

Raspberry Pi를 활용한 IoT 프로젝트

나만의 가상비서 만들기(구글 어시스턴트 & 스마트미러)

5일차

담당교수 : 조도은

<https://github.com/DoEunCho/raspberrypi>



1일차 : 라즈베리파이 소개와 환경 구축(3H)

2일차 : 라즈베리 파이를 위한 리눅스 기초 배우기(3H)

3일차 : 파이썬 기초 명령어 익히기(3H)

4일차 : 라즈베리 파이 GPIO와 센서 동작하기(3H)

5일차 : 나만의 가상비서 만들기(구글 어시스턴트)(3H)



- **인공지능 비서, AI 어시스턴트 & 스마트 미래**

- 하드웨어 및 네트워크 액세스 설정
- 오디오 마이크 설정 및 테스트하기
- 개발자 프로젝트 및 계정 설정 구성
- MagicMirror 설치하기
- 구글 어시스턴트 사용자 인증
- 구글 어시스턴트 모듈 설치
- 나만의 프로젝트 구상하기



- 필수 하드웨어

- 라즈베리 파이 4 모델 B 및 전원 공급 장치(권장 사양: 3B 이상)
- USB 마이크
- USB 스피커
- 라즈비안 OS 설치된 SD 카드
- USB 케이블, USB 마우스 및 HDMI 케이블이 있는 모니터



- 하드웨어 연결 및 네트워크 액세스 구성

- 마이크와 스피커를 Raspberry Pi에 연결
- SD 카드를 Raspberry Pi에 삽입
- 이더넷 케이블을 연결하거나 Wi-Fi 네트워크에 연결

- 날짜 및 시간 구성

- \$ **sudo apt-get install rdate**

※참고

날짜 또는 시간이 올바르지 않으면 SSL 오류가 발생할 수 있습니다.
따라서 시간 서버와 동기화 해주는 작업이 필요합니다.

시간이 안맞는 경우 명령 실행

```
$ sudo apt-get install rdate  
$ sudo rdate -s time.bora.net  
$ sudo reboot
```



■ 스피커 연결

- USB 단자와 스피커 잭을 연결



■ 마이크 연결

- USB 단자에 연결





마이크와 스피커 Card & Device number 확인

~\$ **sudo raspi-config**

```
Raspberry Pi Software Configuration Tool (raspi-config)
```

1 System Options	Configure system settings
2 Display Options	Configure display settings
3 Interface Options	Configure connections to peripherals
4 Performance Options	Configure performance settings
5 Localisation Options	Configure language and regional settings
6 Advanced Options	Configure advanced settings
8 Update	Update this tool to the latest version
9 About raspi-config	Information about this configuration tool

<Select> <Finish>

1번 선택 -> S2 Audio -> 0 **Headphone** <확인>



마이크와 스피커 Card & Device number 확인

- 녹음 및 재생 장치 찾기

- 캡처 하드웨어 장치 목록에서 **USB 마이크** 찾기

\$ **arecord -l** # 카드번호, 장치번호 기록

```
pi@raspberrypi:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

- 재생 하드웨어 장치 목록에서 **스피커** 찾기

\$ **aplay -l** # 카드번호, 장치번호 기록

```
pi@raspberrypi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: Headphones [bcm2835 Headphones], device 0: bcm2835 Headphones [bcm2835 Headphones]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
```




~\$ nano ~/.asoundrc

