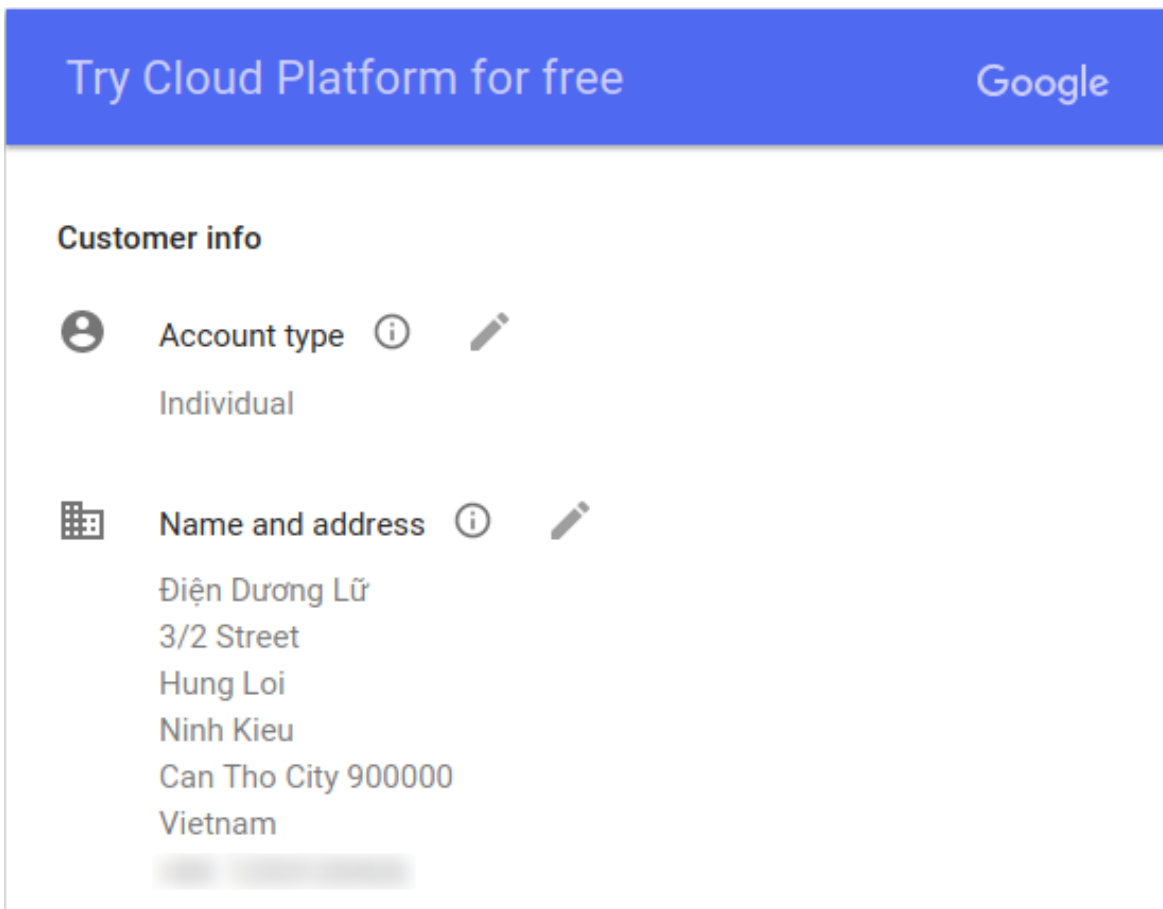
Để truy cập đến các dịch vụ của Google Cloud Platform, ta sử dụng tài khoản Google. Để nhận \$300 credit dùng thử trong 1 năm từ Google, ta truy cập vào link:

`https://console.cloud.google.com/freetrial`

và điền đầy đủ các thông tin mà Google yêu cầu, bao gồm cả thông tin thẻ tín dụng (Hình 1.1 và 1.2).




