



Cisco Developers Connect

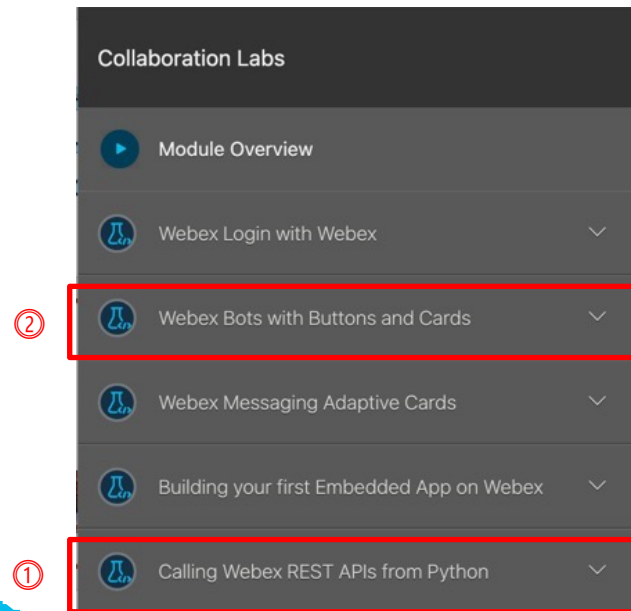
May 23rd 2024, Seoul Korea

Collaboration Labs for Webex

김도경 프로, Test SE

Hands-on 순서

- Python에서 Webex REST API 호출
- 미션 #1
- Webex 가상 화폐 시세 봇 만들기
- 미션 #2



필요한 통화 기능, 어디서나 사용 가능

The image is a collage of screenshots from the Cisco Webex application, demonstrating its features. The central screenshot shows a video call in progress with a woman speaking. Surrounding this are several smaller screenshots and callouts:

- Top Left:** A screenshot of a contact card for Emily Nakagawa, showing her profile picture, name, and work phone number (+1 408 555 2346). Below the card are buttons for Audio, Video, Delete, and Mark as unread.
- Top Right:** A screenshot of a contact card for Umar Patel, showing his profile picture, name, and various contact information including work and mobile phone numbers, email, and home address. It also has a 'Call on Webex' button.
- Bottom Left:** A screenshot of the 'Call settings' menu, showing options like 'Hunt group', 'Forward phone calls to', 'Single number reach', and 'Mobile'.
- Bottom Center:** A screenshot of the 'Call settings' menu with the 'Open Call Preferences' button highlighted.
- Bottom Right:** A screenshot of a video call interface showing a woman speaking, with a 'Mute' button and other controls at the bottom.

Callouts and labels point to specific features:

- 음성메일을 텍스트로 변환** (Transcription of voice mail to text) - points to the 'Transcription' section in the contact card.
- 채신전화 (CFD), Single Number Reach** (Call Forwarding, Single Number Reach) - points to the 'Call settings' menu.
- Open call preferences** - points to the 'Open Call Preferences' button.
- 데스크 전화기 및 화상장치와 페어링 (인텔리전트 프록시미터)** (Pairing desk phone and video device (Intelligent Proximity Meter)) - points to the 'Connect to a device' button.
- Webex App 콜** (Webex App Call) - points to the 'Call on Webex' button.
- 콘텐츠/화면 공유중 음성/영상 통화** (Voice/Video call with content/screen sharing) - points to the video call interface.
- AI 기반 오디오 인텔리전스: 노이즈 제거, 보류/해제, 돌려주기, 컨퍼런스, 전화통화중 웹엑스 미팅으로 전환** (AI-based audio intelligence: noise removal, hold/unhold, transfer, conference, switch to Webex meeting during phone call) - points to the video call interface.

Meet your workforce productivity needs with **one app**



콜링

Calling capabilities such as intelligent routing, hold / retrieve, IP Phone control



미팅

Full-feature Webex Meetings UX



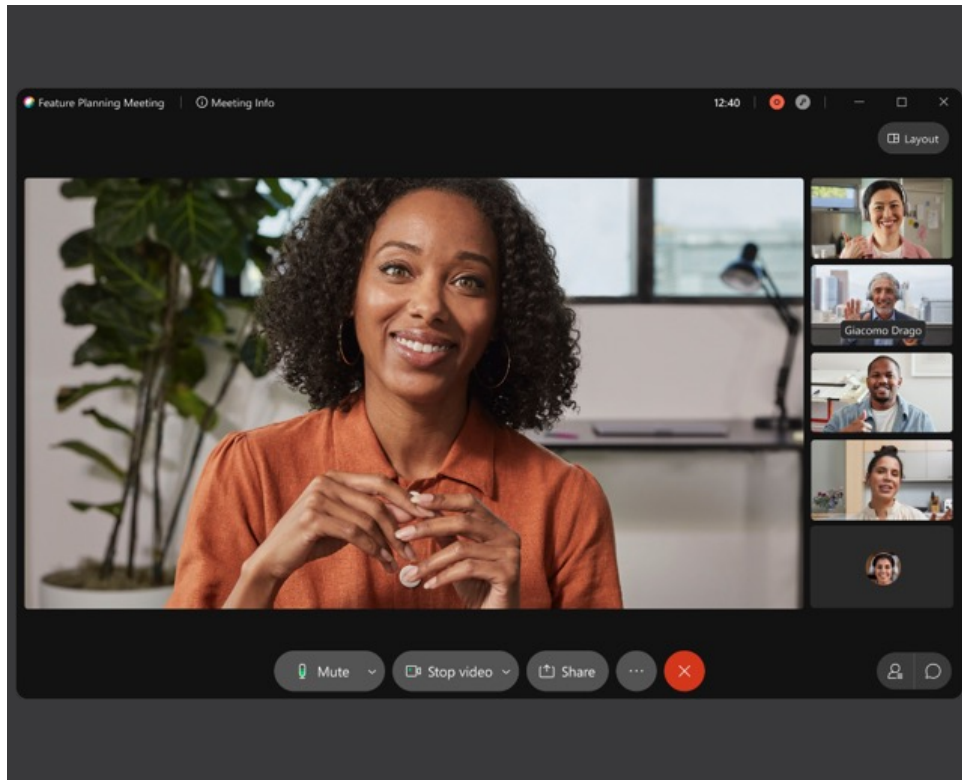
협업

Persistent chat with capabilities such as reactions and threading



보안

Integration with DLP; Cisco CloudLock-ready or 3rd-party CASB integration



디바이스 무선 연결

Pair/share wirelessly with Webex video devices to join meetings or share content



API 연동

Simplify and accelerate workflows by integrating business applications directly into the app



