

| 카카오톡 봇 강좌 | >

[고급] 파파고 번역 api를 이용한 번역 기능 만들기 (내장 메소드 미사용)



Dark Tornado 카페스텝 1:1 채팅

2020.06.28. 11:23 조회 291

댓글 5 URL 복사

강좌할 주제

=> `Api.papagoTranslate()`: 메소드를 사용하지 않고 파파고 번역 기능 구현

강좌 내용

=> ~~갯갬~~ 하우스

이 게시글에서는 `Api.papagoTranslate()`: 메소드를 사용하지 않고 파파고 번역 기능을 구현하는 것을 다룹니다.

이 게시글에서는 `Jsoup` 라이브러리를 사용합니다.

따라서, 이 게시글에서 다루는 소스(?)는 채팅 자동응답 봇(초록봇), 메신저봇, 메신저봇R, Nusty 내장 챗봇과 같이 Jsoup 라이브러리가 포함된 봇 구동 앱에서만 작동합니다.

~~원말인지 잘 모르겠다면, 메봇R의 경우 그냥 `Api.papagoTranslate()`를 사용하시면 되고,~~

~~채팅 자동응답 봇의 경우 3.0이 출시되기를 기다렸다가 `Api.papagoTranslate()`를 사용하시면 됩니다.~~

메신저봇R에 있는 파파고 번역 기능은 아마도 정식으로 ~~돌아서(?)~~ 만든게 아니라 api 키를 발급받아서 만든거일겁니다.

~~초록봇 3.0에서 `Api.papagoTranslate()` 추가 예정~~

파파고 api 사용시 한 번에 최대 5000자까지 번역이 가능하며, 하루에 10000자까지만 번역이 가능합니다.

모든 메신저봇R은 모두 같은 키를 사용하기 때문에, 모든 메봇R에서 `Api.papagoTranslate()`: 함수로 번역된 글자 수가 10000자를 넘어가면 그날은 모든 메봇R에서 파파고 번역 함수가 작동하지 않을겁니다.

실제로, 6월 13일에 모든 메봇R의 `Api.papagoTranslate()`:가 죽었던 적이 있습니다(?).

```
org.mozilla.javascript.EvaluatorException:
{count/quota=10003/10000} : Query limit
exceeded. (쿼리 한도를 초과했습니다.)
(번역#12)
```

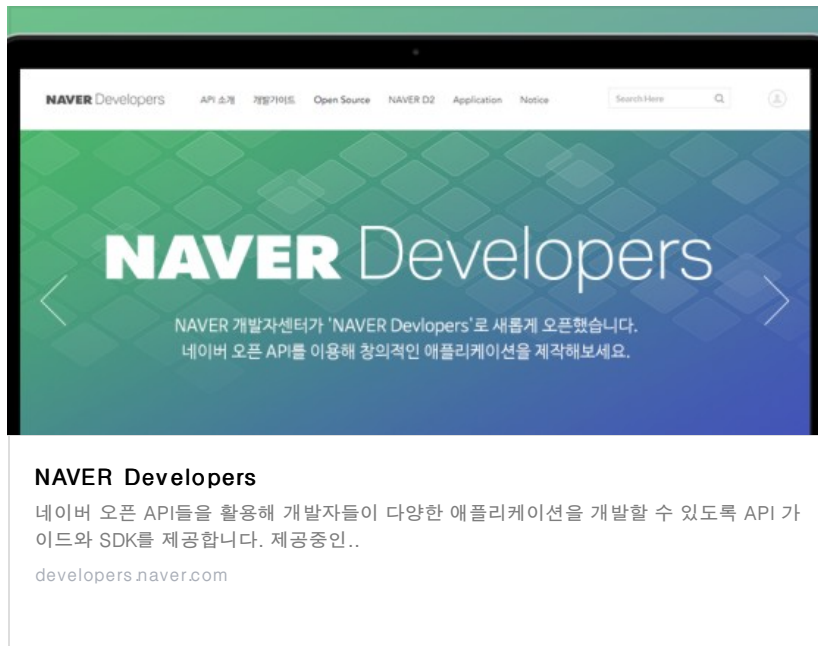
Message...