1.2 Tạo một project trên Google API Console

1. Truy cập vào Google Cloud Platform Console.
2. Tạo một project tên là “translation” (Hình 1.3).






Try Cloud Platform for free Google

Customer info

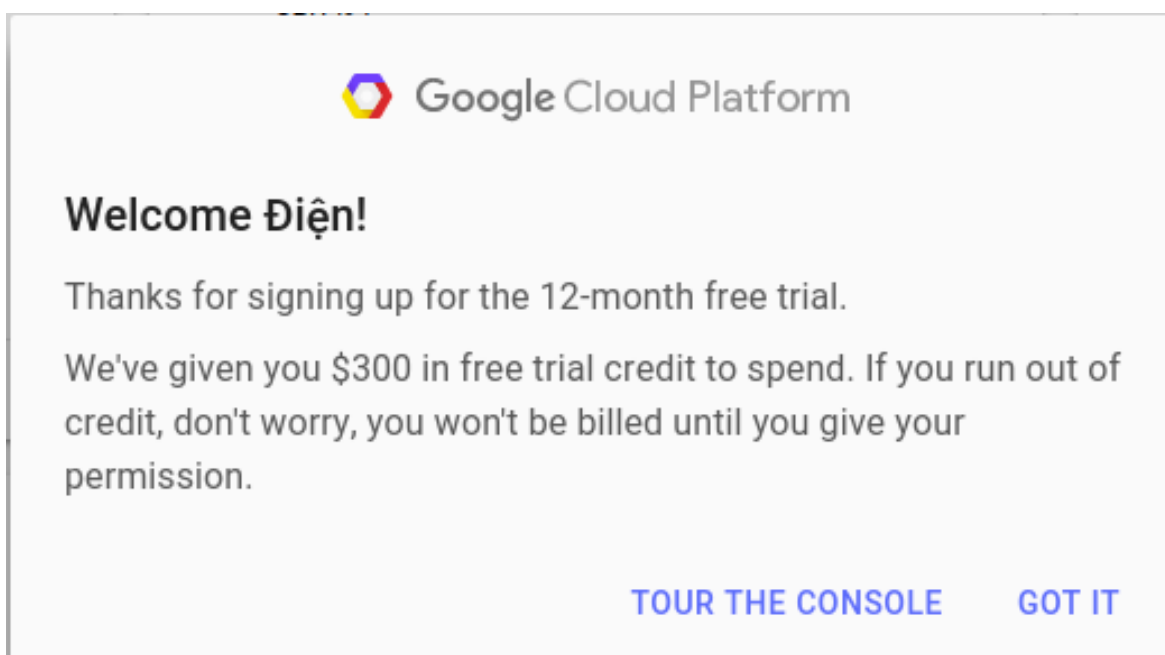
 **Account type**  

Individual

 **Name and address**  

Điện Dương Lữ
3/2 Street
Hung Loi
Ninh Kieu
Can Tho City 900000
Vietnam

Hình 1.1: Điền đầy đủ thông tin Google yêu cầu



Hình 1.2: Thông báo của Google khi đăng ký thành công

New Project

i You have 9 projects remaining in your quota. [Learn more.](#)

Project name ?

translation

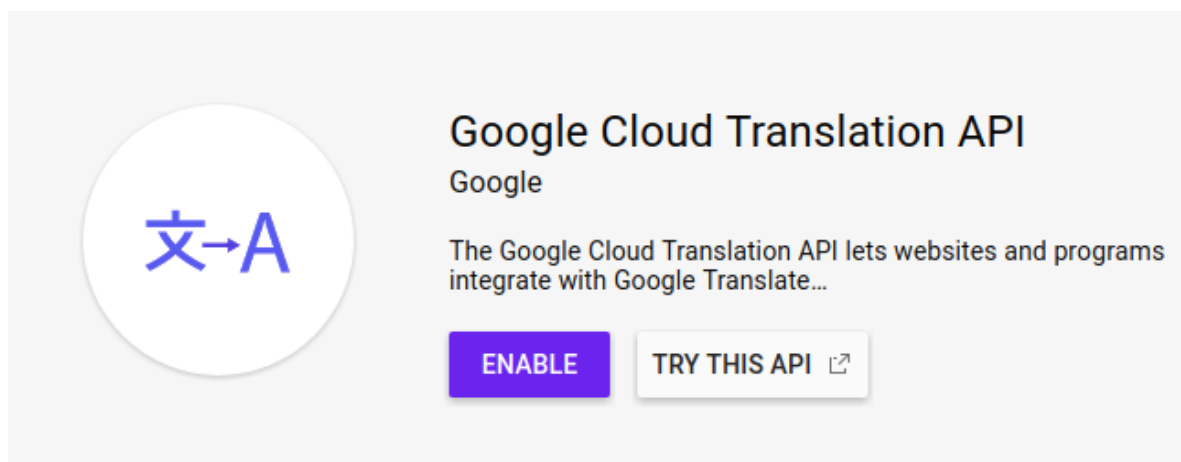
Your project ID will be translation-183411 ? [Edit](#)

Create **Cancel**

Hình 1.3: Tạo một project trên Google Cloud Platform

1.3 Kích hoạt Google Cloud Translation API

1. Từ thanh tìm kiếm, nhập từ khoá “Google Cloud Translation API”.
2. Nhấp vào nút “Enable” để kích hoạt dịch vụ Translation (Hình 1.4).