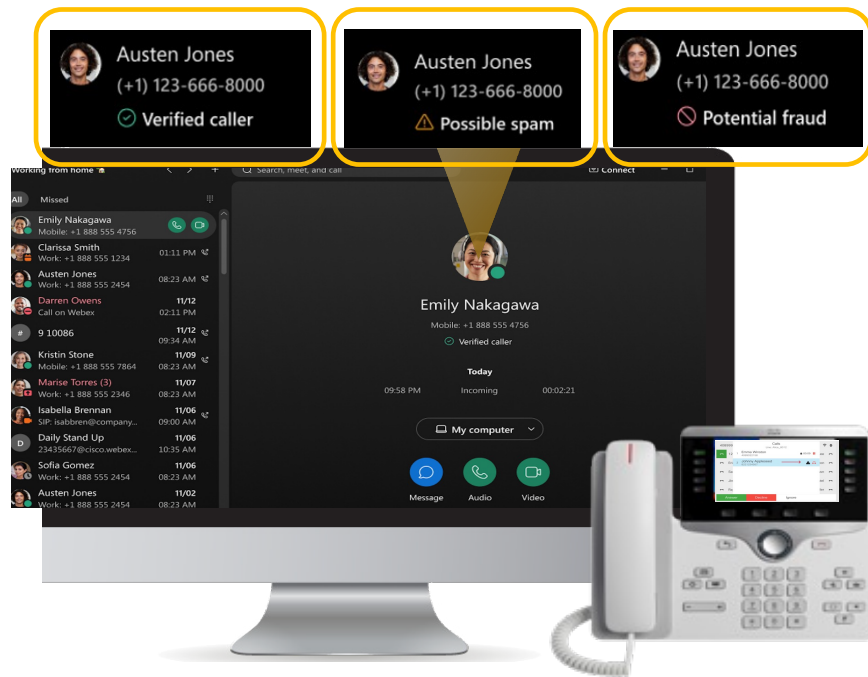
중앙 집중 통합 관리

Control Hub: single pane of glass to deploy, manage and support the app

Webex App

스팸 등록

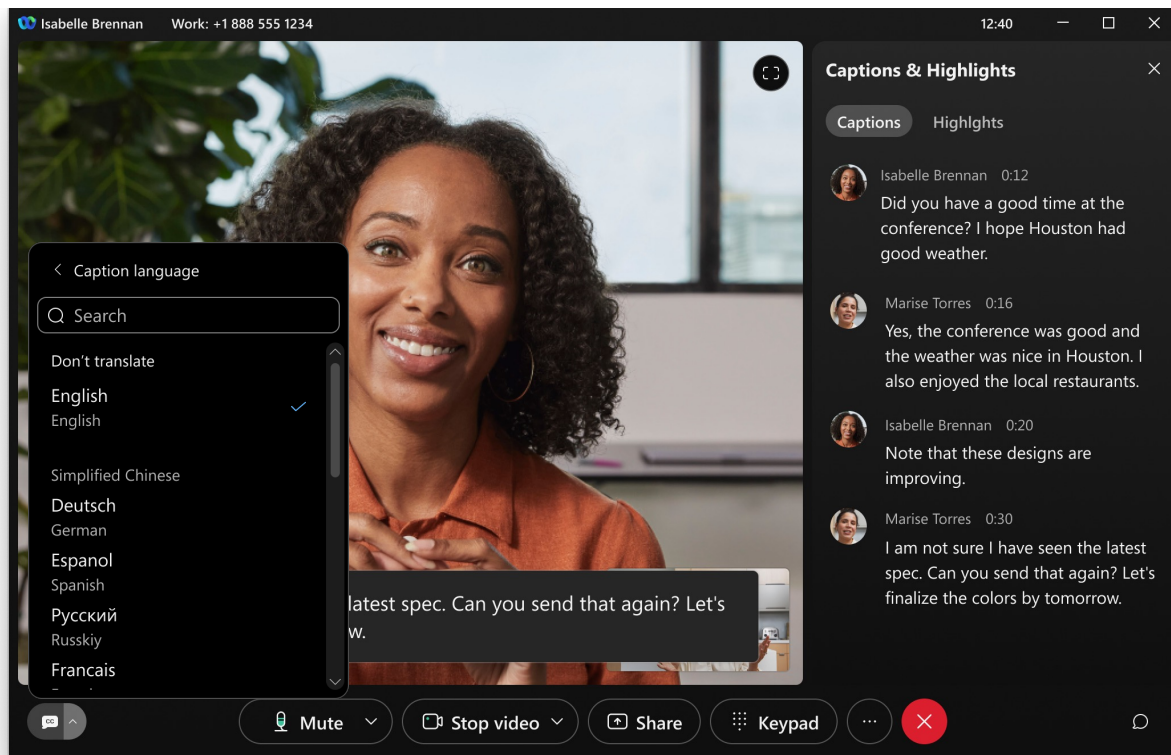
- Webex App과 시스코의 IP-Phone에서는 수신된 전화번호를 식별
- STIR/Shaken 프로토콜을 활용하여, 개인이 설정 또는 그룹관리자가 정책을 정의
 - ✓ Verified Caller: 확인 가능(한) 전화번호
 - ✓ Possible Spam: Spam 전화
 - ✓ Potential fraud: 알 수 없는 전화(또는 Fraud Calls)



Webex App

실시간 번역기능

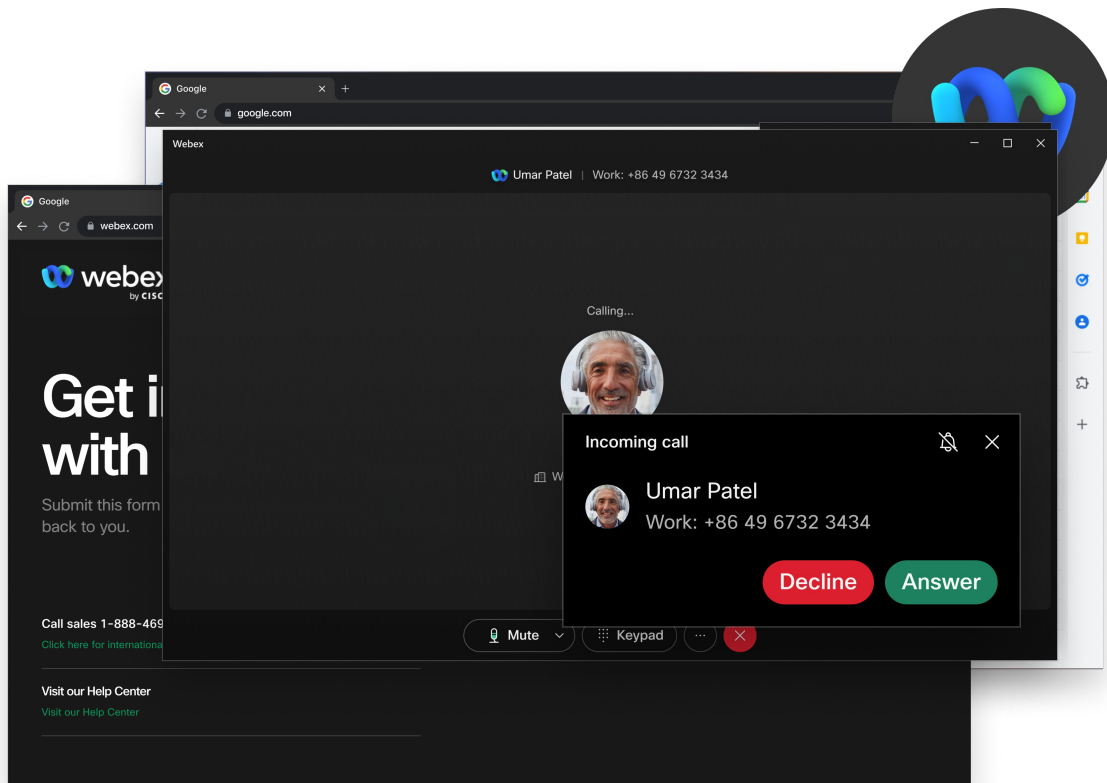
- 미팅에 적용된 실시간 번역기능을 전화에서도 사용
- 원클릭으로 캡션 활성화/비활성



Webex App

구글/크롬 Softphone

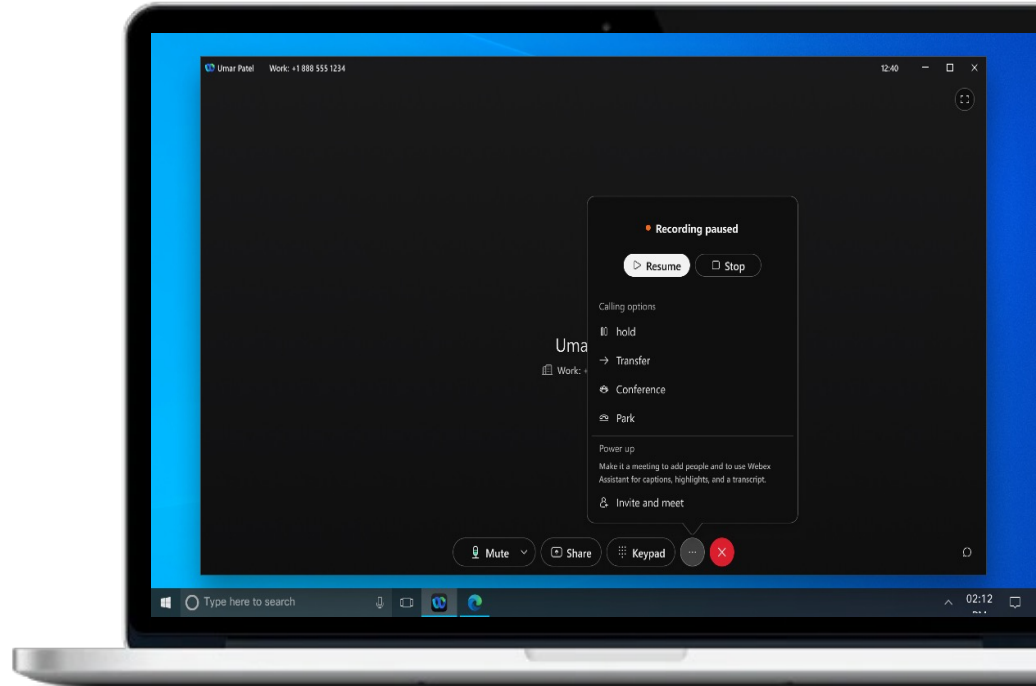
- WebRTC기반의 소프트 폰
- Webex Calling Chrome Extension으로 등록 가능
- 돌려주기/홀드 기능 지원
- 구글 워크플레이스나 크롬에서 클릭투 콜 제공



Webex App

녹취 서비스

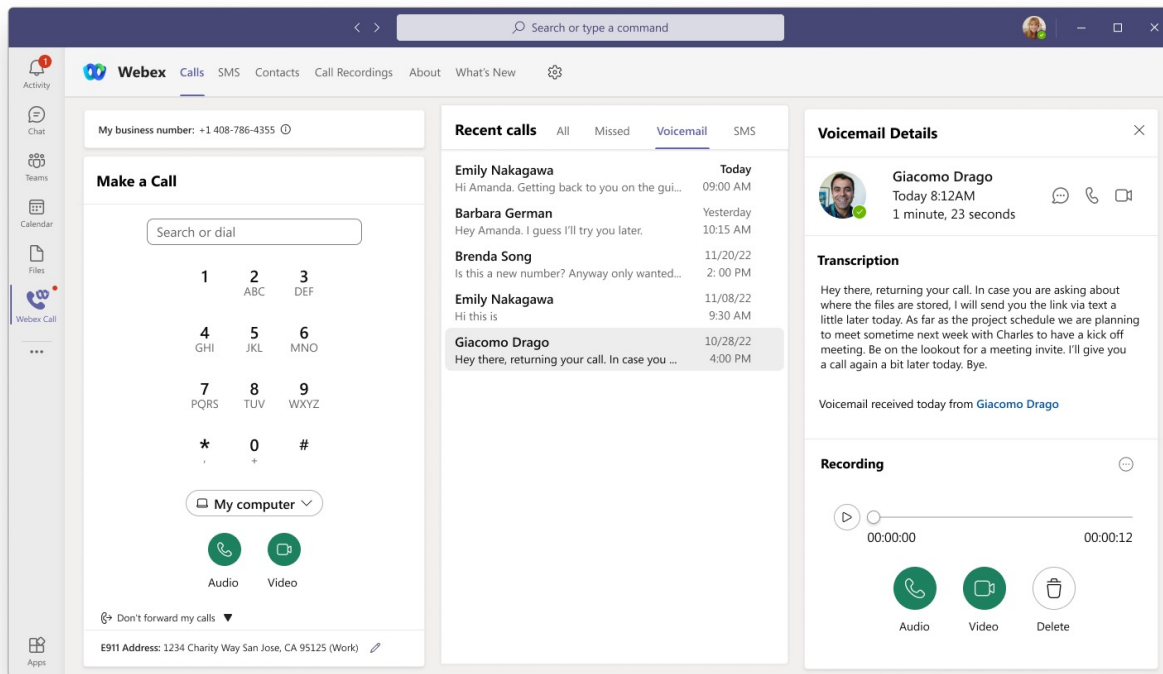
- 무료 서비스(30일간 저장공간) 제공
- 전수녹취 / 부분녹취 기능 제공
- 녹취(Dubber) 포탈 제공