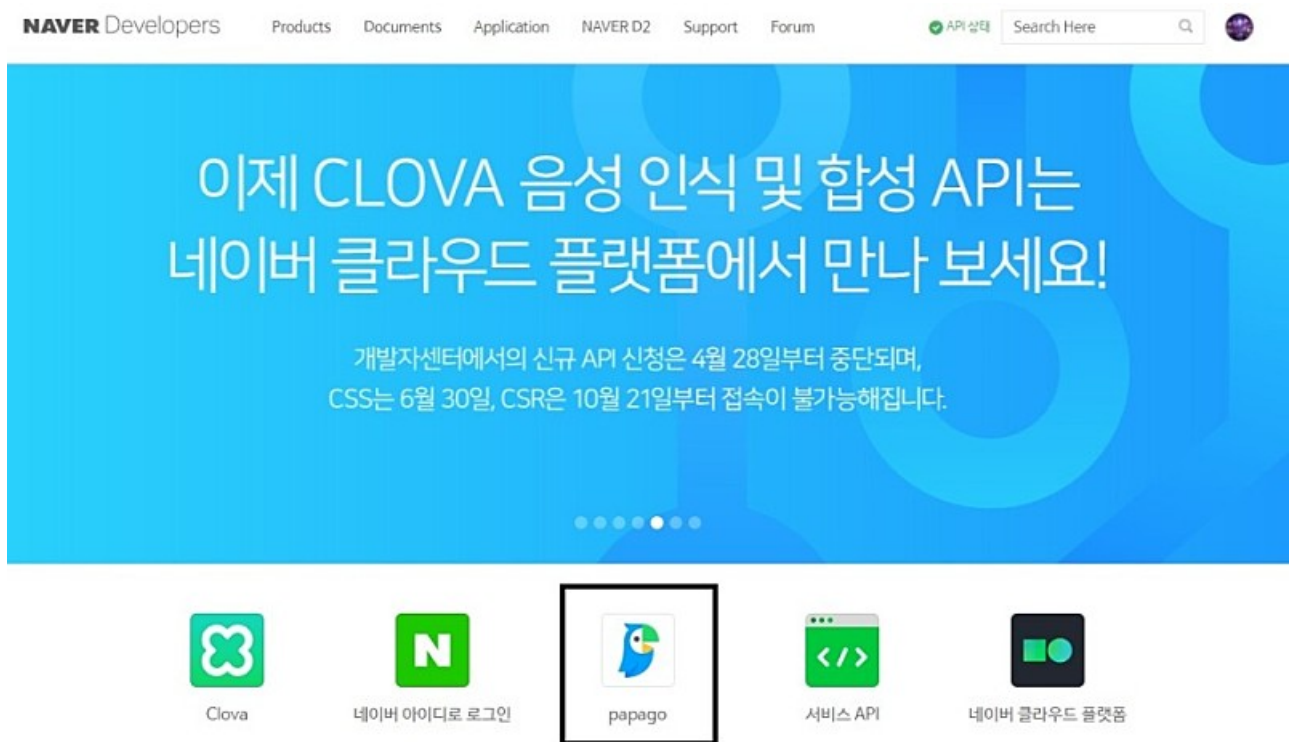
그런고로, api 키를 발급받아서 혼자서만 써봅시다.

일단, 네이버 개발자 센터에 들어갑니다.

<https://developers.naver.com/main/>



그러면 나오는 화면에서 papago를 눌러줍니다.



본인인증 하라고 나오면 본인인증을 하시면 됩니다.

그리고 이 화면으로 넘어온다면,



API 이용 안내

Clova

네이버 아이디로 로그인

파파고

Papago 번역

언어 감지

한글인명-로마자 변환

서비스 API

Neural Machine Translation

N 트윗 공유하기 18기

NMT는 Neural Machine Translation(인공신경망 기반 기계번역)의 약어입니다. 파파고의 NMT 기술은 입력 문장을 문장벡터로 변환하는 신경망(encoder)과 문장벡터에서 번역하는 언어의 문장을 생성하는 신경망(decoder)를 대규모의 병렬 코퍼스부터 자동으로 학습합니다. 입력문장의 일부가 아니라 문장 전체 정보를 바탕으로 번역을 수행하기때문에 기존 SMT방식의 번역보다 더욱 정확하고 문장 맥락에 맞는 번역을 하는것이 특징입니다.