Hình 1.4: Kích hoạt dịch vụ Translation

1.4 API key

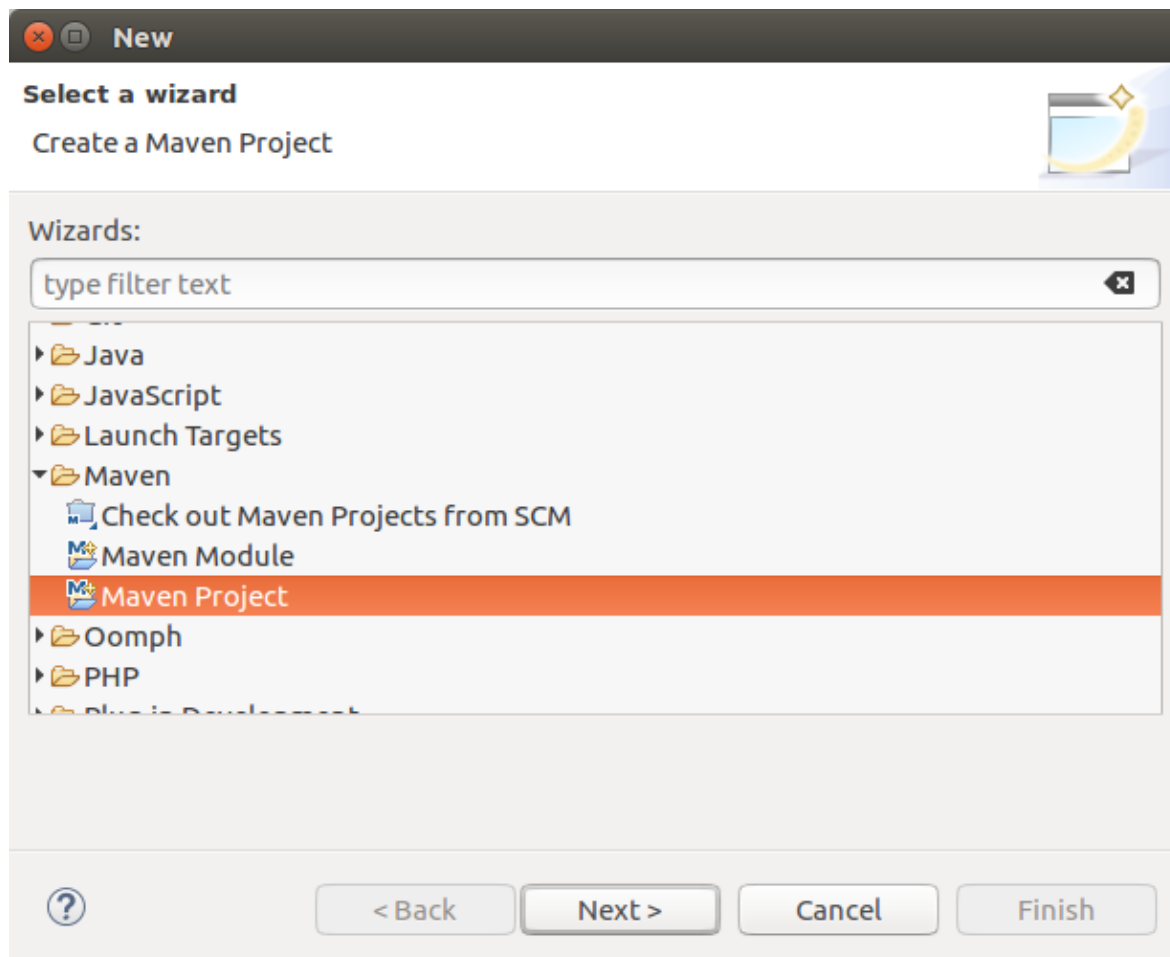
API key là một chuỗi được mã hoá nhằm giúp cho Google định danh được ứng dụng đang sử dụng dịch vụ, tính toán băng thông và tính phí dịch vụ. Để lấy API key cho ứng dụng, ta thực hiện các bước sau:

