



The bridge to possible

Webex Teams with API

Chiwon Choi
Cisco Systems

Webex?

기업/학교에서 협업에 필요한 모든 기능을 하나의 Webex App에서 안전하게 제공

Call

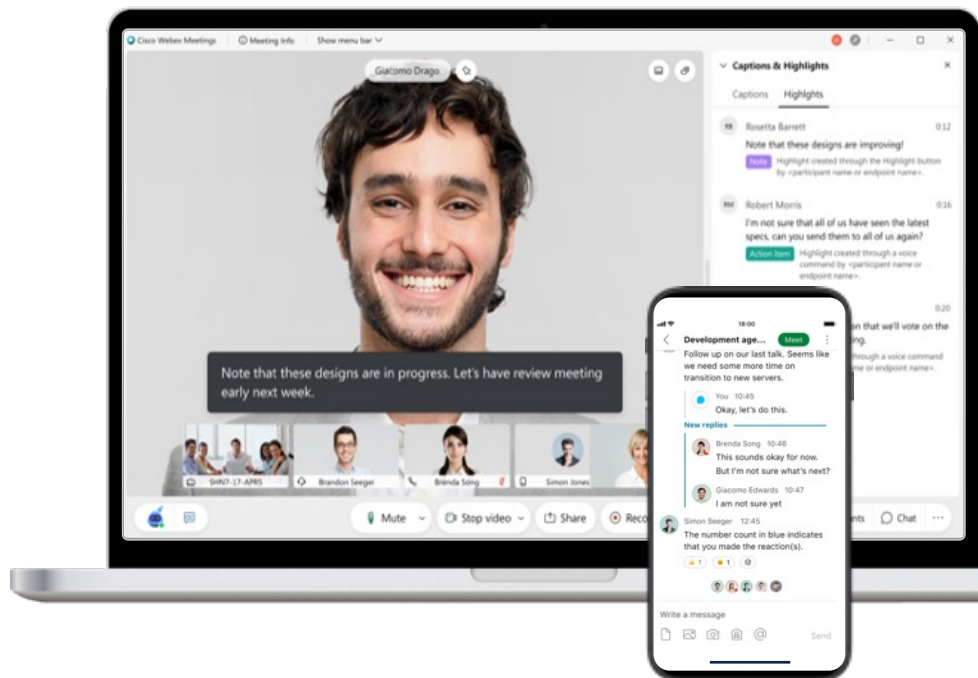
고품질의 인터넷 기반 비디오,보이스 전화 서비스 제공

Message

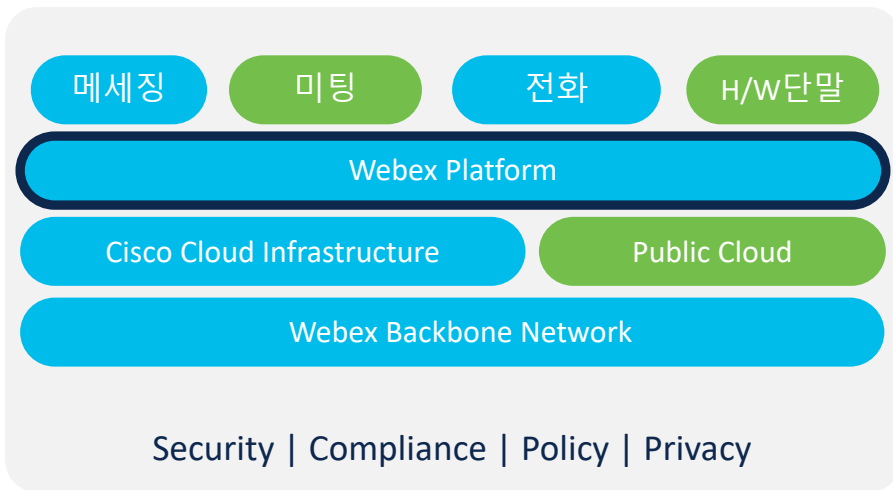
그룹/다이렉트 메시지, 파일등을 기업 외부인과의 손쉽고 빠르게 공유.
보다 빠른 협업 환경 제공