풍부한 언어 처리 경험

다국어 언어 처리에 대한 네이버의 기술과 경험을 번역 엔진에 적용해 보다 정확한 번역 결과를 제공합니다. 네이버 서비스를 통해 축적해온 번역 품질에 대한 피드백들을 적극적으로 반영해 단순 통역에만 의존하는 기존 기술의 한계를 보완했습니다.



다양한 학습 데이터 구축

자연스러운 번역을 가능하게 하는 대규모의 학습 데이터를 보유하고 있습니다. 언어의 규칙을 효과적으로 파악하는 핵심 번역 기술을 강화하기 위해 품질 높은 학습 모델을 사용합니다.

"오픈 API 신청"이라고 쓰여있는 초록색 버튼을 누르시면 됩니다.

손쉬운 사용

웹 기반의 콘솔을 통해 쉽고 편리한 사용이 가능합니다. RESTful 형태로 지원되는 API를 통해 고객의 서비스에 번역 기능을 간단하게 적용할 수 있습니다.

* 한국어(ko)-영어(en), 한국어(ko)-일본어(ja), 한국어(ko)-중국어 간체(zh-CN), 한국어(ko)-중국어 번체(zh-TW), 한국어(ko)-스페인어(es), 한국어(ko)-프랑스어(fr), 한국어(ko)-러시아어(ru), 한국어(ko)-베트남어(vi), 한국어(ko)-태국어(th), 한국어(ko)-인도네시아어(id), 한국어(ko)-독일어(de), 한국어(ko)-이탈리아어(it), 중국어 간체(zh-CN) - 중국어 번체(zh-TW), 중국어 간체(zh-CN) - 일본어(ja), 중국어 번체(zh-TW) - 일본어(ja), 영어(en)-일본어(ja), 영어(en)-중국어 간체(zh-CN), 영어(en)-중국어 번체(zh-TW), 영어(en)-프랑스어(fr)를 지원합니다.

* 처리한도 : 10,000글자/일

오픈 API 이용 신청

개발 가이드 보기

애플리케이션 이름은 능력껏 입력하시면 되고,

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

하와와



애플리케이션 이름	<ul style="list-style-type: none"> 네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요. 40자 이내의 영문, 한글, 숫자, 공백문자, "-", "_" 만 입력 가능합니다.
사용 API	<div>선택하세요. ▼</div> <div>Papago 번역 ×</div>
비로그인 오픈 API 서비스 환경	<div>환경 추가 ▼ ①</div>

- [비로그인 오픈 API 서비스 환경] 설정을 확인해 주세요.

등록하기

취소

서비스 환경은 셋 중 하나 이상을 능력껏 적으시면 됩니다. 전부 다 해야하는게 아니에요.
필자는 실제로 해당 기능을 넣을 안드로이드 앱의 패키지명으로 했습니다.

사용 API	Papago 번역 ×
비로그인 오픈 API 서비스 환경	<div>WEB 설정 × ^</div> <div> <div>웹 서비스 URL (최대 10개)</div> <div> <input type="text" value="http://도메인주소, file:///로컬URI"/> <div>— ①</div> </div> <ul style="list-style-type: none"> 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다. http와 https는 구분하지 않습니다. www는 빼고 입력해 주세요. 예) http://naver.com 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com) 하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file:///로컬URI) </div>
	<div>Android 설정 × ^</div> <div> <div>안드로이드 앱 패키지 이름</div> <div> <input type="text"/> <div>①</div> </div> <div>안드로이드 앱 패키지 이름을 입력하세요. 예) com.example.mynavermap</div> </div>
	<div>iOS 설정 × ^</div> <div> <div>iOS Bundle ID</div> <div> <input type="text"/> <div>①</div> </div> <div>iOS앱의 Bundle ID를 입력하세요. 예) com.naver.example.MyNaverMap</div> </div>

그러면 이런 화면이 나옵니다.

저기에 있는 Client ID와 Client Secret를 잘 기억해두시면 됩니다. 유출되지 않도록 주의해주세요.
Client Secret의 경우, 보기 버튼을 누르면 나옵니다.