1. Từ Google Cloud Platform Console menu, vào APIs and services > Credentials.

2. Tạo một API key và tiến hành hạn chế quyền truy cập đến API key này.

1.5 Tạo một project trên Eclipse

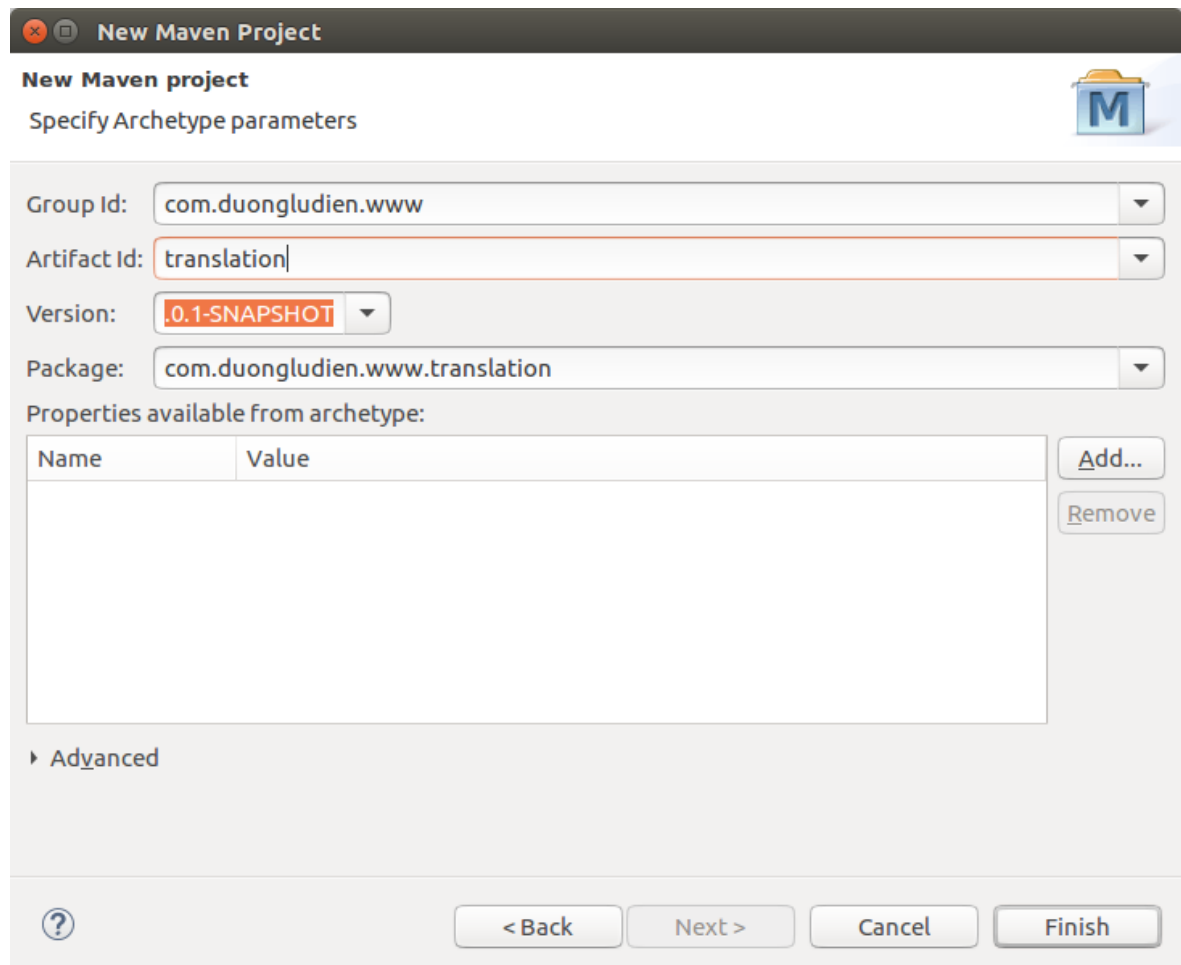
1. Cài đặt gói M2Eclipse.
2. Vào File > New > Other... > Maven > Maven Project (Hình 1.5 và 1.6).



Hình 1.5: Tạo một project Maven trên Eclipse

3. Mở file pom.xml và thêm gói Translation API vào mục Dependencies¹ (Hình 1.7).
 - Group Id: `com.google.apis`
 - Artifact Id: `google-api-services-translate`
 - Version: `v2-rev51-1.23.0`