Webex App

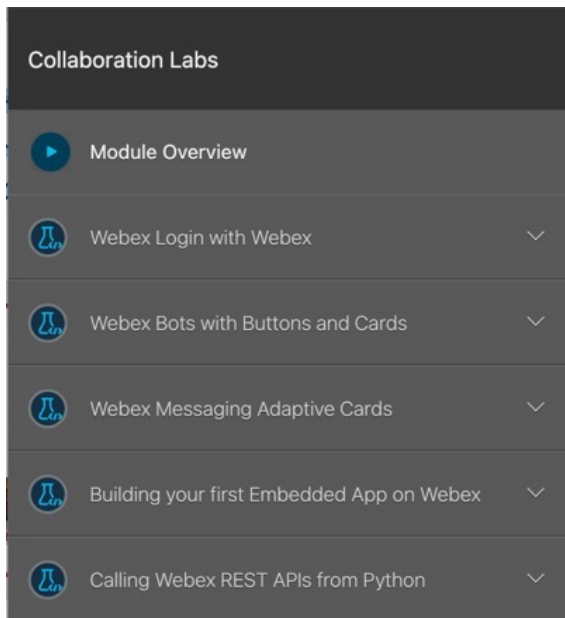
Microsoft Teams 연동

- MS Teams 앱에 플러그인 형태로 설치되고 크로스 런치 형태로 구동
 - inapp에 다이얼패드 및 콜링 옵션이 활성화

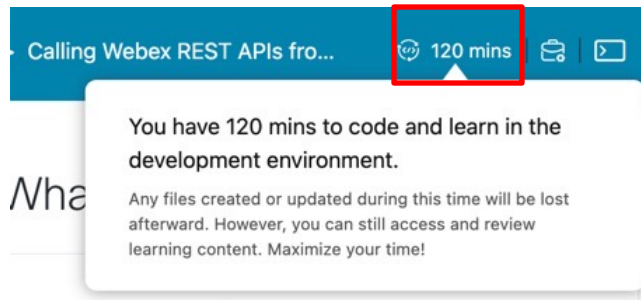
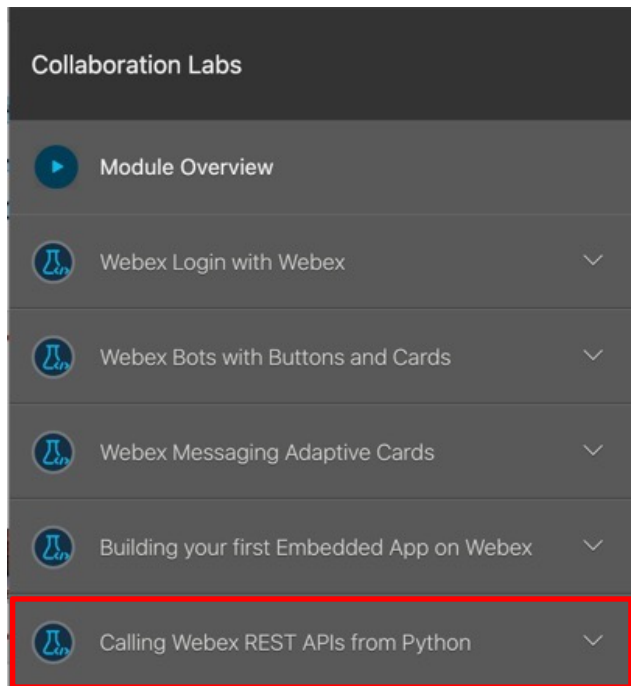


DEVNET 러닝랩 접속

<https://developer.cisco.com/learning/modules/collaboration-labs/>



Python에서 Webex REST API 호출



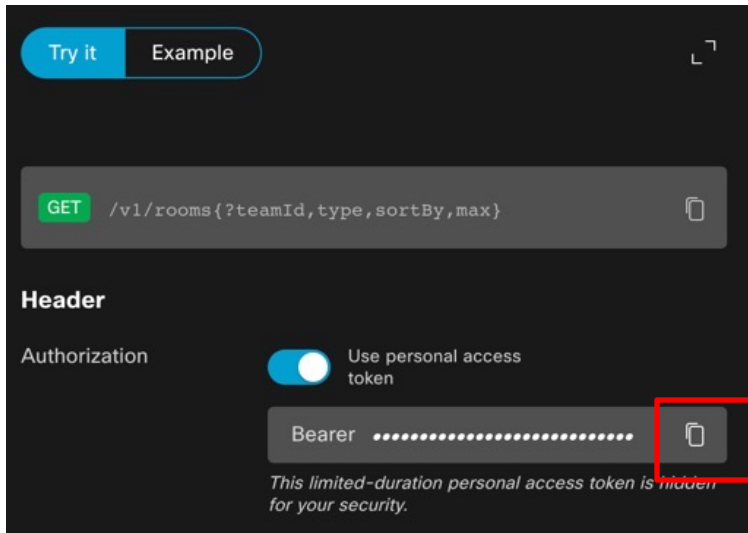
Python에서 Webex REST API 호출

Access Token 가져오기

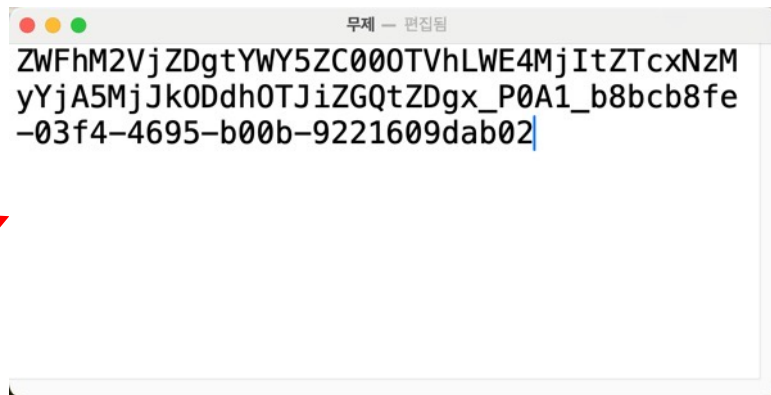
<https://developer.webex.com/docs/api/v1/rooms/list-rooms>

2

REST GET Request with the Interactive Documentation



메모장에 Access Token 복사



Python에서 Webex REST API 호출

Query Parameters 종류

teamId

string

List rooms associated with a team, by ID. Cannot be set in combination with `orgPublicSpaces`.

type

string

List rooms by type. Cannot be set in combination with `orgPublicSpaces`.

Possible values: direct, group

orgPublicSpaces

boolean

Shows the org's public spaces joined and unjoined. When set the result list is sorted by the `madePublic` timestamp.

from

string

Filters rooms, that were made public after this time. See `madePublic` timestamp

to

string

Filters rooms, that were made public before this time. See `madePublic` timestamp

sortBy

string

Sort results. Cannot be set in combination with `orgPublicSpaces`.

Possible values: id, lastactivity, created

max

number

Limit the maximum number of rooms in the response. Value must be between 1 and 1000, inclusive.

Default: 100

Python에서 Webex REST API 호출

파이썬 가상 환경 셋팅

4

REST GET Request with Python

1. To prevent any Python packages we install from disrupting the OS or other projects, create and activate a new Python `venv` :

```
# For Windows use: py -3  
python3 -m venv venv  
source venv/bin/activate
```



Python venv 활성화

2. Next install the Python `requests` package, which will be needed for making HTTP REST requests:

```
pip install requests
```

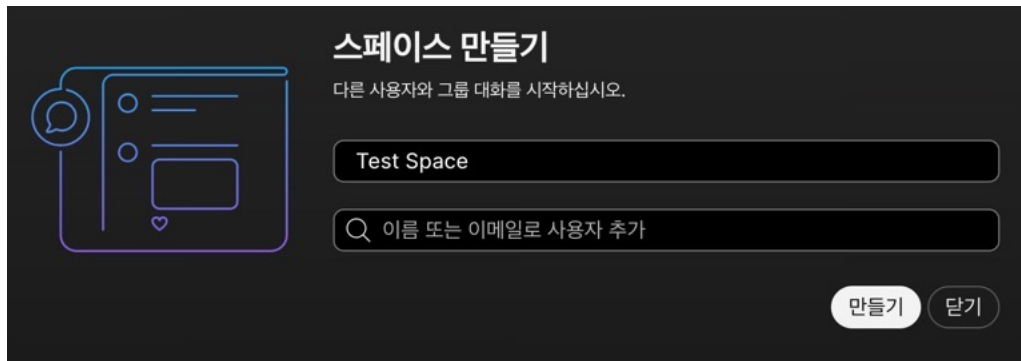


Python requests 패키지 설치

Python에서 Webex REST API 호출

Webex 스페이스 생성

<https://web.webex.com/sign-in>



The image shows a dark-themed user interface for creating a Webex Space. On the left is a line-art icon of a document with a speech bubble and a heart. To the right, the title '스페이스 만들기' (Create Space) is displayed in white. Below the title is the instruction '다른 사용자와 그룹 대화를 시작하십시오.' (Start a conversation with other users and groups). There are two input fields: the first contains the text 'Test Space', and the second is a search field with a magnifying glass icon and the placeholder text '이름 또는 이메일로 사용자 추가' (Add user by name or email). At the bottom right are two buttons: '만들기' (Create) and '닫기' (Close).