개요

API 설정

멤버관리

로그인 통계

API 통계

Playground(Beta)

애플리케이션 정보

Client ID	<div></div>
Client Secret	<div>...</div> <div>보기</div>

파파고 NMT API 가이드

가이드에 따라, cURL을 이용해 동작을 확인해 보세요.

```
curl "https://openapi.naver.com/v1/papago/n2mt" \
-H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" \
-H "X-Naver-Client-Id: " \
-H "X-Naver-Client-Secret: " \
-d "source=ko&target=en&text=만나서 반갑습니다." -v
```

개발 가이드(<https://developers.naver.com/docs/nmt/reference/>)에 나와있는 바와 같이,

<https://openapi.naver.com/v1/papago/n2mt> 라는 주소로 post요청을 날리시면 되는데,

2. API 기본 정보

API명	메서드	요청 URL	출력 포맷
N2MT	POST	https://openapi.naver.com/v1/papago/n2mt	JSON

🔗 📄 📄 📄 📄

파라미터로 source, target, text를 넘겨주면 됩니다.

text로 넘어온 내용을 source에 해당하는 언어라고 가정하고, target에 해당하는 언어로 번역

3. 요청 변수

요청 변수명	타입	필수 여부	설명
source	string	Y	번역 대상 언어 코드. 1.ko : 한국어 2.en : 영어 3.zh-CN : 중국어 간체 4.zh-TW : 중국어 번체 5.es : 스페인어 6.fr : 프랑스어 7.vi : 베트남어 8.th : 태국어 9.id : 인도네시아어 ko(-)en, ko(-)zh-CN, ko(-)zh-TW, ko(-)es, ko(-)fr, ko(-)vi, ko(-)th, ko(-)id, en(-)ja, en(-)fr 조합만 가능
target	string	Y	번역 결과 언어 코드. 1.ko : 한국어 2.en : 영어 3.zh-CN : 중국어 간체 4.zh-TW : 중국어 번체 5.es : 스페인어 6.fr : 프랑스어 7.vi : 베트남어 8.th : 태국어 9.id : 인도네시아어 ko(-)en, ko(-)zh-CN, ko(-)zh-TW, ko(-)es, ko(-)fr, ko(-)vi, ko(-)th, ko(-)id, en(-)ja, en(-)fr 조합만 가능
text	string	Y	번역할 text (1회 호출시 최대 5,000자까지 가능)

출력 결과는 이렇다고 합니다.

4. 출력 결과 [↗](#)

필드명	타입	설명
srcLangType	string	source language
tarLangType	string	target language
translatedText	string	번역된 text

일단 한 번 넘겨봅시다.

아까 위에서 잘 기억해두라고 했던 Client ID와 Client Secret는 다음과 같이 헤더에 넣으시면 됩니다.

This is test.라는 문자열을 영어라고 가정하고 한국어로 번역시켜볼겁니다.

```
1 function response(room, msg, sender, isGroupChat, replier) {
```