¹<https://developers.google.com/api-client-library/java/apis/translate/v2>



New Maven Project

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

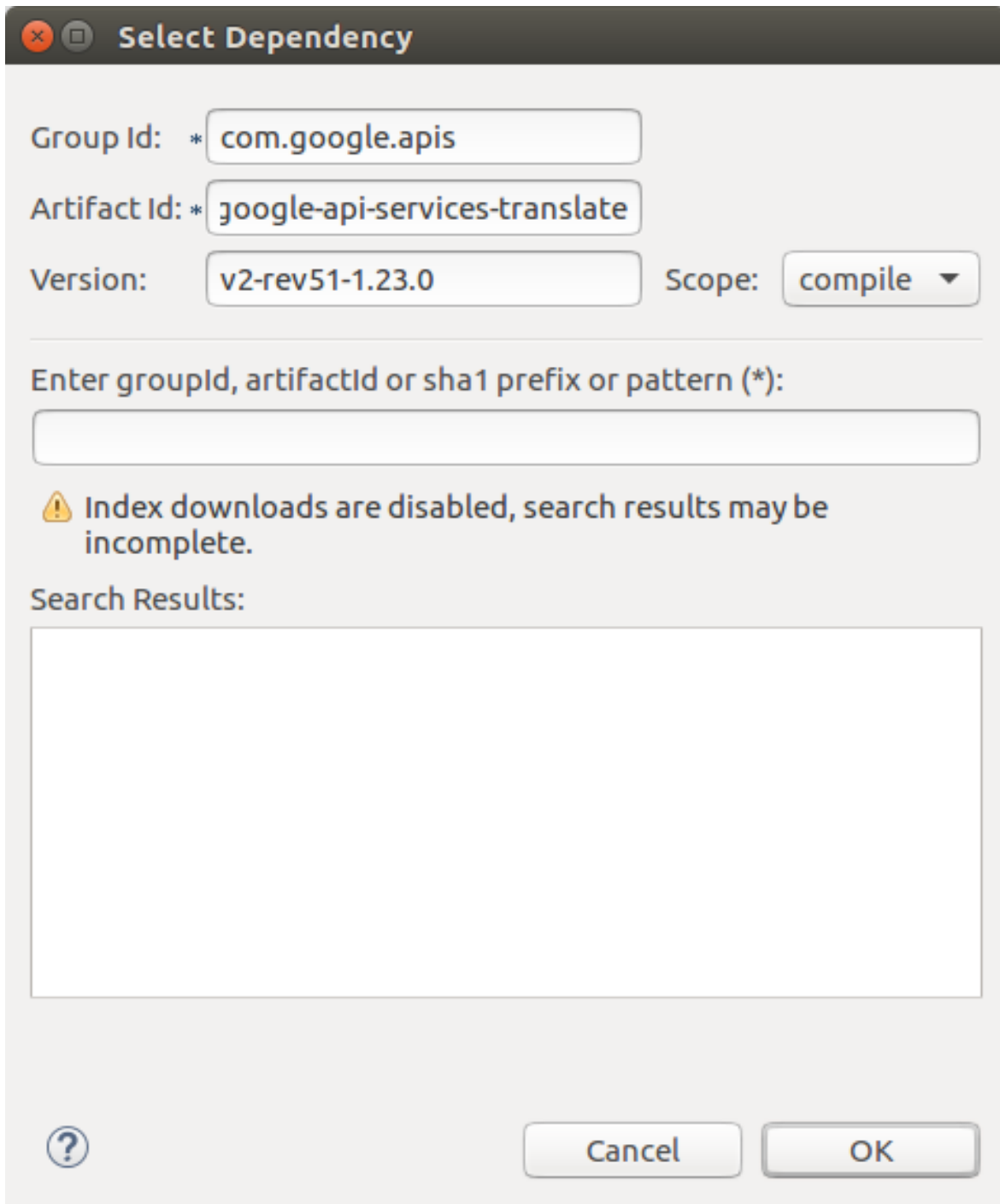
Package:

Properties available from archetype:

Name	Value
------	-------

Advanced

Hình 1.6: Thiết lập Archetype Parameters



Hình 1.7: Thêm Dependencies

Chương 2

DỊCH VĂN BẢN

Chương này sẽ mô tả cách sử dụng Cloud Translation API để dịch văn bản thông qua dịch vụ Translate.

2.1 HTTP request

Lệnh POST được định nghĩa như sau:

```
POST https://translation.googleapis.com/language/translate/v2?
    key=YOUR_API_KEY
```

Các tham số:

q	string <i>Bắt buộc.</i> Đoạn văn bản cần dịch. Lặp lại tham số này nếu có nhiều đoạn.
target	string <i>Bắt buộc.</i> Mã ISO-639-1 của ngôn ngữ mà ta muốn dịch sang.
format	string Định dạng của đoạn văn bản cần dịch. Có giá trị là html nếu là HTML và text nếu là plain-text.
source	string Mã ISO-639-1 của q . Nếu không cung cấp tham số này, API sẽ tự động phát hiện ngôn ngữ.
model	string Mô hình dịch. Nếu có giá trị là base thì API sẽ sử dụng mô hình Phrase-Based Machine Translation (PBMT), hoặc nmt cho mô hình Neural Machine Translation (NMT). Nếu bỏ qua tham số này, mô hình NMT sẽ được ưu tiên sử dụng. Mô hình NMT chỉ hỗ trợ khi ngôn ngữ nguồn hoặc đích là tiếng Anh (en).
key	string API key. Nếu sử dụng OAuth 2.0 thì không cần tham số này.

2.2 Kết quả trả về

Nếu thành công, kết quả trả về ở dạng JSON như sau:

```
{
  "data": {
    object(TranslateTextResponseList)
  },
}
```

Các trường dữ liệu:

data	object(TranslateTextResponseList) Danh sách các kết quả dịch cho mỗi giá trị của q.
------	--

2.3 TranslateTextResponseList

Mỗi phần tử trong danh sách kết quả là một mảng các thông tin liên quan đến giá trị q đã được dịch. Nó có dạng JSON như sau:

```
{
  "translations": [
    array
  ],
}
```

Các trường dữ liệu:

translations[]	array (TranslateTextResponseTranslation) Kết quả dịch.
----------------	---

2.4 Dịch văn bản

2.4.1 Dịch chuỗi văn bản

Sử dụng tham số q để định các chuỗi văn bản cần dịch. HTTP request sau dịch 2 chuỗi sang tiếng Phần Lan:

```
POST https://translation.googleapis.com/language/translate/v2
?key=YOUR_API_KEY
```

```
{
  'q': 'Hello world',
  'q': 'My name is Jeff',
  'target': 'fi'
}
```