스페이스 만들기

다른 사용자와 그룹 대화를 시작하십시오.

Test Space

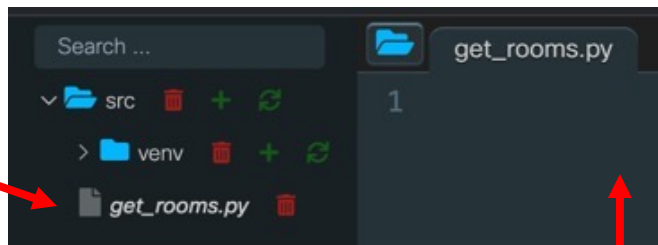
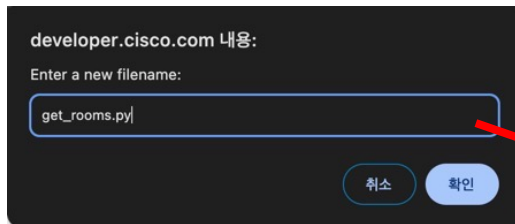
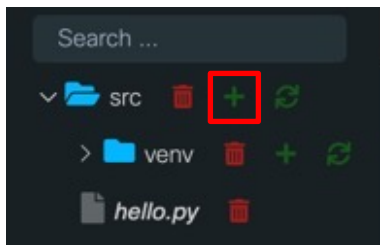
이름 또는 이메일로 사용자 추가

만들기 닫기

Python에서 Webex REST API 호출

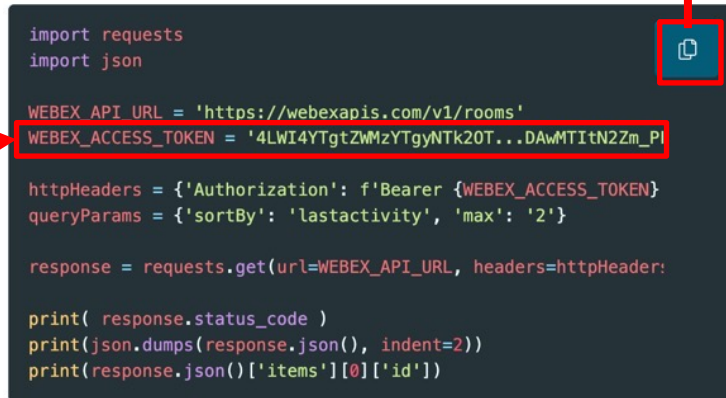
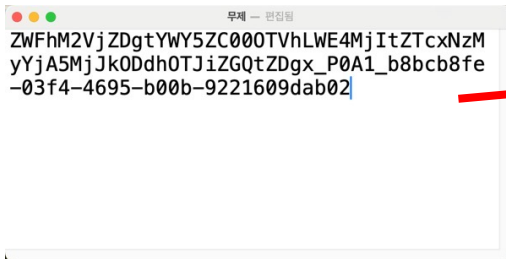
REST GET Request

/src 폴더에 get_rooms.py 생성



get_rooms.py에 코드 복사

메모장에 복사했던 Access Token 붙여넣기



Python에서 Webex REST API 호출

REST GET Request

```
import requests
import json

WEBEX_API_URL = 'https://webexapis.com/v1/rooms'
WEBEX_ACCESS_TOKEN = '4LWI4YTgtZWZYTgyNTk20T...DAwMTItN2Zm_PI

httpHeaders = {'Authorization': f'Bearer {WEBEX_ACCESS_TOKEN}'
queryParams = {'sortBy': 'lastactivity', 'max': '2'}

response = requests.get(url=WEBEX_API_URL, headers=httpHeaders, params=queryParams)

print( response.status_code )
print(json.dumps(response.json(), indent=2))
print(response.json()['items'][0]['id'])
```

- JSON 데이터 형식을 가져오기 위한 라이브러리

- HTTP 헤더를 보관할 변수
- 항목 필터를 위한 HTTP 쿼리 파라미터

- API 요청을 수행

Python에서 Webex REST API 호출

REST GET Request

get_rooms.py 실행

```
python get_rooms.py
```



최근 활동이 있는 room 2개의 정보 출력

```
(venv) developer:src > python get_rooms.py
```

```
200
```

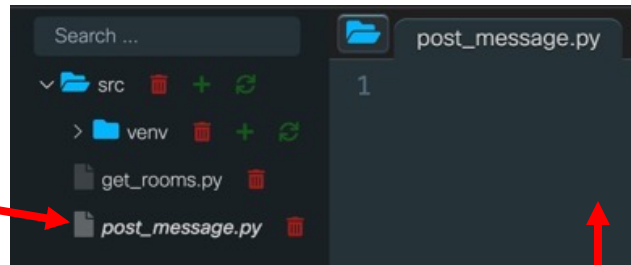
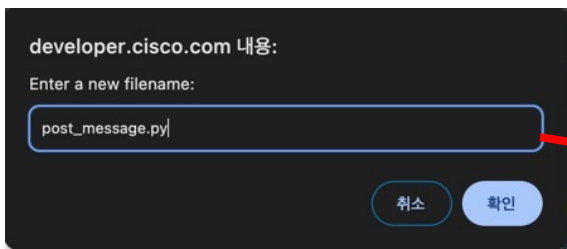
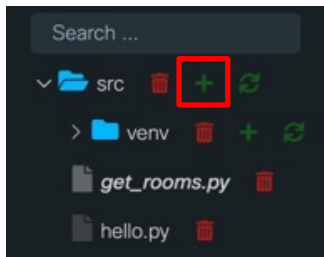
```
{
  "items": [
    {
      "id": "Y2lzY29zcGFyazovL3Vyb2pURUFN0nVzLXdlc3QtMl9yL1JPT00vZjI5MGE0NjAtMGMyYS0xMwVmLTk3NTEtMGYzM2U0ZDBjN2Q1",
      "title": "Test Space",
      "type": "group",
      "isLocked": false,
      "lastActivity": "2024-05-07T04:33:26.950Z",
      "creatorId": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS8xNGMxMjhmMi1hZDZlLTRlOTQtODY4My1kNGNiMTc1N2MzYzg",
      "created": "2024-05-07T04:33:26.950Z",
      "ownerId": "Y2lzY29zcGFyazovL3VzL09SR0F0SVpBVElPTi9iOGJjYjhmZS0wM2Y0LTQ2OTU0YjAwYi05MjIxNjA5ZGFmIDI",
      "isPublic": false,
      "isReadOnly": false
    },
  ],
}
```

다음 실습을 위해 Room id 메모장에 복사

Python에서 Webex REST API 호출

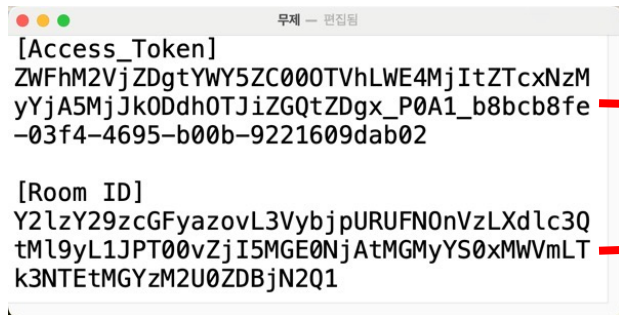
REST API POST Request

/src 폴더에 post_message.py 생성



post_message.py에 코드 복사

메모장에 복사했던 Access Token, Room ID 붙여넣기



Python에서 Webex REST API 호출

Response Properties 종류

<https://developer.webex.com/docs/api/v1/messages/create-a-message>

text

string

The message, in plain text. If `markdown` is specified this parameter may be *optionally* used to provide alternate text for UI clients that do not support rich text.

markdown

string

The message, in Markdown format.

html

string

The text content of the message, in HTML format. This read-only property is used by the Webex clients.

files

array[string]

Public URLs for files attached to the message. For the supported media types and the behavior of file uploads, see [Message Attachments](#).

personId

string

The person ID of the message author.

personEmail

string

The email address of the message author.

mentionedPeople

array[string]

People IDs for anyone mentioned in the message.

mentionedGroups

array[string]

Group names for the groups mentioned in the message.

attachments

array[Attachment]

Message content attachments attached to the message. See the [Cards Guide](#) for more information.