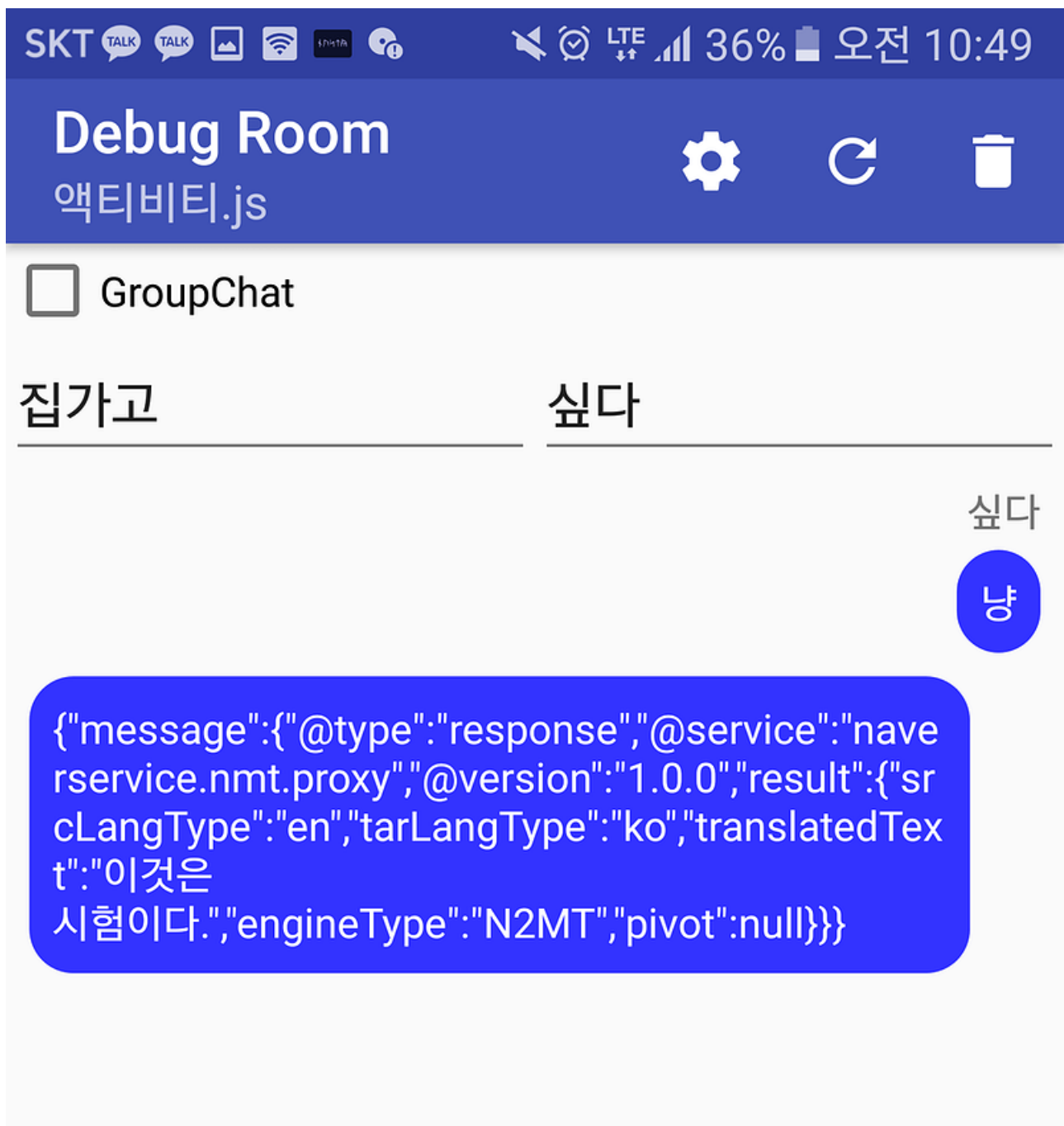
```

2 var data = org.jsoup.Jsoup.connect( https://openapi.naver.com/v1/
  papago/n2mt")
3 .header("X-Naver-Client-Id", "Client ID 입력")
4 .header("X-Naver-Client-Secret", "Client Secret 입력")
5 .data("source", "en")
6 .data("target", "ko")
7 .data("text", "This is test.")
8 .ignoreContentType(true).post().text();
9
10 replier.reply(data);
11
12 }

```

실행 결과

(Client ID와 Client Secret를 넣지 않으면 오류됩니다.)



Enter message

SEND

반환 결과가 다음과 같군요.

```
{
  "message": {
    "@type": "response",
    "@service": "haverservice.nmt.proxy",
    "@version": "1.0.0",
    "result": {
      "srcLangType": "en",
      "tarLangType": "ko",
      "translatedText": "이것은 시험이다.",
      "engineType": "N2MT",
      "pivot": null
    }
  }
}
```

그러면, `data["message"]["result"]["translatedText"]`로 하면 번역된 내용만 나오겠지요?

```
5 .data("source", "en")
6 .data("target", "ko")
7 .data("text", "This is test.")
8 .ignoreContentType(true).post().text();
9 data = JSON.parse(data);
10
11 replier.reply(data["message"]["result"]["translatedText"]);
12
13 ,
```


13 }
14

뽕. 잘 나옵니다.

SKT   

    29%  오전 10:56

Debug Room

액티비티.js

☐ GroupChat

집가고

싶다

싶다

냥

```
{"message":{"@type":"response","@service":"naver.service.nmt.proxy","@version":"1.0.0","result":{"srcLangType":"en","tarLangType":"ko","translatedText":"이것은 시험이다.","engineType":"N2MT","pivot":null}}}
```

싶다

냐앙

이것은 시험이다.