Meet

고품질 다자간 화상회의를 넘어
일정관리, AI 기반의 소음제거, 자동
번역 및 안면인식 기능 제공



The Webex Platform



Modernized
Application
Methodologies



Realtime
Analytics



Webex API/SDK
Developer
Community



AI/ML
+ Facial
Recognition

Meetings

Team Tools

미팅



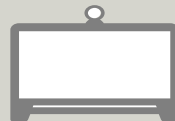
메세징

화이트 보드

파일 공유

3rd party 연동

전화



Webex API / SDK

Webex app: 연동 방법

#1

Native Integrations

Webex app에서 이미 제공하는
기능들을 손쉽게 가져다 활용

실제 사용자들이

#2

Integration Services

다른 app과 연동을 시켜주는
서비스를 통해 좀더 Webex를
강력하고/효과적으로 활용

IT팀 / Power 사용자

#3

Teams for Developers

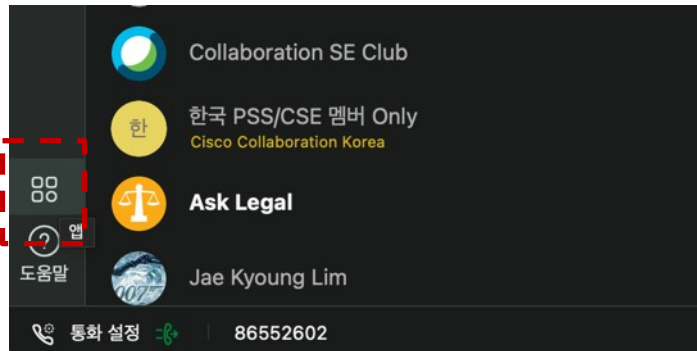
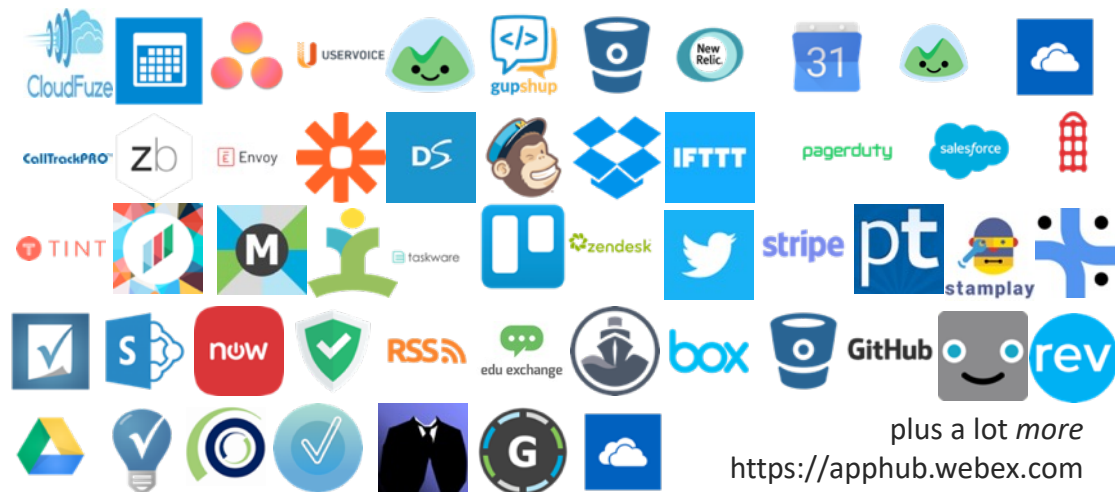
Cisco Webex app API를 통해 직접
내가 원하는 대로 개발하여
기업/학교의 서비스와 연동

developer.webex.com

IT팀 / 개발자

#1 .. Native Integrations: Webex App Hub

App Hub에 올라온 앱 중 나에게
필요한 서비스를 선택해서 연동
추가 개발(X)
커스터마이징(X)



#2 .. Integration & Bot Platforms



만약, 새로운 소식(RSS)이 블로그에 올라오면,

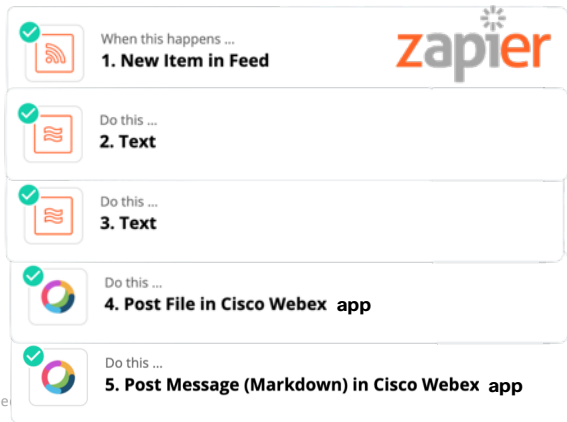
이미지를 추출하고

텍스트도 추출해서

Webex에 이미지를 업로드 하고

Webex에 텍스트도 업로드 해줘

© 2021 Cisco and/or its affiliates. All rights reserved.