REST API POST Request

- JSON 데이터 형식을 가져오기 위한 라이브러리

- HTTP 헤더를 보관할 변수
- 항목 필터를 위한 HTTP 쿼리 파라미터

```
body = {'roomId': 'Y2lzY29zcGFyZredactedxMWU2LTg1WEt0WYxODkzNjRkMGUx', 'text': 'Hello World!'}
```

```
response = requests.post(url=WEBEX_API_URL, headers=httpHeader)
```

- API 요청을 수행

```
print(response.status_code)
print(json.dumps(response.json(), indent=2))
```



Python에서 Webex REST API 호출

REST API POST Request

post_message.py 실행

```
python post_message.py
```



실행 결과

```
(venv) developer:src > python post_message.py
200
{
  "id": "Y2lzY29zcGFyazovL3VybjpURUFN0nVzLXdlc3QtMl9yL01FU1NBR0UvNzM2YTQyNjAtMGMMyZC0xMwVmLTg2YTktZGI1YThiNDNkMDk3",
  "roomId": "Y2lzY29zcGFyazovL3VybjpURUFN0nVzLXdlc3QtMl9yL1JPT00vZjI5MGE0NjAtMGMMyYS0xMwVmLTk3NTEtMGYzM2U0ZDBjN2Q1",
  "roomType": "group",
  "text": "Hello World!",
  "personId": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS8xNGMxMjhmMi1hZDZlLTRlOTQtODY4My1kNGNiMTc1N2MzYzg",
  "personEmail": "stargirl7924@gmail.com",
  "created": "2024-05-07T04:51:22.118Z"
}
(venv) developer:src > █
```

Test Space ☆

메시지

사용자 (1)

콘텐츠

Meetings

김

본인 오후 1:51

Hello World!

미션!



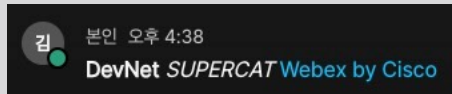
미션 내용



미션 #1

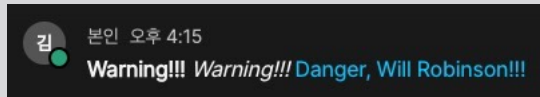
Webex 미팅 룸에 아래와 같은 마크다운 형식의 구문을 사용하여 스타일이 지정된 메시지 보내기

링크 : <https://www.webex.com/ko/index.html>



Hint :

****Warning!!!**** _Warning!!!_ [Danger, Will Robinson!!!](https://en.wikipedia.org/wiki/Lost_in_Space#Catchphrases)

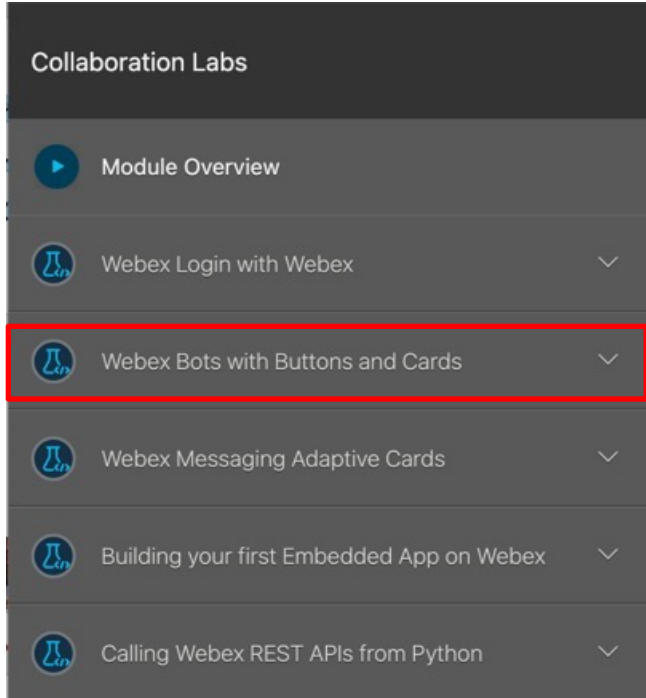


```
body = {'roomId': 'Y2lzY29zcGFyazredactedxMWU2LTg1OWEtOWYxODkzNjRkMGUx', 'text' 'Hello World!'}
```



Text 말고 어떤 게 들어가야 할까요?

Webex Coin bot 만들기



Webex Coin bot 만들기

GitHub 샘플 코드 복제

1 Sample Code Setup



<https://www.coingecko.com/en/api>

Collaboration Labs > Webex Bots with Buttons and Cards

120 mins

Your development environment will be timed out in 120 minutes. Any files created or updated during this time will be lost afterward. Please back up your files if needed.

Sample Code Setup

Talk to us

To get started we are going to clone an existing code sample available on GitHub.

This example implements a bot using the Webex Bot Node.js Framework to respond to events in the Webex client. A bot can only access messages sent to it directly. In group spaces, bots must be @mentioned to access the message. In 1-to-1 spaces, a bot has access to all messages from the user. The [Webex Bot Node.js Framework](#) makes this process very easy to build.

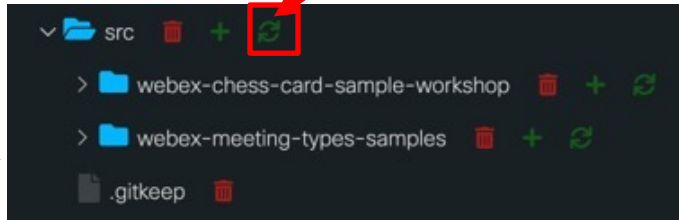
Cloning The GitHub Repo

clone the [GitHub repo](#) by executing the following command:

```
git clone https://github.com/WebexSamples/webex-bot-lab.git
```

120분동안 랩 이용 가능

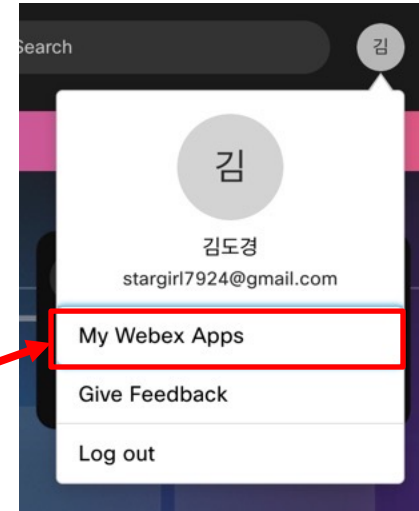
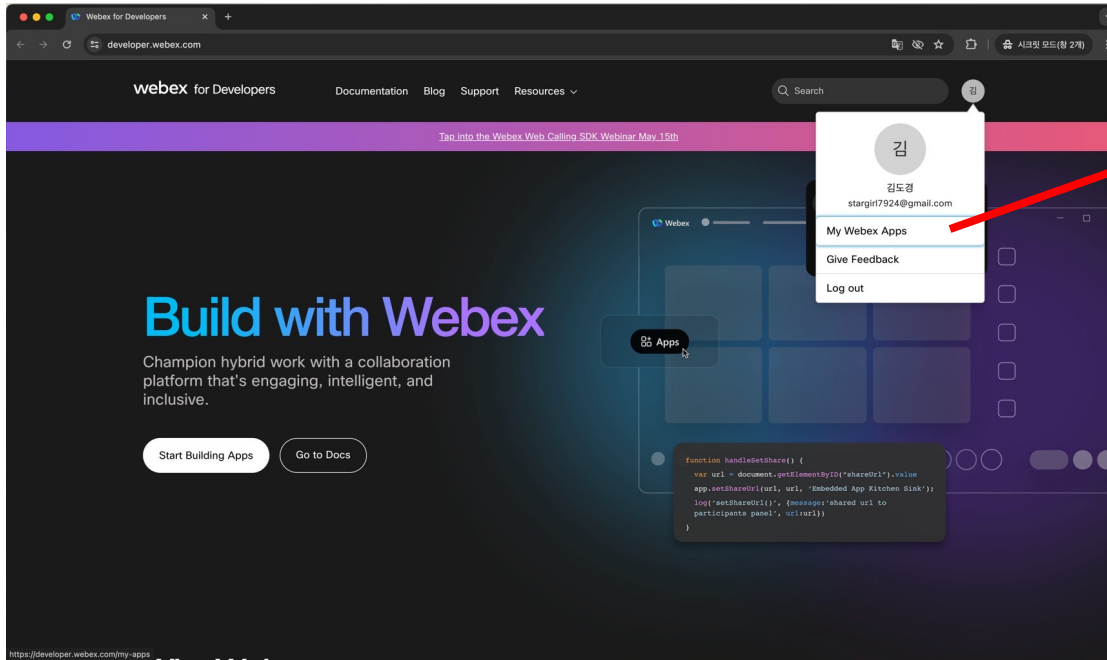
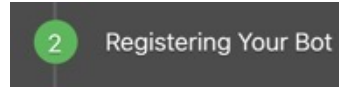
Github repo 가져온 후 /src 새로고침



Webex Coin bot 만들기

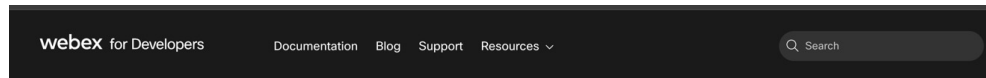
Webex 봇 생성하기

<https://developer.webex.com/>

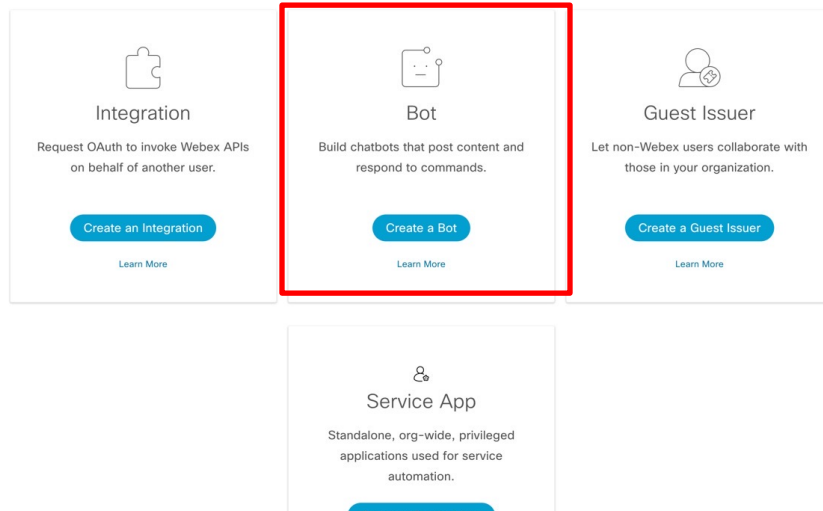


Webex Coin bot 만들기

Webex 봇 생성하기



Create a New App



Webex Coin bot

Webex 봇 생성하기

New Bot

Bot name*

Name of your bot as it will appear in Webex.