Enter message

SEND

Utils.papagoTranslate();라는 함수로 따로 분리시켜 보았습니다.

```
1  Utils.papagoTranslate = function(lang1, lang2, value){
2  try{
3  var data = org.jsoup.Jsoup.connect("https://openapi.naver.com/v1/
  papago/n2mt")
4  .header("X-Naver-Client-Id", "Client ID 입력")
5  .header("X-Naver-Client-Secret", "Client Secret 입력")
6  .data("source", lang1)
7  .data("target", lang2)
8  .data("text", value)
9  .ignoreContentType(true).post().text();
10 data = JSON.parse(data);
11 return data["message"]["result"]["translatedText"];
12 }
13 catch(e){
14 return null;
15 }
16 };
17
```

전체 소스입니다. 아마자아나, 알아서 일일이 쳐서 가져라가는 거시예오(?)

/번역 [언어코드1] [언어코드2] [내용]

해당 내용을 언어코드1에 해당하는 언어라고 가정하고 언어코드2로 번역합니다.

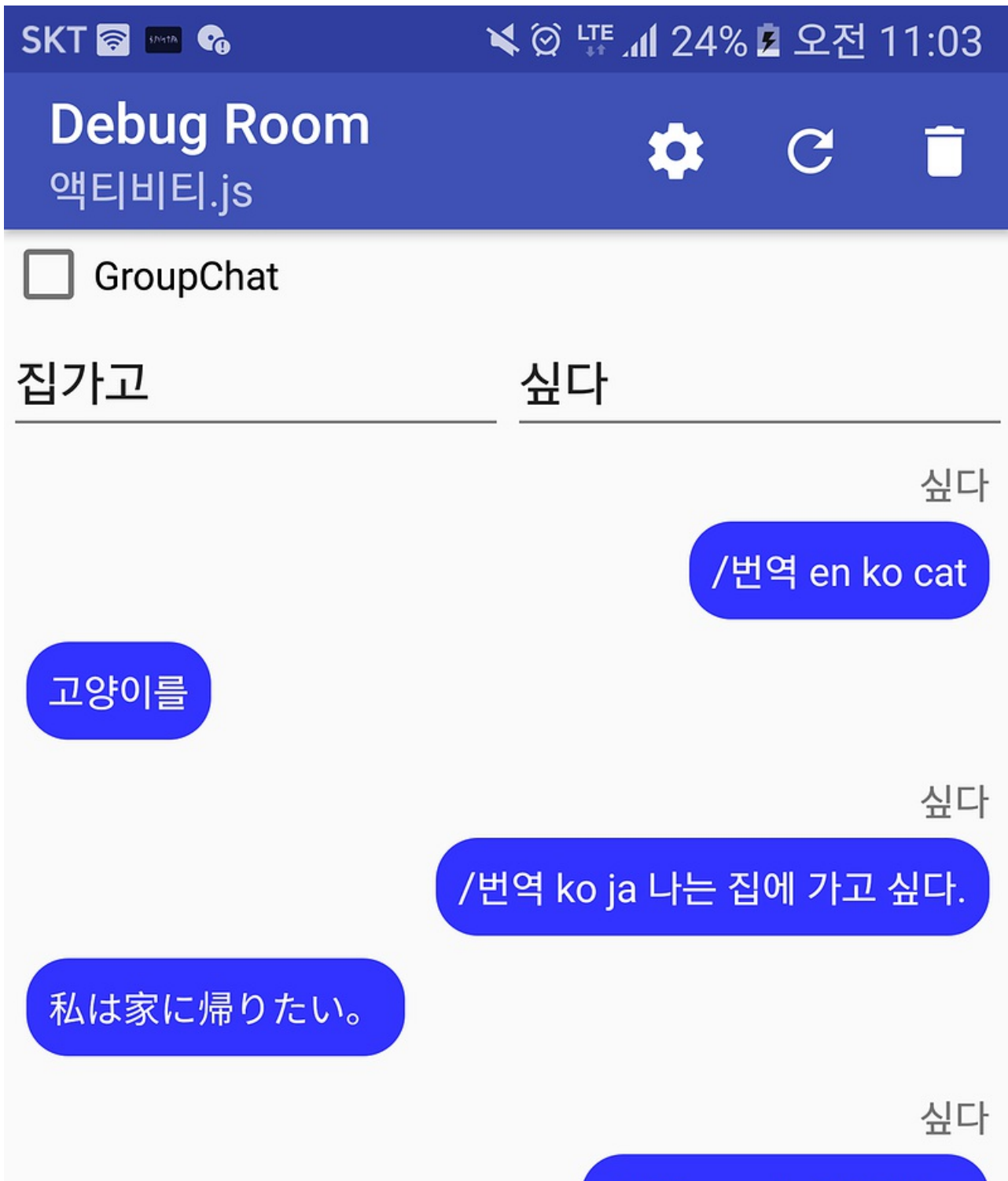
```
1  Utils.papagoTranslate = function(lang1, lang2, value) {
2      try {
3          var data = org.jsoup.Jsoup.connect("https://openapi.naver.com/v1/papago/n2mt")
4              .header("X-Naver-Client-Id", "Client ID 입력")
5              .header("X-Naver-Client-Secret", "Client Secret 입력")
6              .data("source", lang1)
7              .data("target", lang2)
8              .data("text", value)
9              .ignoreContentType(true).post().text();
10         data = JSON.parse(data);
11         return data["message"]["result"]["translatedText"];
12     } catch (e) {
13         return null;
14     }
15 };
16
```

```

17 function response(room, msg, sender, isGroupChat, replier) {
18     var cmd = msg.split(" ")[0];
19     var data = msg.replace(cmd + " ", "");
20     if (cmd == "/번역") {
21         var data2 = data.split(" ");
22         var lang1 = data2[0];
23         var lang2 = data2[1];
24         var value = data.replace(lang1 + " " + lang2 + " ", "");
25         var result = Utils.papagoTranslate(lang1, lang2, value);
26         if (result == null) replier.reply("번역 실패");
27         else replier.reply(result);
28     }
29 }
30