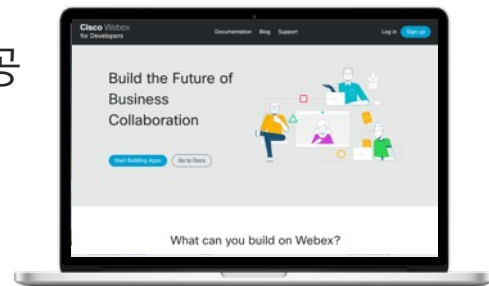


연동 플랫폼을 이용해서
내가 사용중인 서비스와 웹엑스를 연계
조건에 맞추어 필요한 행동을 취하도록
설정 가능

개발(X)
커스터마이징(일부 가능)

#3 Webex API's

1. **REST/XML API** – 미팅/메세지/전화/사용자 컨트롤 등
2. **Buttons & Cards** – 메세지에서 UI강화
3. **SDK** – iOS, Android, Browser, Node.JS, Java를 활용한 개발툴 제공
4. **Widget** – 웹 브라우저에 손쉽게 미팅/메세지 기능 이식
5. **Bots** – User interaction
6. **Integrations** – 사용자 '대신' 업무를 처리
7. **Guest Issuer** – for non-Webex Teams users
8. **Room Device API** – Cisco Device 컨트롤



#1 ... REST APIs


/People	/Rooms	/Membership	/Messages	/Webhooks	/Teams
<div>GET</div> List	<div>GET</div> List	<div>GET</div> List	<div>GET</div> List	<div>GET</div> List	<div>GET</div> List
<div>GET</div> Get details	<div>POST</div> Create	<div>POST</div> Create Membership	<div>POST</div> Create	<div>POST</div> Create	<div>POST</div> Create
<div>GET</div> Get my details	<div>GET</div> Get Room Details	<div>GET</div> Get details	<div>GET</div> Get details	<div>GET</div> Get details	<div>GET</div> Get details
	<div>PUT</div>	<div>PUT</div>		<div>PUT</div>	<div>PUT</div>
	<div>DELETE</div>	<div>DELETE</div>	<div>DELETE</div>	<div>DELETE</div>	<div>DELETE</div>

<https://developer.webex.com/quick-reference.html>

+ 관리자/전화/비디오 등 다양한 기능을 API로 제어 가능


#2 - Buttons & Cards

- 카드 형식의 메시지를 통한
시인성/사용자 경험 확보
- 하나의 앱에서 손쉽게 시스템 활용

**iProcurement Request**
Dirk-Jan Uittenbogaard
requested: Mon, June 14, 2019

Requested the purchase of:

Product: Logitech Wireless Presenter R400
Amount: 1
Price: € 39
Remark: Need presenter device every day
Approver comment:
DJ is presenting every day





Approve

Decline

Details

예시 - 승인/결재

 Collaboration API & Integration survey
duittenb@cisco.com



What is your role?

☐ Technical

☐ Commercial

☐ Leader

☐ Other

Who could use help with API's & integrations?

☐ me

☐ customer

☐ partner

☐ other

☐ don't need help

How could we help?

☐ Session or Training

☐ Workshop

☐ Hackathon

other

예시 - Survey

#3 - SDK

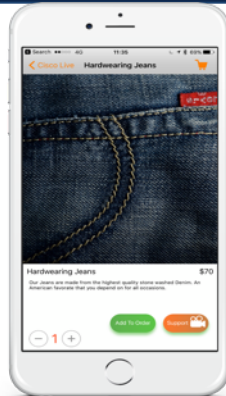
- Webex app의 기능을 별도 Application에 Embedded 하여 활용



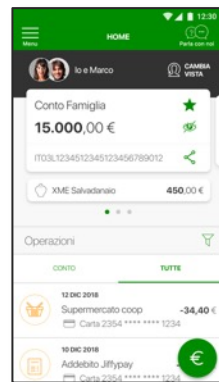
Healthcare



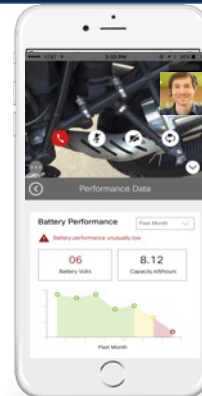
Retail



Finance



Tech support



SDK Support



iOS SDK

- iPhone & iPad
- iOS 11 +
- Swift 5+
- XCode 10+



Browser WebRTC JS SDK

- Firefox for desktop - latest
- Chrome for desktop – latest
- Edge for desktop - latest
- Safari for desktop - latest
- Safari for iOS
- Chrome for Android



Android SDK

- Android 5.0+
- Android Studio 2.3+
- Android SDK Tools 25+
- Android API Level 21+
- Java JDK 8+