Nếu thành công, HTTP trả về mã trạng thái là 200 OK thì kết quả nhận được là một chuỗi JSON có dạng như sau:

```
{
  "data": {
    "translations": [
      {
        "translatedText": "Hei maailma",
        "detectedSourceLanguage": "en"
      },
      {
        "translatedText": "Nimeni on Jeff",
        "detectedSourceLanguage": "en"
      }
    ]
  }
}
```

2.4.2 Sử dụng một mô hình cụ thể

Mặc định, mô hình NMT được ưu tiên sử dụng. Trong trường hợp không sử dụng được mô hình NMT, mô hình PBMT được sử dụng. Ta sẽ thử dịch cùng một đoạn văn bản sử dụng 2 mô hình này và so sánh kết quả.

Sử dụng mô hình NMT, HTTP request có các tham số sau:

```
{
  'q': 'Education is the process of facilitating learning,
    or the acquisition of knowledge, skills, values,
    beliefs, and habits. Educational methods include
    storytelling, discussion, teaching, training, and
    directed research',
  'target': 'vi',
  'model': 'nmt'
}
```

Bản dịch nhận được là: “Giáo dục là quá trình tạo điều kiện thuận lợi cho học tập, hoặc thu thập kiến thức, kỹ năng, giá trị, niềm tin và thói quen. Các phương pháp giáo dục bao gồm kể chuyện, thảo luận, giảng dạy, đào tạo và nghiên cứu trực tiếp”.

Sử dụng mô hình PBMT, HTTP request có các tham số sau:

```
{
  'q': 'Education is the process of facilitating learning,
    or the acquisition of knowledge, skills, values,
    beliefs, and habits. Educational methods include
    storytelling, discussion, teaching, training, and
    directed research',
  'target': 'vi',
  'model': 'base'
}
```

Bản dịch nhận được là: "Giáo dục là quá trình tạo điều kiện học tập, hoặc tiếp thu kiến thức, kỹ năng, giá trị, niềm tin, và thói quen. phương pháp giáo dục bao gồm kể

chuyện, thảo luận, giảng dạy, đào tạo và nghiên cứu đạo".

Có thể thấy rằng mô hình NMT cho kết quả dịch thuật chính xác hơn và mang ý nghĩa dịch thuật cao hơn.

2.5 Dịch vụ Translate trong Java

Để sử dụng dịch vụ Translate trong Java, ta thực hiện theo quy trình sau:

1. Tạo một đối tượng thuộc lớp `Translate.Builder`.
2. Sử dụng phương thức `build()` của đối tượng `Builder` vừa tạo để khởi tạo một dịch vụ `Translate`.
3. Tạo một yêu cầu (đối tượng thuộc lớp `Translations.List`) và sử dụng các setter để định các tham số cho lệnh `POST`.
4. Thực thi yêu cầu bằng phương thức `execute()`. Kết quả trả về ở dạng `JSON`, thuộc lớp `TranslationsListResponse`.
5. Dùng phương thức `getTranslations()` nhận về một `List` các đối tượng thuộc lớp `TranslationResource`.
6. Dùng các phương thức `getDetectedSourceLanguage()` và `getTranslatedText()` để nhận về các chuỗi đã dịch và ngôn ngữ được phát hiện (nếu tham số `source` chưa được xác định).

Chương 3

CÁC NGÔN NGỮ ĐƯỢC HỖ TRỢ

Như đã đề cập trong phần giới thiệu, Google Translate hỗ trợ trên 100 ngôn ngữ khác nhau. Để liệt kê các ngôn ngữ được hỗ trợ, ta sử dụng HTTP request gọi đến dịch vụ Supported Languages.

3.1 HTTP request

Lệnh GET được định nghĩa như sau:

```
GET https://translation.googleapis.com/language/translate/v2/  
languages?key=YOUR_API_KEY
```

Các tham số:

target	string Ngôn ngữ của kết quả mà ta muốn nhận được (Ví dụ: en). Nếu target được xác định thì tên của các ngôn ngữ sẽ được trả về trong trường name . Ngược lại, ta chỉ nhận được mã ISO-639-1 của chúng.
model	string Mô hình dịch. Nếu tham số này có giá trị là base thì kết quả trả về sẽ là các ngôn ngữ được hỗ trợ bởi mô hình Phrase-Based Machine Translation (PBMT), hoặc nmt thì kết quả trả về ngôn ngữ được hỗ trợ bởi mô hình Neural Machine Translation (NMT). Nếu bỏ qua tham số này, tất cả các ngôn ngữ được hỗ trợ sẽ được trả về trong kết quả. Mô hình NMT chỉ hỗ trợ khi ngôn ngữ nguồn hoặc đích là tiếng Anh (en).
key	string API key. Nếu sử dụng OAuth 2.0 thì không cần tham số này.