```
pcm.!default {
    type asym
    capture.pcm "mic"
    playback.pcm "speaker"
}
pcm.mic {
    type plug
    slave {
        pcm "hw:1,0"
    }
}
pcm.speaker {
    type plug
    slave {
        pcm "hw:0,0"
    }
}
```

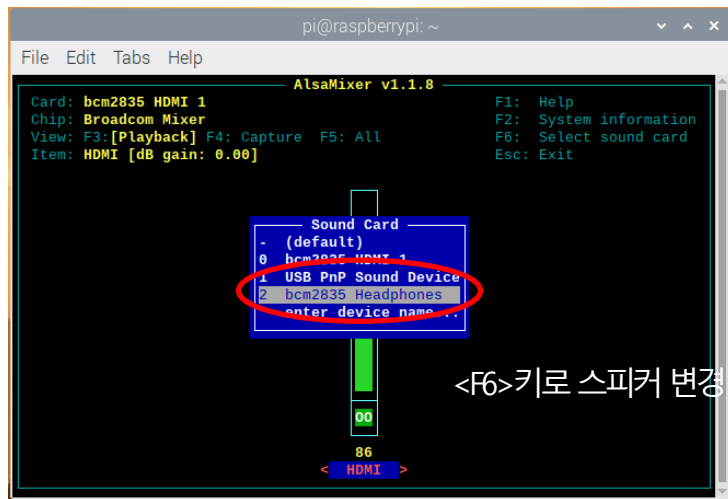
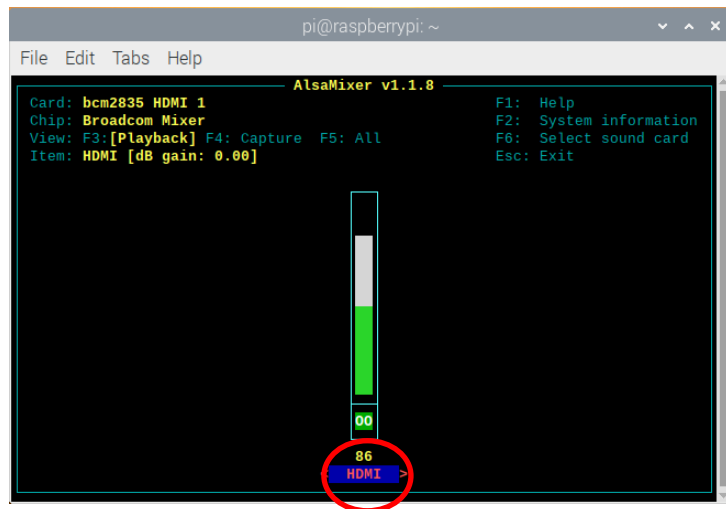


오디오 구성 및 테스트

- 녹음 및 재생 확인

- 재생 음량 조절

\$ sudo **alsamixer** # 위쪽 화살표 키를 눌러 재생 볼륨 레벨을 약 70으로 설정



<F6>키로 스피커 변경



오디오 구성 및 테스트

- 스피커 소리 테스트하기(중지 **Ctrl +C**)
\$ **speaker-test -t wav**
- 소리가 안나오는 경우 환경설정에서 변경
\$ **sudo raspi-config**

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ gedit .asoundrc  
pi@raspberrypi:~ $ alsamixer  
pi@raspberrypi:~ $ speaker-test -t wav  
  
speaker-test 1.1.8  
  
Playback device is default  
Stream parameters are 48000Hz, S16_LE, 1 channels  
WAV file(s)  
Rate set to 48000Hz (requested 48000Hz)  
Buffer size range from 512 to 65536  
Period size range from 512 to 65536  
Using max buffer size 65536  
Periods = 4  
was set period_size = 16384  
was set buffer_size = 65536  
  0 - Front Left  
Time per period = 0.360085  
  0 - Front Left  
Time per period = 1.371370  
  0 - Front Left  
^CTransfer failed: Bad address  
pi@raspberrypi:~ $
```



오디오 구성 및 테스트

- 마이크 녹음 및 재생 확인

- 짧은 오디오 클립을 녹음

```
$ arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw  
out.raw
```

- 녹음 재생

```
$ aplay --format=S16_LE --rate=16000 out.raw
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ arecord --format=S16_LE --duration=5 --rate=16000 --file-type=  
=raw out.raw  
Recording raw data 'out.raw' : Signed 16 bit Little Endian, Rate 16000 Hz, Mono  
pi@raspberrypi:~ $ aplay --format=S16_LE --rate=16000 out.raw  
Playing raw data 'out.raw' : Signed 16 bit Little Endian, Rate 16000 Hz, Mono  
pi@raspberrypi:~ $
```



MagicMirror 설치 명령어 실행

\$ bash -c "\$(curl -sL https://raw.githubusercontent.com/sdetweil/MagicMirror_scripts/master/raspberry.sh)"

```

의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
nodejs is already the newest version (16.15.1-deb-1nodesource1).
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
  libfuse2
Use 'sudo apt autoremove' to remove it.
0개 업그레이드, 0개 새로 설치, 0개 제거 및 1개 업그레이드 안 함.
Node.js installation Done! version=v16.15.1
Check current NPM installation ...
NPM currently installed. Checking version number.
Minimum npm version: V7.11.2
Installed npm version: V8.11.0
No npm upgrade necessary.
Cloning MagicMirror ...
'MagicMirror'에 복제합니다...
remote: Enumerating objects: 327, done.
remote: Counting objects: 100% (327/327), done.
remote: Compressing objects: 100% (255/255), done.
remote: Total 327 (delta 66), reused 166 (delta 58), pack-reused 0
오브젝트를 받는 중: 100% (327/327), 446.80 KiB | 1.02 MiB/s, 완료.
델타를 알아내는 중: 100% (66/66), 완료.
Cloning MagicMirror Done!
Installing dependencies ...
( ) : idealTree:serialize-error: sill placeDep ROOT type-fest@
  