#4 - Widgets

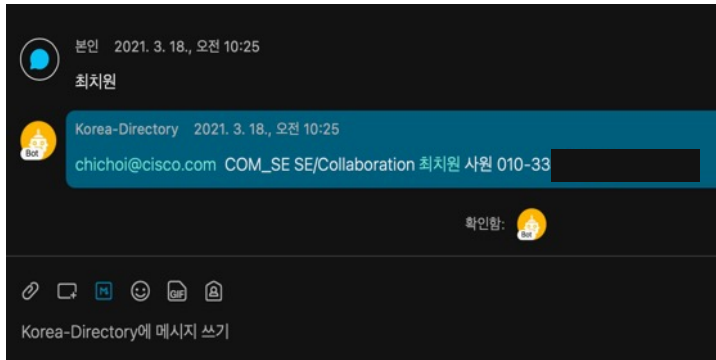
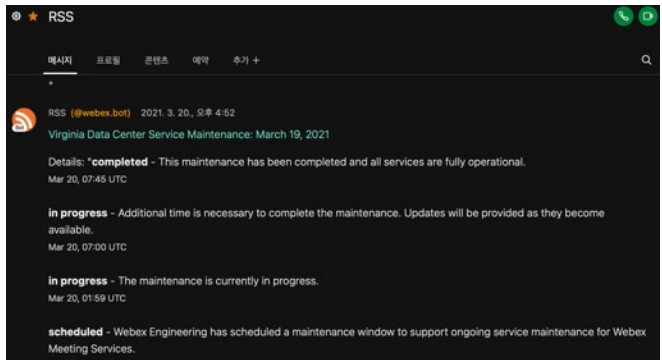
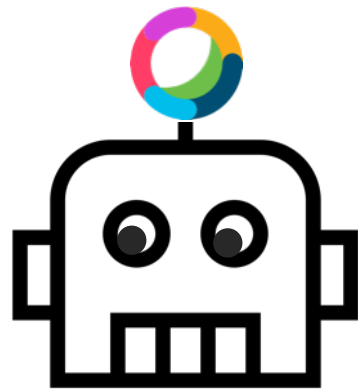
Recents
Widget

Video
Call

Space
Widget

#5 - Bots

- “Machine” 계정
- 사용자와 상호작용(대화)
- 사용자에게 알림



- 서비스 추적 봇
- 미팅 예약 봇
- Expense bot
- 번역 봇
- 알림/공지 봇

국내 사례

The screenshot displays the Virtual COLLAB interface. The top header shows the Virtual COLLAB logo and a user profile for '홍길동' (Hong Gildong) with email 'honglove@vcamp.co.kr'. Below the header, there's a search bar with the placeholder text '검색할 키워드를 입력하세요.' (Enter keywords to search for). The main content area features two meeting cards. The first card, titled '주간회의' (Weekly Meeting), shows the date '19.10.23' and time '12:00 ~ 13:00', with the topic '11주차 정기 주간회의' (11th Week Regular Weekly Meeting) and a status '! 시작까지 10분 남음' (10 minutes remaining until start). The second card, titled '사내교육' (In-house Education), shows the same date and time, with the topic 'IT 트렌드와 차세대 UX환경' (IT Trends and Next-Generation UX Environment) and a status '53분 동안 진행 되었습니다.' (Completed for 53 minutes). Below the topic, it shows '13명' (13 people) in a blue bar, '2명' (2 people) in an orange bar, and '17명' (17 people) in a grey bar. At the bottom of the cards are buttons for '시작하기' (Start) and '출결정보' (Attendance Information). On the right side, there's a Webex Teams sidebar with a search bar and a list of teams: 'SW사업부 (125)', '시스템사업부 (30)', '시스템사업팀 (10)', '서비스팀 (10)', '통신사업부 (10)', '경영지원실 (10)', and '경영지원팀 (10)'. At the bottom of the sidebar is a button '+ 사용자 추가하기' (Add User).

대학교 LMS 연동

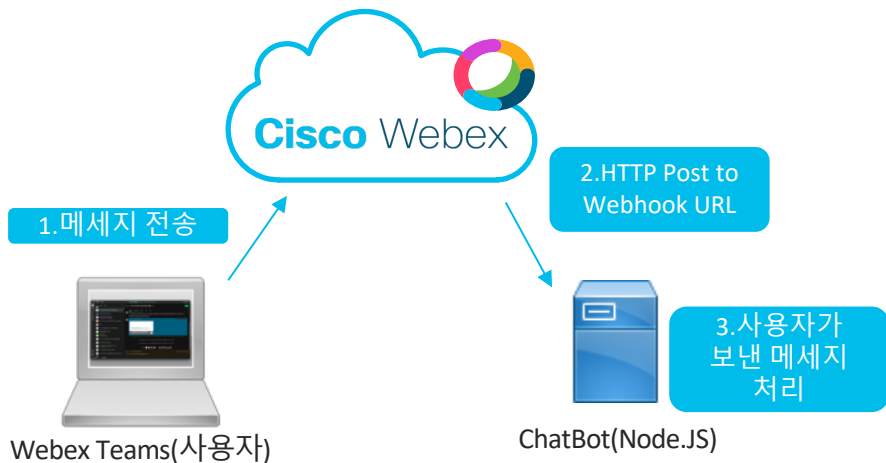
- 학교 LMS 사이트에서 원격 강의(webex) 예약/접속
- 자동 출결 체크
- 수업 종료 후 레코딩을 통한 다시보기
- Webex Teams를 이용한 조별과제



Let's Try!