3.2 Kết quả trả về

Nếu thành công, kết quả trả về ở dạng JSON như sau:

```
{
  "data": {
    object(GetSupportedLanguagesResponseList)
  },
}
```

Các trường dữ liệu:

data	object(GetSupportedLanguagesResponseList) Danh sách các ngôn ngữ được hỗ trợ bởi Translation API.
------	--

3.3 GetSupportedLanguagesResponseList

Mỗi phần tử trong danh sách kết quả là một mảng các thông tin liên quan đến ngôn ngữ được hỗ trợ. Nó có dạng JSON như sau:

```
{
  "languages": [
    {
      object(GetSupportedLanguagesResponseLanguage)
    },
  ],
}
```

Các trường dữ liệu:

languages[]	array (GetSupportedLanguagesResponseLanguage) Tập các ngôn ngữ được hỗ trợ.
-------------	--

3.4 Các ngôn ngữ được hỗ trợ

3.4.1 Nhận về danh sách các ngôn ngữ được hỗ trợ

HTTP request sau dùng để nhận về mã ISO-639-1 của các ngôn ngữ được hỗ trợ:

```
GET https://translation.googleapis.com/language/translate/v2/
languages?key=YOUR_API_KEY
```

Nếu thành công, HTTP trả về mã trạng thái là 200 OK thì kết quả nhận được là một chuỗi JSON có dạng như sau:

```
{
  "data": {
    "languages": [
      {
        "language": "af"
      },
      ...
      {
        "language": "vi"
      }
    ]
  }
}
```

```
    },
    {
      "language": "zh"
    },
    {
      "language": "zh-TW"
    },
    {
      "language": "zu"
    }
  ]
}
```

3.4.2 Nhận về danh sách các ngôn ngữ được hỗ trợ dưới dạng một ngôn ngữ

HTTP request sau dùng để nhận về mã ISO-639-1 và tên của các ngôn ngữ được hỗ trợ bằng tiếng Anh:

```
GET https://translation.googleapis.com/language/translate/v2/
languages?key=YOUR_API_KEY
```

```
{
  "target": "en"
}
```

Nếu thành công, HTTP trả về mã trạng thái là 200 OK thì kết quả nhận được là một chuỗi JSON có dạng như sau:

```
{
  "data": {
    "languages": [
      {
        "language": "af",
        "name": "Afrikaans"
      },
      ...
      {
        "language": "vi",
        "name": "Vietnamese"
      },
      {
        "language": "cy",
        "name": "Welsh"
      },
      {
        "language": "xh",
        "name": "Xhosa"
      }
    ]
  }
}
```

```
    },
    {
      "language": "yi",
      "name": "Yiddish"
    },
    {
      "language": "yo",
      "name": "Yoruba"
    },
    {
      "language": "zu",
      "name": "Zulu"
    }
  ]
}
```

3.5 Dịch vụ Supported Languages trong Java

Để sử dụng dịch vụ Supported Languages trong Java, ta thực hiện theo quy trình sau:

1. Tạo một đối tượng thuộc lớp `Translate.Builder`.
2. Sử dụng phương thức `build()` của đối tượng `Builder` vừa tạo để khởi tạo một dịch vụ `Translate`.
3. Tạo một yêu cầu (đối tượng thuộc lớp `Languages.List`) và sử dụng các setter để định các tham số cho lệnh `GET`.
4. Thực thi yêu cầu bằng phương thức `execute()`. Kết quả trả về ở dạng JSON, thuộc lớp `LanguagesListResponse`.
5. Dùng phương thức `getLanguages()` nhận về một `List` các đối tượng thuộc lớp `LanguagesResource`.
6. Dùng các phương thức `getLanguage()` để nhận về mã ISO-639-1 và phương thức `getName()` để nhận tên của các ngôn ngữ (nếu có).

Chương 4

NHẬN DIỆN NGÔN NGỮ

Một trong những tính năng ưu việt của Google Translate là phát hiện ngôn ngữ. Tính năng này được bao gồm trong tính năng dịch, tuy nhiên Google cũng phát triển nó thành một dịch vụ riêng gọi là Detecting Language để ta có thể phát hiện ngôn ngữ mà không cần phải dịch.

4.1 HTTP request

Lệnh POST có dạng như sau:

```
POST https://translation.googleapis.com/language/translate/v2/
    detect
```

Các tham số:

q	string <i>Bắt buộc.</i> Đoạn văn bản cần phát hiện ngôn ngữ. Lặp lại tham số này nếu có nhiều đoạn.
key	string API key. Nếu sử dụng OAuth 2.0 thì không cần tham số này.