```

실행 결과 :



中国

Enter message

SEND

이제 다른 사람들과는 다른 api 키를 이용하여 파파고 번역을 사용하게 되었으니,
그날 내장 메소드로 번역시킨 내용이 10000자를 넘어가거 다른 파파고 번역 기능들이 죽어도 내겐 잘 작동할겁니다(?).



Dark Tornado님의 게시글 더보기 >

좋아요 6
 댓글 5

공유 | 신고

댓글 등록순 최신순

댓글알림



사로로

Api.papagoTranslate(id ,secret); 만드는데 어때신가요?
귀찮아서 그러는게 아니 읍읍

2020.06.28. 11:30 답글 쓰기



Dark Tornado 작성자

이미 초록봇 3.0에 Api.papagoTranslate();가 있는 거시네요.
id, secret도 넘길 수 있도록 구현 각.

2020.06.28. 11:33 답글 쓰기



윤동원

하와와

2020.06.28. 11:33 답글 쓰기



Dark Tornado 작성자

하와와

2020.06.28. 11:36 답글 쓰기



젤리파덕

메신저봇에서 쓸려고하면 안드로이드 패키지명을 뭐라고써야하나여???

2020.07.05. 02:19 답글 쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'카카오톡 봇 강좌' 게시판 글

전체 [고급] 말머리 글

이 게시판 새글 구독하기 ☐

[응용] 반복문 응용 - 원주율 값 구하기 🤖 [15]	도미 doami2005	2020.06.29.
[중급] 함수지향과 객체지향(5) - arguments를 알아보자 [5]	choi0108	2020.06.29.
[고급] 파파고 번역 api를 이용한 번역 기능 만들기 (내장 메소드 미사용) 🤖 📄 [5]	Dark Tornado	2020.06.28.
[기타] 안드로이드 우분투 설치, arm64에 노드 설치	핀밀크	2020.06.26.
[중급] 함수지향과 객체지향(4) - 익명함수란 무엇인가 [4]	choi0108	2020.06.26.

1 2 3

전체보기

이 카페 인기글



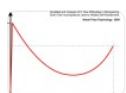
카카오톡이 이런걸 추천하다니

카톡 청지에 관해서

리스트
♡0 💬15

기본 개념에 대한 강좌 | 조건문과 반복문

OtakoidTony
♡0 💬6



카톡봇 사람들은 꼭봐야됨

버튼 누르면 intent가 함수를 실행하게하려
면

jomin398
♡0 💬18



메신저봇 글씨 색

1 2 3 4 5

원주율 100자리 가져오기

T봇
♡0 💬6

보호조치로 인해 오픈채팅
제한되었습니다.

자세히 보기

유허참여 하고싶지만 이용자보호조치 때문
에 참여가 제한
카봇커 채팅방을 그룹으로 만들면 오실분?

전국봇
성현
♡0 💬13
♡1 💬52