나도 할 수 있다 손쉬운 채팅 봇 만들기#준비단계

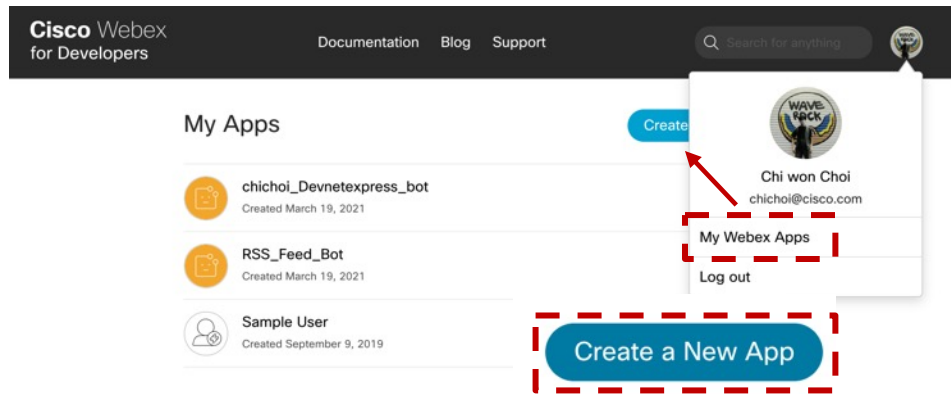
1. Webex 계정 : 내가 운영하는 Bot의 계정을 만들기 위해 필요!
2. Node.JS/npm Runtime : 채팅봇이 내 PC에서 Node.js로 운영 되요.
3. Ngrok : 내 PC에서 운영되는 봇 서버가 Webex와 Public IP없이 통신
4. Git : 실습을 할 코드들의 샘플들을 카피 받을 거예요
5. Sublime(option) : 코드가 이쁘게 보여요..



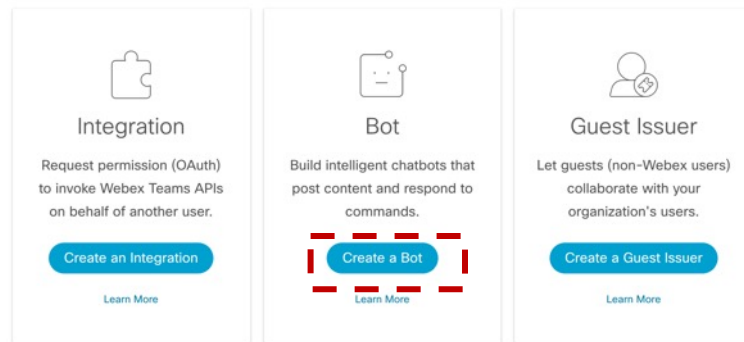
- #1 내가 Bot에게 전달한 메시지는
- #2 Webex Cloud로 전달이 되고
- #3 Webex Cloud가 내 ChatBot Server가 있는곳으로 메시지는 다시 전달
- #4 ChatBot Server는 보낸 메시지를 처리

Step#1 Bot Account 생성

<https://developer.webex.com>



Create a New App



Step#1 Bot Account 생성

New Bot

Bot name*
Name of your bot as it will appear in Webex Teams.

Bot username*
The username users will use to add your bot to a space. Cannot be changed later.

Icon*
Upload your own or select from our defaults. Must be exactly 512x512px in JPEG or PNG format.

Description*
Provide some details about what your bot does, how it benefits users, and most importantly, how a user can get started using it. The description should be under 1500 characters. You can use bullets, links, and Markdown formatting. If your app is listed on the Webex App Hub, this field will be used as the listing's description.

By creating this app, you accept the [Terms of Service and Privacy Statement](#).

Cancel Add Bot

Bot의 이름입니다. 나중에 언제든지 바꿀 수 있어요

Bot의 AccessToken입니다. 내 nodeJS 봇 서버가 이 토큰을 통해 내가 만든 봇 계정과 연결이 되요. Token값은 메모장에 따로 저장해놓고 알려주지 마세요!

Bot의 메일 주소(ID)입니다. 이 주소를 통해 대화를 나눌 수 있어요 바꿀 수 없어요

다 입력하고 Add Bot 클릭

Congratulations!

chiwon_devnet_bot is one step closer to becoming a reality.

chiwon_devnet_bot

Next Step: Use your Bot Access Token to set up your webhook and finish building your bot.

Bot access token

Non-expiring (good for 100 years)
access token for your bot. Save this token to set up your webhook.

Helpful resources to build bots

GUIDES AND BLOGS

- Guide: Bots
- Blog: From Zero to Webex Teams Chatbot in 15 minutes

LEARNING LABS

- Creating Chatbots for Webex Teams

WEBEX APP HUB

Think chiwon_devnet_bot can help others too? It's always nice to share. Consider submitting to Webex App Hub for others to add.

- ☐ Explore: Webex App Hub
- ☐ Read our Guidelines: App Hub Submission Process

Bot name*
Name of your bot as it will appear in Webex Teams and Webex App Hub.