DevnetTestBot

봇 이름 띄어쓰기 X

Bot username*

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

devnet_test_bot1

@webex.bot

devnet_test_bot1@webex.bot is available

봇 유저 네임은 중복 불가

Icon*

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



Edit

아이콘 자유롭게 선택

App Hub Description*

What does your app do, how does it benefit users, how do users get started? Does your app require a non-Webex account? If your app is not free or has additional features for paid users, please note that and link to pricing information. 1024 character limit.

This is a test bot

Supported markdown

1482 characters remaining

설명 자유롭게 입력

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

Cancel

Add Bot

Webex Coin bot 만들기

Webex 봇 Access Token 저장

3

Responding To Events

Congratulations! 🎉

DevnetTestBot is one step closer to becoming a reality.

DevnetTestBot

👉 **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.

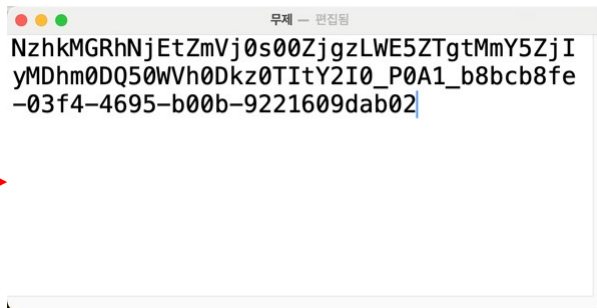
OTYwMTVmYmQtMDY3MS00NGM0LWExYjktNTUwZmU5MTkwYmQ

Copy Token

💡 **Tip:** Save this token!

It won't be shown again (but you can regenerate a new one if needed).

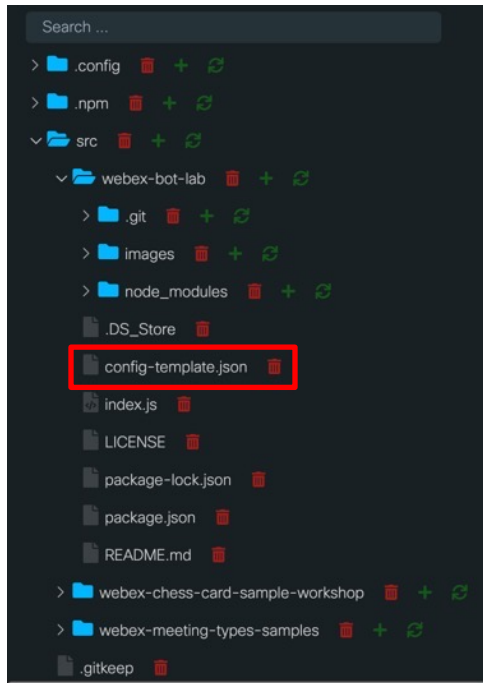
메모장에 Access Token 복사



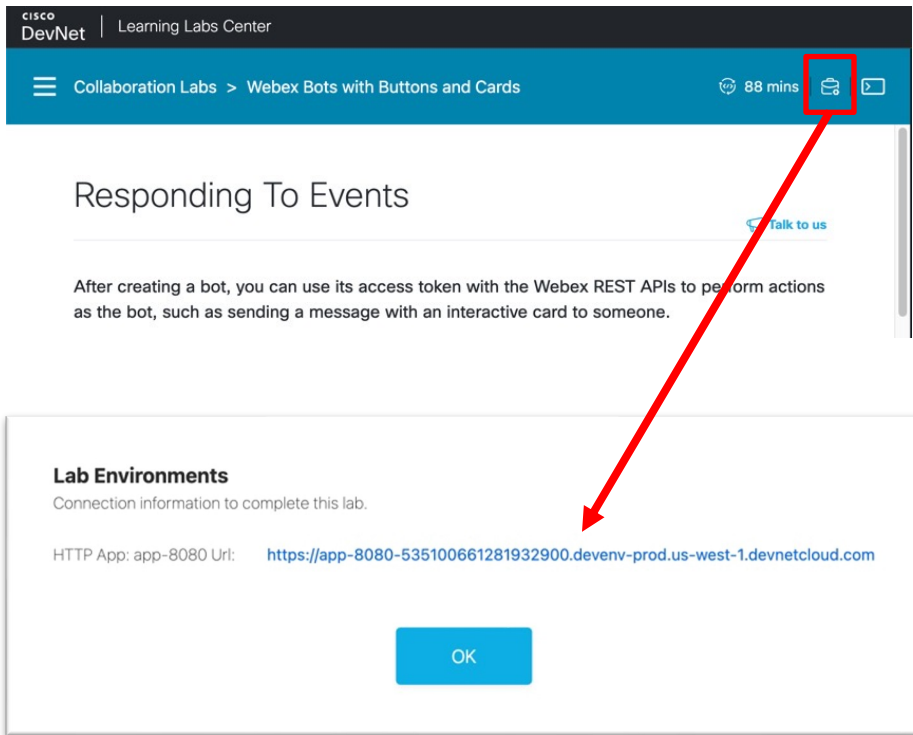
Webex Coin bot 만들기

이벤트 응답을 위한 웹훅 URL 가져오기

HTTP App: app-8080 Url 복사



/src/Webex-bot-lab/config-template.json



Webex Coin bot 만들기

Responding To Events

```
config-template.json
1 {
2   "webhookUrl": "REPLACE WITH URL FOR WEBHOSTING",
3   "token": "REPLACE WITH BOT ACCESS TOKEN",
4   "port": 8080
5 }
6
```

config-template.json에
HTTP App: app-8080 Url 붙여넣기
메모장에 복사했던 Access Token 붙여넣기

```
config-template.json
1 {
2   "webhookUrl": "https://app-8080-535100661281932900.devenv-prod.us-west-1.devnetcloud.com",
3   "token": "NzhkMGRhNjEtZmVjOS00ZjgzLWE5ZTgtMmY5ZjIyMDhmODQ5OWVhODkzOTItY2I0_P0A1_b8bcb8fe-03f4-4695-b00b-9221609dab02",
4   "port": 8080
5 }
6
```

Webex Coin bot 만들기

Coin bot 구동 시키기

Interacting With Our Bot

To get started with interacting with our bot, let's navigate to the sample code repo directory by executing the following command :

```
cd webex-bot-lab/
```



Then we will start our bot server.

```
npm start
```



디렉터리 이동 후 봇 서버 구동

```
Terminal x + A- A+
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 -)

added 1 package from 3 contributors, removed 610 packages and updated 336 packages in 27.378s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

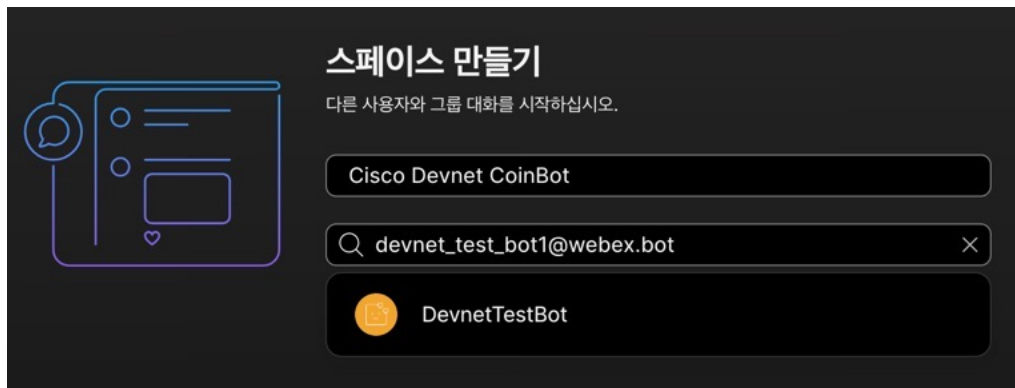
LoggerProxy->info#NO_LOGGER_DEFINED
Starting framework, please wait...
framework listening on port : 8080
framework is all fired up! [Press CTRL-C to quit]
█
```