4.2 Kết quả trả về

Nếu thành công, kết quả trả về ở dạng JSON như sau:

```
{
  "data": {
    object(DetectLanguageResponseList)
  },
}
```

Các trường dữ liệu:

data	object (ListValue) Danh sách các kết quả phát hiện ngôn ngữ. Danh sách này sẽ bao gồm kết quả phát hiện ngôn ngữ cho mỗi giá trị của q trong HTTP request.
-------------	---

4.3 DetectLanguageResponseList

Mỗi phần tử trong danh sách kết quả là một mảng các thông tin liên quan đến ngôn ngữ được phát hiện. Nó có dạng JSON như sau:

```
{
  "detections": [
    array
  ],
}
```

Các trường dữ liệu:

detections[]	<p>array (ListValue format)</p> <p>Kết quả phát hiện ngôn ngữ cho mỗi giá trị của q. Mỗi ListValue gồm các trường sau:</p> <ul style="list-style-type: none">• language (string) - Ngôn ngữ được phát hiện.• isReliable (boolean) - Kết quả phát hiện ngôn ngữ có đáng tin hay không.• confidence (float) - Độ tin cậy.
--------------	---

4.4 Nhận diện ngôn ngữ

4.4.1 Nhận diện ngôn ngữ trên một chuỗi

HTTP request sau dùng để nhận diện ngôn ngữ của chuỗi "He is a student":

```
POST https://translation.googleapis.com/language/translate/v2/
detect?key=YOUR_API_KEY
```

```
{
  'q': 'He is a student',
}
```

Nếu thành công, HTTP trả về mã trạng thái là 200 OK thì kết quả nhận được là một chuỗi JSON có dạng như sau:

```
{
  "data": {
    "detections": [
      {
        "confidence": 1,
        "isReliable": false,
        "language": "en"
      }
    ]
  }
}
```

```
}
    }
}
```

4.4.2 Nhận diện ngôn ngữ trên nhiều chuỗi

Để nhận diện ngôn ngữ trên nhiều chuỗi, ta sử dụng tham số **q** để định nghĩa cho mỗi chuỗi. Ví dụ này sử dụng 2 chuỗi:

```
POST https://translation.googleapis.com/language/translate/v2/
detect?key=YOUR_API_KEY
```

```
{
  'q': 'Hello world',
  'q': 'Hei maailma'
}
```

Nếu thành công, HTTP trả về mã trạng thái là 200 OK thì kết quả nhận được là một chuỗi JSON có dạng như sau:

```
{
  "data": {
    "detections": [
      [
        {
          "confidence": 1,
          "isReliable": false,
          "language": "en"
        }
      ],
      [
        {
          "confidence": 1,
          "isReliable": false,
          "language": "fi"
        }
      ]
    ]
  }
}
```

4.5 Dịch vụ Detecting Language trong Java

Để sử dụng dịch vụ Detecting Language trong Java, ta thực hiện theo quy trình sau:

1. Tạo một đối tượng thuộc lớp `Translate.Builder`.
2. Sử dụng phương thức `build()` của đối tượng `Builder` vừa tạo để khởi tạo một dịch vụ `Translate`.

3. Tạo một yêu cầu (đối tượng thuộc lớp `Detections.List`) và sử dụng các setter để định các tham số cho lệnh `POST`.
4. Thực thi yêu cầu bằng phương thức `execute()`. Kết quả trả về ở dạng JSON, thuộc lớp `DetectionsListResponse`.
5. Dùng phương thức `getDetections()` nhận về một `List` mà mỗi phần tử của nó là một `List` các đối tượng thuộc lớp `DetectionsResourceItems`.
6. Chạy 2 vòng lặp `for` và dùng phương thức `getLanguage()` để nhận về mã ISO-639-1 của các ngôn ngữ được phát hiện.

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Chương trình tích hợp đầy đủ các tính năng chính của Translation API, bao gồm: dịch văn bản, liệt kê các ngôn ngữ được hỗ trợ, phát hiện ngôn ngữ. Tuy nhiên, chương trình vẫn chưa khai thác hết các tùy chọn trong từng tính năng như định nghĩa mô hình sẽ sử dụng hay dịch văn bản định dạng HTML.

Một hạn chế khác là chương trình sử dụng phương thức xác minh bằng API key nên có độ bảo mật thấp, dễ lộ API key.

5.2 Hướng phát triển

Kết hợp Translation API và Speech API để phát triển một thiết bị có thể lắng nghe và dịch nhiều ngôn ngữ khác nhau.

THAM KHẢO

1. Documentation - Translating Text: <https://goo.gl/Qbziwx>
2. REST Reference - translate: <https://goo.gl/uRtqmi>
3. Documentation - Discovering Supported Languages: <https://goo.gl/2N7Dum>
4. REST Reference - languages: <https://goo.gl/EipHGM>
5. Documentation - Detecting Language: <https://goo.gl/snRKzJ>
6. REST Reference - detect: <https://goo.gl/pFRQsD>
7. Google Cloud Translation API v2 Java Docs: <https://goo.gl/bnU3ES>