Bot username*
The username users will use to add your bot to a space. Cannot be changed later.

Step#2 Bot Sample Code 다운로드-Git

```
botkit-template -- chicho@CHICHOI-M-X18P -- zsh -- 118x24
..tkit-template

x chicho@CHICHOI-M-X18P ~~/Desktop/DevnetExpress
x chicho@CHICHOI-M-X18P ~~/Desktop/DevnetExpress git clone https://github.com/CiscoDevNet/botkit-template

Cloning into 'botkit-template'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 427 (delta 0), reused 0 (delta 0), pack-reused 424
Receiving objects: 100% (427/427), 395.02 KiB | 2.03 MiB/s, done.
Resolving deltas: 100% (264/264), done.
chicho@CHICHOI-M-X18P ~~/Desktop/DevnetExpress cd botkit-template

chicho@CHICHOI-M-X18P ~~/Desktop/DevnetExpress/botkit-template master npm install

> core-js@2.6.11 postinstall /Users/chicho/Desktop/DevnetExpress/botkit-template/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

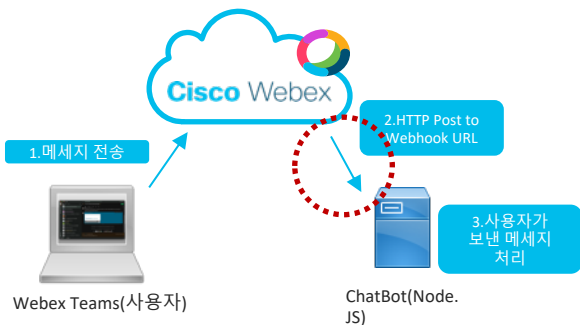
The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)
```

git clone https://github.com/CiscoDevNet/botkit-template
명령어를 통해 github에 저장된 샘플 코드를 다운로드 받습니다

다운로드 후, botkit-template 로 이동하여 npm을 설치하여 테스트
환경구성을 마무리 합니다

Step#3 Ngrok를 이용하여 터널링 구성



Ngrok가 필요한 이유,

Webex Cloud Platform이 사용자가 전송한 메시지를 챗봇 서버에 보내기 위해서는 **클라우드에서 챗봇서버에게 도착 가능한 Public 주소가** 필요합니다.

Ngrok는 잠시 동안 Local PC와 Ngrok 클라우드와 터널링을 뚫어 지정한 로컬 PC 위치로 Redirect 시켜줍니다.

```
botkit-template -- chicho@CHICHOI-M-X18P -- ssh -- 118x24
Last login: Wed Mar 24 09:32:26 on ttys001
chicho@CHICHOI-M-X18P: ~$ curl -X POST -H "Content-Type: application/json" -d '{"text": "hello"}' http://localhost:3000
{"text": "hello"}
chicho@CHICHOI-M-X18P: ~$ ngrok http 3000
ngrok by @inconshreveable

Session Status: online
Session Expires: 1 hour, 59 minutes
Version: 2.3.35
Region: United States (us)
Web Interface: http://127.0.0.1:4040
Forwarding: https://59883cfb15.ngrok.io -> http://localhost:3000
Forwarding: https://59883cfb15.ngrok.io -> http://localhost:3000
Connections:
  ttl    opn    rt1    rt5    p50    p99
   0     0     0.00  0.00  0.00  0.00
```

```
# Local port where your bot will be started
# defaults to 3000

PORT=3000

# Webex JavaScript SDK debug level
# Values: error, warn, info, debug, trace
WEBEX_LOG_LEVEL=error
```

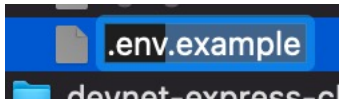
Ngrok **http 3000** 명령어를 입력하여 3000포트로 리다이렉트 시킬수 있도록 Ngrok를 실행 시킵니다.

Why 3000? > 봇킷의 샘플코드가 3000번을 통해 수신하도록 설정되어 있습니다. **.env.example** 에서 이 포트번호를 바꾸면 그에 맞추어 리다이렉트 시킬수 있도록 설정을 바꿉니다.

Session Status가 **online**이 나오면 성공

Forwarding에 나온 https 주소를 메모장에 저장해 둡니다.