봇 서버 구동 확인

Webex Coin bot 만들기

Webex 스페이스 생성

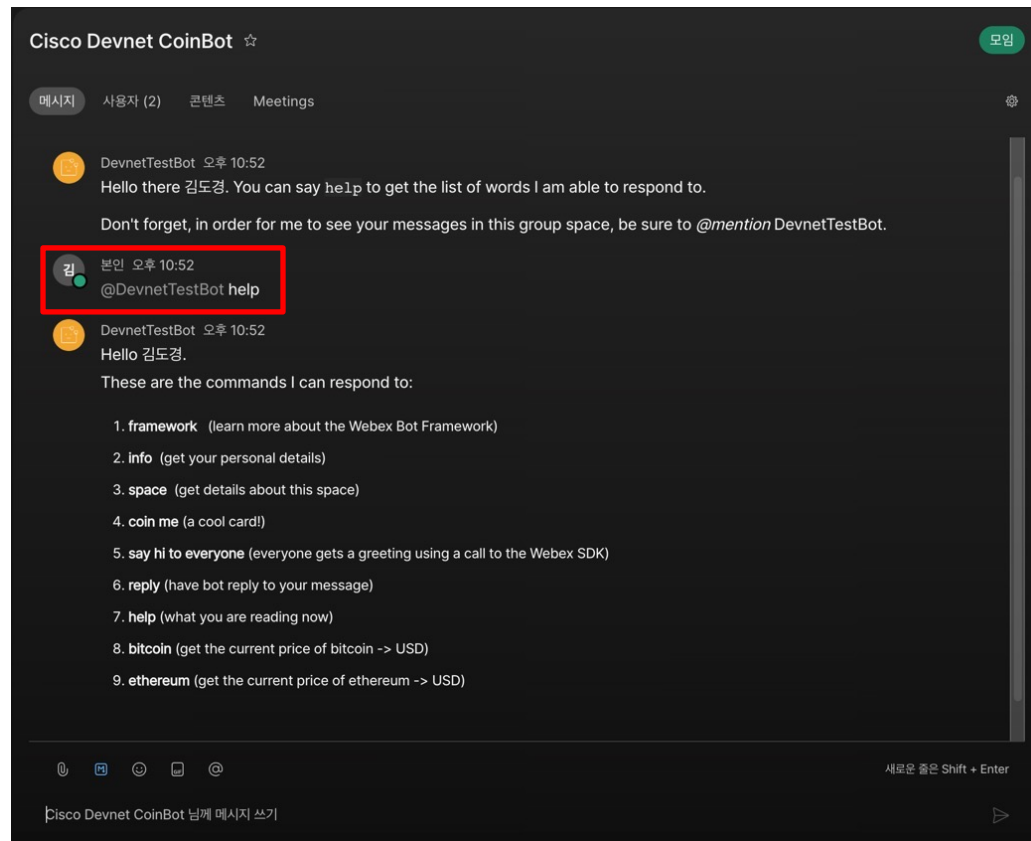
<https://web.webex.com/spaces>



봇을 초대하고 스페이스 생성

Webex Coin bot 만들기

봇 태그 후 help 입력

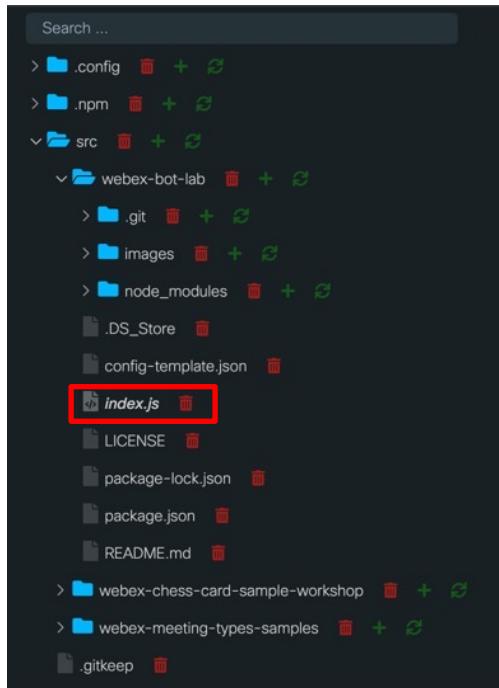


Webex Coin bot 만들기

Webex Bot Node.js Framework

4

Webex Bot Node.js Framework



Framework 시작

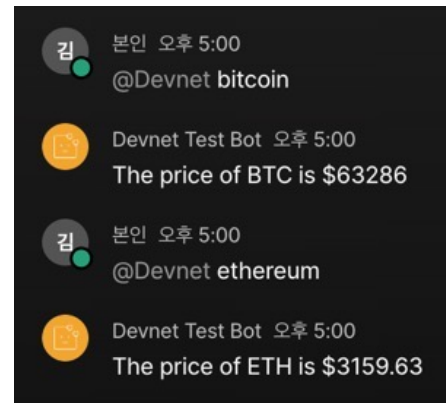
```
12 // init framework
13 var framework = new framework(config);
14 framework.start();
15 console.log("Starting framework, please wait...");
16
17 framework.on("initialized", function () {
18   console.log("framework is all fired up! [Press CTRL-C to quit]");
19 });
```

봇이 특정 이벤트/메세지를 수신하도록 하는 데

```
128 /* On mention with command
129 ex User enters @botname help, the bot will write back in markdown
130 */
131 framework.hears("help", function (bot, trigger) {
132   console.log('someone needs help! They asked ${trigger.text}');
133   bot.say('Hello ${trigger.person.displayName}.');
134   .then(() => sendHelp(bot))
135   .catch((e) => console.error('Problem in help handler: ${e.message}'));
136 });
```

사용자가 봇으로 메세지를 띄우는 데 사용

```
// Simple example
framework.hears('hello', (bot, trigger) -> {
  bot.say('hello')
});
```



/src/Webex-bot-lab/index.js



- “Coingecko API” line 65~89 : bitcoin
line 102~126 : ethereum

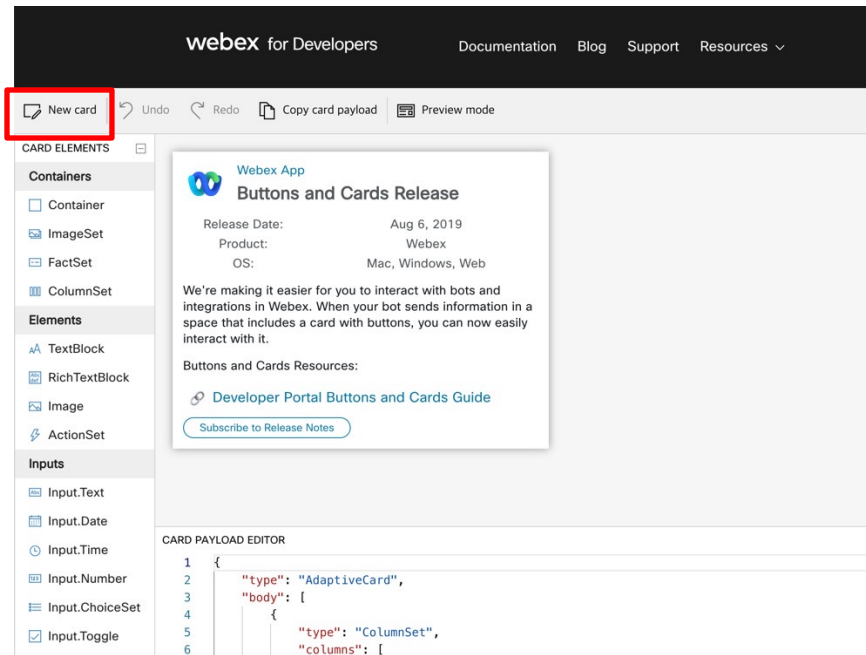
Webex Bot Node.js 프레임워크 활용

Webex Coin bot 만들기

Webex Adaptive Cards

<https://developer.webex.com/buttons-and-cards-designer>

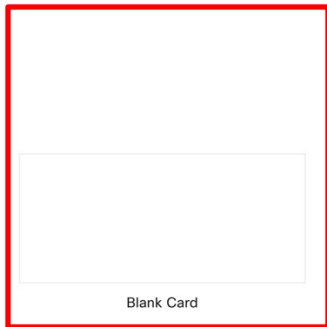
5 Webex Adaptive Cards



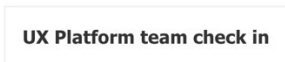
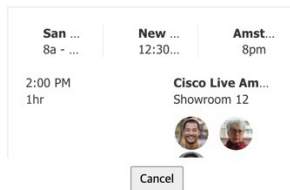
Webex Coin bot 만들기

Create -> Blank Card 선택

Create



Explore



Webex Coin bot 만들기

RichTextBlock 드래그

The screenshot shows the Webex bot builder interface. On the left, the 'CARD ELEMENTS' panel is visible, with 'Containers' and 'Elements' sections. In the 'Elements' section, 'RichTextBlock' is highlighted with a red box. A red arrow points from this box to a dashed box in the main workspace labeled 'Empty AdaptiveCard'. Below the workspace, the 'CARD PAYLOAD EDITOR' shows a JSON payload for an AdaptiveCard. The payload is as follows:

```
1 {
2   "type": "AdaptiveCard",
3   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
4   "version": "1.3",
5   "body": []
6 }
```

“text”: “Select a Coin...” 으로 변경

The screenshot shows the 'CARD PAYLOAD EDITOR' with a JSON payload for a RichTextBlock. The payload is as follows:

```
1 {
2   "type": "AdaptiveCard",
3   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
4   "version": "1.3",
5   "body": [
6     {
7       "type": "RichTextBlock",
8       "inlines": [
9         {
10          "type": "TextRun",
11          "text": "Select a Coin..."
12        }
13      ]
14    }
15  ]
16 }
```

Webex Coin bot 만들기

Input.ChoiceSet 드래그

The screenshot shows the Webex Bot Builder interface. On the left, the 'Inputs' panel lists various input types, with 'Input.ChoiceSet' highlighted and a red box around it. A red arrow points from this element to a 'Select a Coin...' input field in the 'Containers' area, which is also highlighted with a red box. The 'CARD PAYLOAD EDITOR' at the bottom shows the JSON payload for the card, including the 'Input.ChoiceSet' configuration.

```
1 {
2   "type": "AdaptiveCard",
3   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
4   "version": "1.3",
5   "body": [
6     {
7       "type": "RichTextBlock",
8       "inlines": [
9         {
10          "type": "TextRun",
11          "text": "Select a Coin..."
12        }
13      ]
14    }
15  ]
16 }
```

Id, Placeholder, Choices 변경

The screenshot shows the 'ELEMENT PROPERTIES' panel for 'Input.ChoiceSet'. The 'Id' is set to 'currency', the 'Placeholder' is set to 'Bitcoin', and the 'Choices' are listed as 'Bitcoin', 'bitcoin', 'Ethereum', and 'ethereum'. The 'Initially visible' checkbox is checked, and the 'Style' is set to 'Compact'.

Input.ChoiceSet

- Data context: (not set)
- Only show when: (not set)
- Id: currency
- Initially visible: ☒
- Label: (not set)
- Placeholder: Bitcoin
- Allow multi selection: ☐
- Style: Compact
- Default value: (not set)

Choices

- Bitcoin
- bitcoin
- Ethereum
- ethereum

Webex Coin bot 만들기

버튼에 따른 봇의 동작

```
212 // Process an Action.Submit button press
213 framework.on('attachmentAction', function (bot, trigger) {
214   if (trigger.type !== 'attachmentAction') {
215     throw new Error(`Invalid trigger type: ${trigger.type} in attachmentAction handler`);
216   }
217   let attachmentAction = trigger.attachmentAction;
218   console.log(attachmentAction);
219   if (attachmentAction.inputs.currency === 'ethereum') {
220     get_ethereum(bot);
221   } else {
222     get_bitcoin(bot);
223   }
224 });
```

Submit 버튼을 클릭했을 때

get_ethereum 함수 호출

get_bitcoin 함수 호출

Webex Coin bot 만들기

ActionSet 드래그

Webex Bot Builder interface showing the 'Containers' and 'Elements' panels. A red box highlights 'ActionSet' in the 'Elements' panel, and a red arrow points from it to a 'Select a Coin...' container in the main workspace. The 'Containers' panel shows a 'Bitcoin' dropdown. The 'CARD PAYLOAD EDITOR' at the bottom shows the JSON payload for the card.

```
1 {
2   "type": "AdaptiveCard",
3   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
4   "version": "1.3",
5   "body": [
```

Action.Submit 선택 후 title 변경

Webex Bot Builder interface showing the 'Add an action' menu. The 'Action.Submit' option is highlighted with a red box. The menu also shows 'Action.OpenUrl', 'Action.ShowCard', and 'Action.ToggleVisibility'.

```
30 ],
31 "actions": [
32   {
33     "type": "Action.Submit",
34     "title": "Submit"
35   }
36 ]
37 }
```

Webex Coin bot 만들기

CARD PAYLOAD EDITOR 코드 복사

CARD PAYLOAD EDITOR

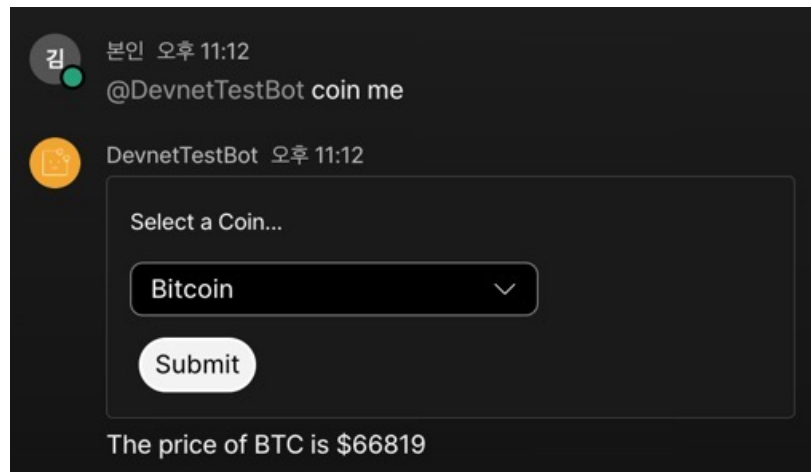
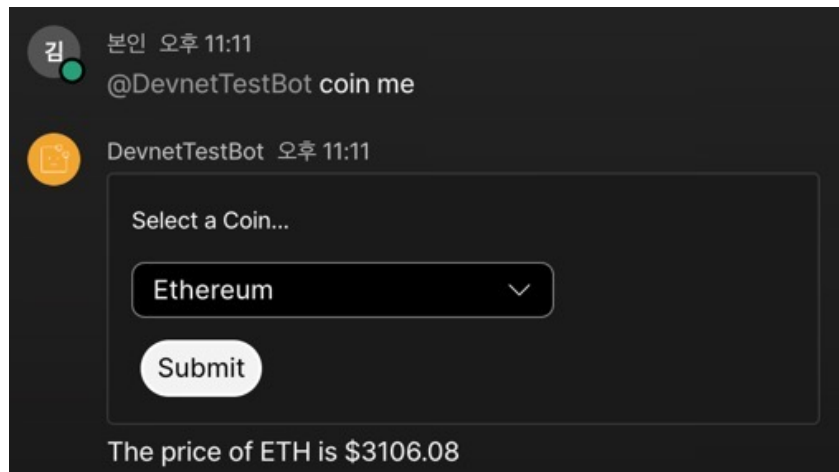
```
1  {
2    "type": "AdaptiveCard",
3    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
4    "version": "1.3",
5    "body": [
6      {
7        "type": "RichTextBlock",
8        "inlines": [
9          {
10             "type": "TextRun",
11             "text": "Select a Coin..."
12           }
13        ]
14      },
15      {
16        "type": "Input.ChoiceSet",
17        "choices": [
18          {
19            "title": "Bitcoin",
20            "value": "bitcoin"
21          }
22        ]
23      }
24    ]
25  }
```

index.js의 202 라인에 복사한 코드 붙여넣기

```
index.js
201 // Buttons & Cards data
202 let coinCard =
203 {
204   "type": "AdaptiveCard",
205   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
206   "version": "1.3",
207   "body": [
208     {
209       "type": "RichTextBlock",
210       "inlines": [
211         {
212           "type": "TextRun",
213           "text": "Select a Coin..."
214         }
215       ]
216     },
217     {
218       "type": "Input.ChoiceSet",
219       "choices": [
220         {
221           "title": "Bitcoin",
222           "value": "bitcoin"
223         }
224       ]
225     }
226   ]
227 }
```


Webex Coin bot 만들기

Coin bot 확인



미션!



미션 내용



미션 #2

저녁약속 봇 만들기

- 봇 태그 후 dinner plan 입력 시 카드 출력
- 예, 아니오, 미정 선택 후 Submit 버튼 클릭 시 각각 참석, 불참, 미정 메시지 출력

Hint 1 :

봇이 '저녁식사'라는 메시지를 출력하게 하려면
`bot.say('저녁식사');`

김 본인 어제, 오후 5:04
 @Devnet dinner plan

Devnet Test Bot 어제, 오후 5:04

저녁약속

연도, 월, 일.

-- --:--

☐ 삼겹살

☐ 샤브샤브

☐ 조개구이

참석 여부

Submit

참석
 미정
 불참

Devnet Test Bot 오후 5:05

저녁약속

2024. 05. 23.

오후 06:00

☐ 삼겹살

☒ 샤브샤브

☐ 조개구이

미정

예
 아니오
 미정

미션 내용



Hint 2 :

실습 때 let Coincard = 뒤에 작성한 코드 지우고
새 카드 만들어서 미션 진행

```

200
201 // Buttons & Cards data
202 let coinCard =
203
204 /* On mention with coin me
205 ex User enters @botname 'coin me' phrase, the bot will
206 */
207 framework.hears('coin me', function (bot, trigger) {
208   console.log("someone asked for a card");
209   bot.sendCard(coinCard, 'This is customizable fallback
210 });
211
  
```

Hint 3 :

Index.js 코드 일부 변경

```

index.js
206 */
207 framework.hears(' ', function (bot, trigger) {
208   console.log("someone asked for a card");
209   bot.sendCard(coinCard, 'This is customizable fallback text for clients that do not support buttons & cards');
210 });
211
212 // Process an Action.Submit button press
213 framework.on('attachmentAction', function (bot, trigger) {
214   if (trigger.type !== 'attachmentAction') {
215     throw new Error(`Invalid trigger type: ${trigger.type} in attachmentAction handler`);
216   }
217   let attachmentAction = trigger.attachmentAction;
218   console.log(attachmentAction);
219   if (attachmentAction.inputs.currency === ' ') {
220     bot.say(' ');
221   } else if (attachmentAction.inputs.currency === ' ') {
222     bot.say(' ');
223   } else {
224     bot.say(' ');
225   }
226 });
  
```

Summary

- Calling Webex REST APIs from Python
 - GET : Webex Space의 스크립트 가져오기
 - POST : Webex Space에 자동 메시지 보내기
- Webex Bots
 - Webex Bot Node.js 프레임워크 활용
 - Webex Button and Cards Designer 활용



Thank you