Step#4 botkit 구성을 위한 env 환경 설정



botkit-template에 있는 .env.example의 파일명을 .env로 바꾸어 주고 파일을 sublime에서 열어 봅니다

```
10 #
11
12 # Webex Teams bot account access token
13
14 WEBEX_ACCESS_TOKEN=ZTYyOTlmZmUtY2NkMi00YmZmLWI0ZGUtY2VlMjI4NmM4MGJjZWExZTA5YzAtMDE2_Pf84_1eb65fdf-9643-4
15 17f-9974-ad72cae0e10f
16
17 # Internet facing URL where your bot can be reached for incoming webhooks or
18 # HTTP web server requests.
19 # Can be empty if WEBSOCKET_EVENTS is True and the Botkit web server is
20 # not needed, e.g. if not serving images related to the buttons & cards feature
21 # or the health check URL
22 PUBLIC_URL=https://5d083c3fbf15.ngrok.id
23
24 # Use websockets instead of webhooks for incoming Webex events (True/False)
25
26 WEBSOCKET_EVENTS=True
27
28 # Local port where your bot will be started
29 # defaults to 3000
30
31 PORT=3000
32
```

WEBEX_ACCESS_TOKEN, PUBLIC_URL에 각각 Step#1과 #3에서 저장해 둔 bot Access Token값과 Ngrok Forwarding URL(HTTPS)를 넣고 저장해 줍니다.

Step#4 botkit 구성을 위한 env 환경 설정

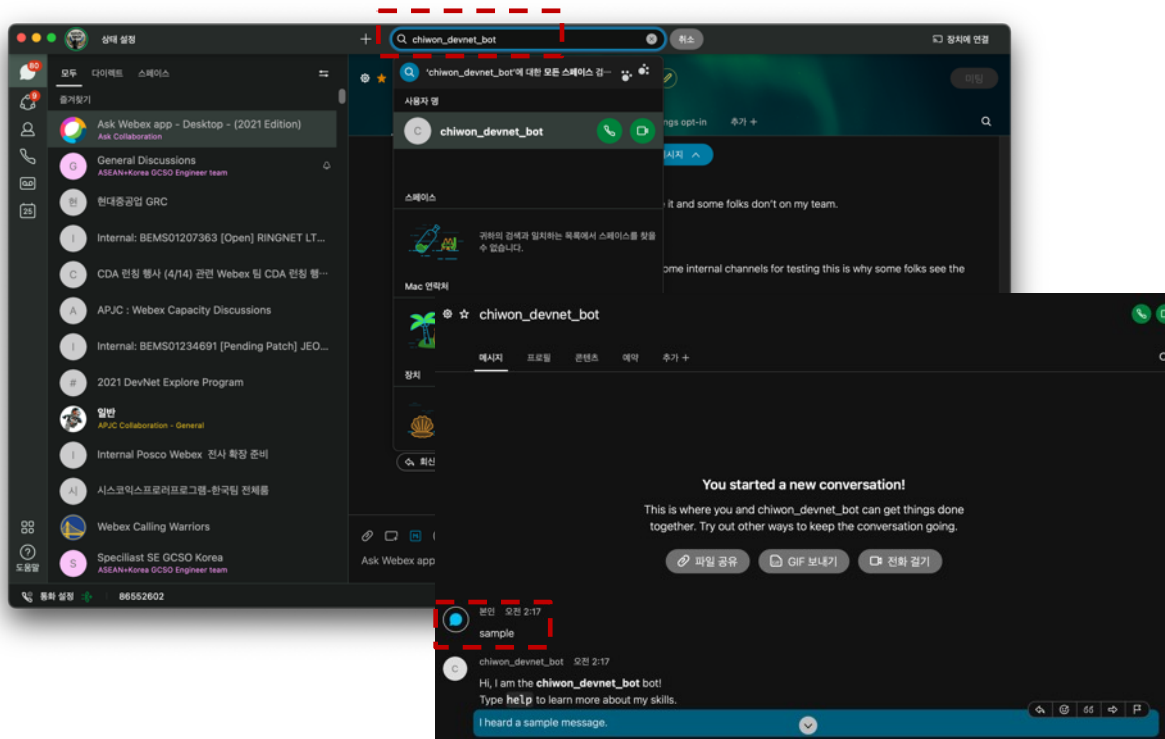
```
chicho@CHICHOI-M-X18P ~/Desktop/DevnetExpress/botkit-template master • node bot.js
LoggerProxy->info:NO LOGGER DEFINED
WARNING: No secret specified. Source of incoming webhooks will not be validated. https://developer.webex.com/webhooks-explained.html#auth
** Your bot is using memory storage and will forget everything when it reboots!
** To preserve dialog state, specify a storage adapter in your Botkit config:
** const controller = new Botkit({storage: myStorageAdapter});
Enabling plugin: Webex Adapter
Health check available at: https://5d083c3fbf15.ngrok.io
Webhook endpoint online: http://localhost:3000/api/messages
Using websockets for incoming messages/events
```

Node bot.js 명령어를 통해 bot.js 코드를 실행 시켜 봅니다.

```
← → ↺ ⌂ 5d083c3fbf15.ngrok.io
Safari KrDag Unified Networkin... Shodan Intranet Google CISCO CC
1 // 20210325020450
2 // https://5d083c3fbf15.ngrok.io/
3
4 {
5   "upSince": "Wed, 24 Mar 2021 17:04:33 GMT",
6   "botName": "chiwon_devnet_bot",
7   "botVersion": "v1.0.1",
8   "botkitVersion": "4.10.0",
9   "platform": "Webex",
10  "code": "https://github.com/CiscoDevNet/botkit-template"
11 }
```

Public URL에 설정한 Ngrok forward URL 을 브라우저를 통해 접속하여 다음과 같은 결과가 나온다면 정상적으로 봇 서비스가 실행 중!

Step#5 실제 대화를 해보자



1. Webex Teams를 실행시켜, Step#1에서 설정한 Bot User Name을 검색하여 대화 시작

2. Sample 을 입력하였을 때의 반응은?

3. Help 입력해서 bot과 할 수 있는 다양한 기능을 확인해 보자

Step#6 Code – 대화형 실전

```
const { BotkitConversation } = require( 'botkit' );

module.exports = function( controller ) {

  const convo = new BotkitConversation( 'color_chat', controller );

  convo.say( 'This is a Botkit conversation sample.' );
  convo.ask(
    'What is your favorite color?',
    async( answer, convo, bot ) => {},
    'stated_color'
  );
  convo.say( `Cool, I like {{ vars.stated_color }} too!` );

  controller.addDialog( convo );

  controller.hears( 'color', 'message,direct_message', async( bot, message ) => {

    await bot.beginDialog( 'color_chat' );

  });

  controller.commandHelp.push( { command: 'color', text: 'Pick a favorite color (Botkit conversations)' } );

}
```

코드를 복사해서 features 폴더안에
color.js 에 저장.

Bot.js를 다시 실행 하고 color 메시지를
입력해보자

Step#6 Code – 대화형 실전

```
const { BotkitConversation } = require( 'botkit' ); //botkit 모듈을 불러옵니다.

module.exports = function( controller ) {

  const convo = new BotkitConversation( 'color_chat', controller ); // color_chat이라는 새로운 대화 개체 생성

  convo.say( 'This is a Botkit conversation sample.' ); //color 모듈 시작시 먼저 해당 메세지 사용자에게 전달
  convo.ask(
    'What is your favorite color?',
    async( answer, convo, bot ) => {},
    'stated_color'
  ); // 질문을 하고, 이후 전달되는 답변을 stated_color 변수에 저장
  convo.say( `Cool, I like {{ vars.stated_color }} too!` ); // 앞에서 저장한 변수 stated_color 를 사용하여 답변

  controller.addDialog( convo );

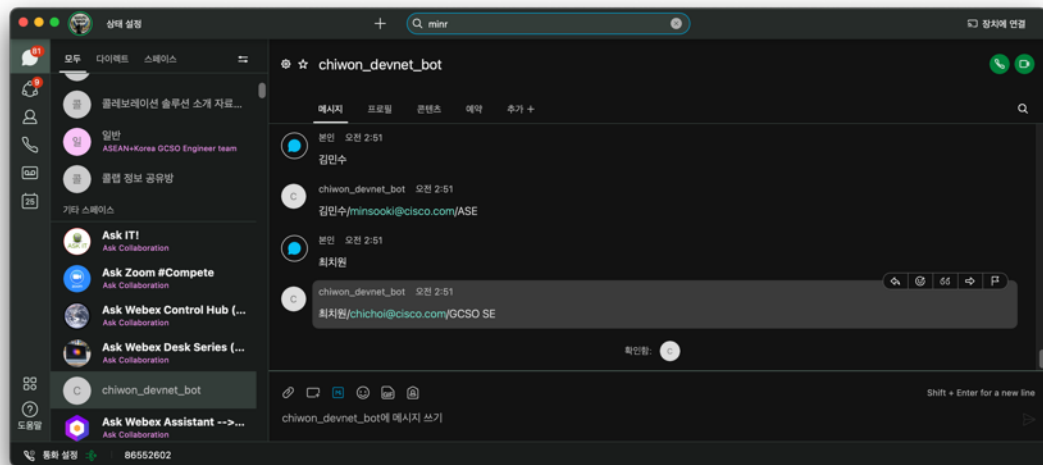
  controller.hears( 'color', 'message,direct_message', async( bot, message ) => {

    await bot.beginDialog( 'color_chat' ); // 다시 color라는 메세지가 올때까지 대기하고 color라는 메세지가 들어올 경우 color_chat 실시
  });

  controller.commandHelp.push( { command: 'color', text: 'Pick a favorite color (Botkit conversations)' } );

}
```

숙제! 아주 간단하게 디렉토리 검색?



이름을 입력하면 내 정보가 나오는
디렉토리 봇을 만들어 보자.

대화형식으로 해도 좋고
이름만 넣으면 답변을 주어도 좋아!

참고할만한 코드
순한맛 : `sample_echo.js`
매운맛 : `color.js` 또는 `coffe.js`

각자의 팀방(팀1~팀4)에 봇을 초대해서
내 이름을 제대로 답변하는지
보여주세요!



The bridge to possible