```

MagicMirro 설치과정 화면

```

오브젝트를 받는 중: 100% (327/327), 446.80 KiB | 1.02 MiB/s, 완료.
델타를 알아내는 중: 100% (66/66), 완료.
Cloning MagicMirror Done!
Installing dependencies ...
npm notice
npm notice New minor version of npm available! 8.11.0 -> 8.13.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.13.2
npm notice Run npm install -g npm@8.13.2 to update!
npm notice
Dependencies installation Done!
setting up initial config.js
Check plymouth installation ...
Splashscreen: checking themes directory.
Splashscreen: Create theme directory if not exists.
Splashscreen: Changed theme to MagicMirror successfully.
Do you want use pm2 for auto starting of your MagicMirror (y/N)?n
Do you want to disable the screen saver? (y/N)?y

We're ready! Run DISPLAY=:0 npm start from the ~/MagicMirror directory to start
your MagicMirror.

pi@raspberrypi:~$
  
```

화면꺼짐방지(Y)
자동실행(N)

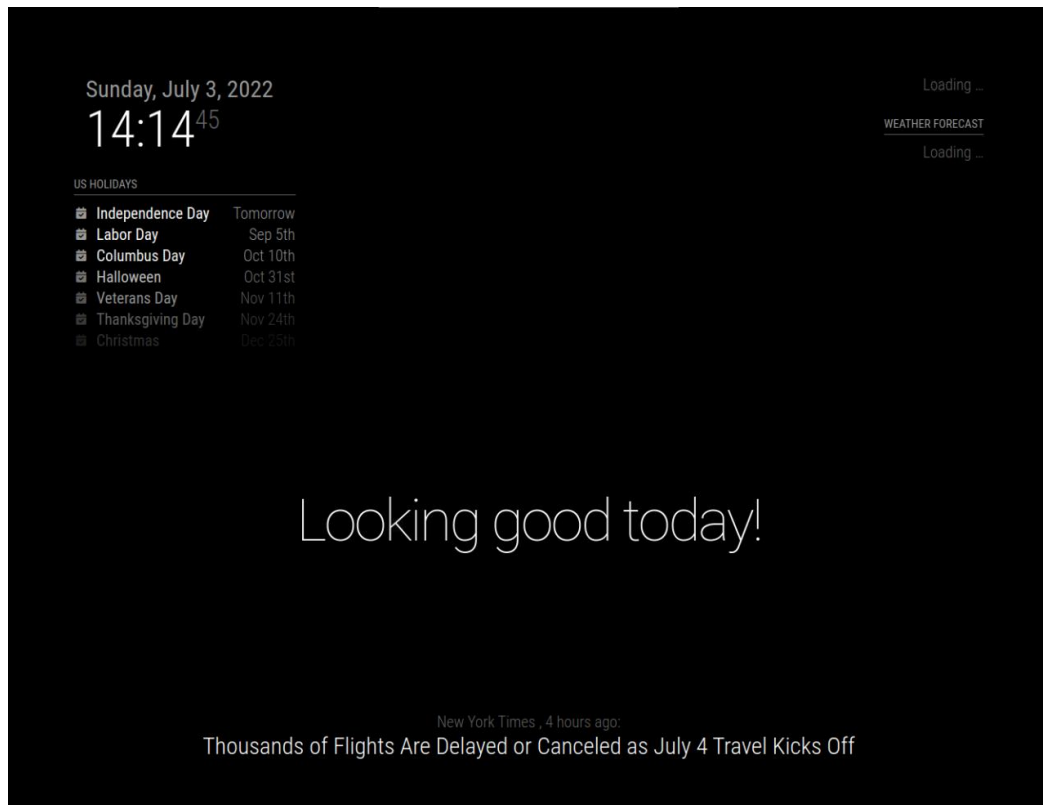
MagicMirror 설치완료 화면



MagicMirror 실행

```
$ cd MagicMirror
```

```
$ npm run start
```





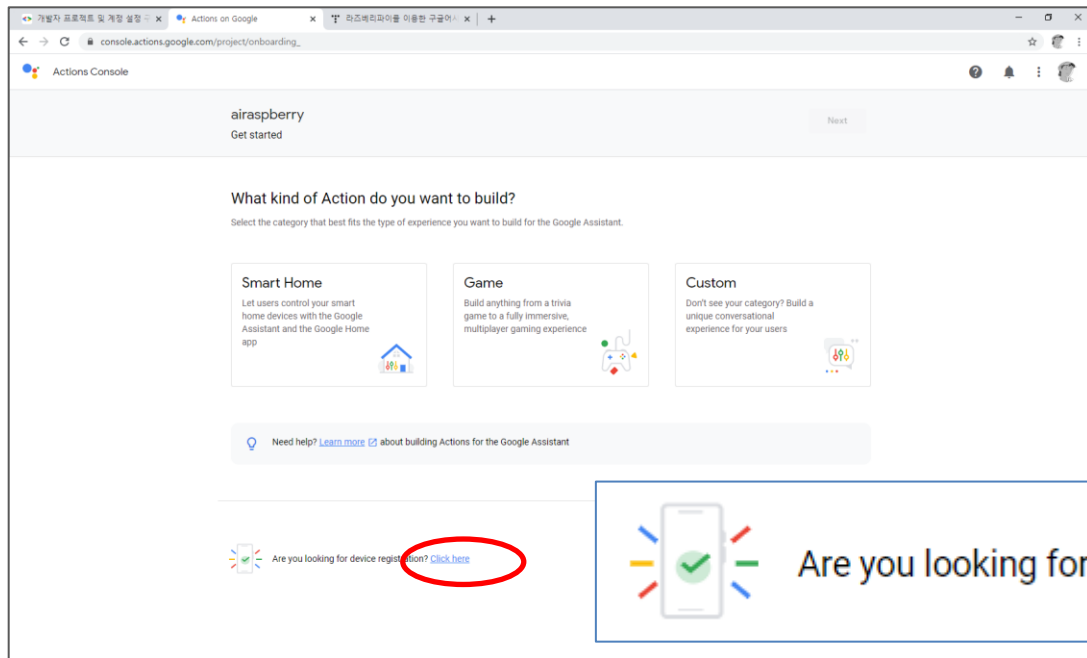
개발자 프로젝트 및 계정 구성 설정

- **Actions Console 프로젝트 구성**

- 작업 콘솔을 열기 (<https://console.actions.google.com/>)
- 새 프로젝트를 만들려면 프로젝트 이름 입력

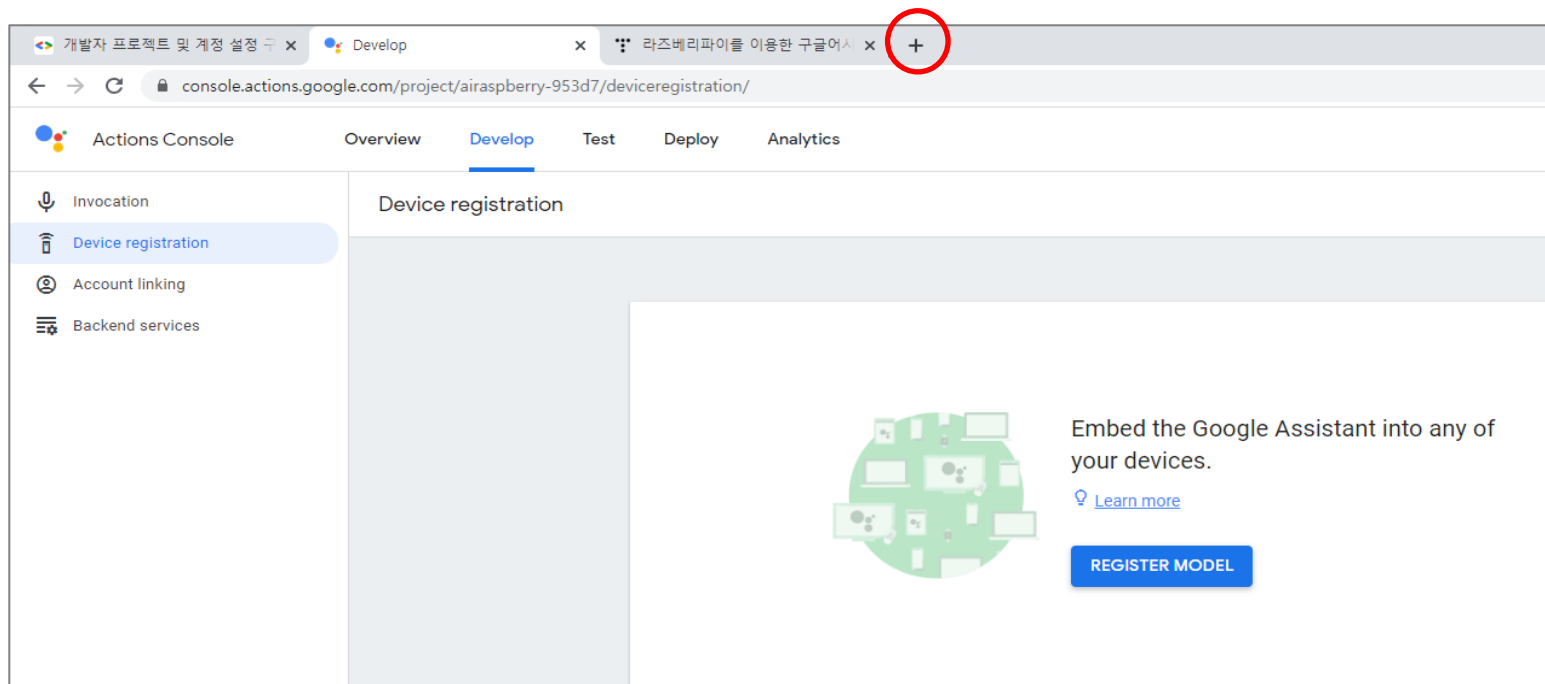


개발자 프로젝트 및 계정 구성 설정



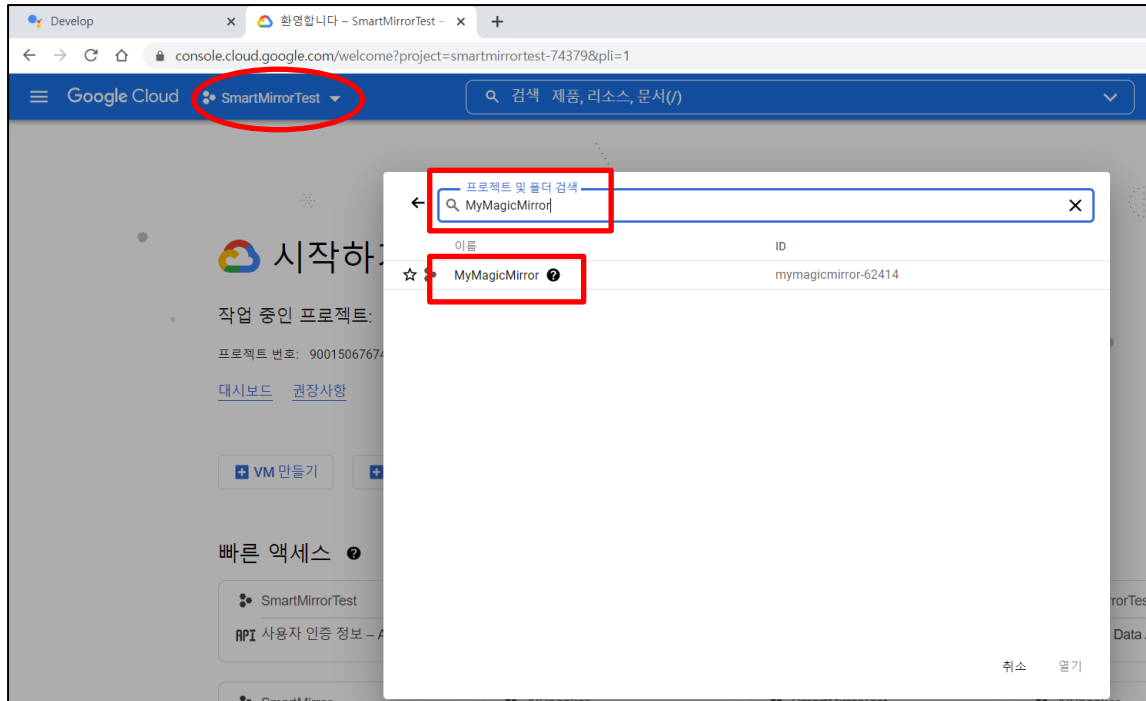


클릭 후 뜨는 창 그대로 두고 [새탭] 클릭



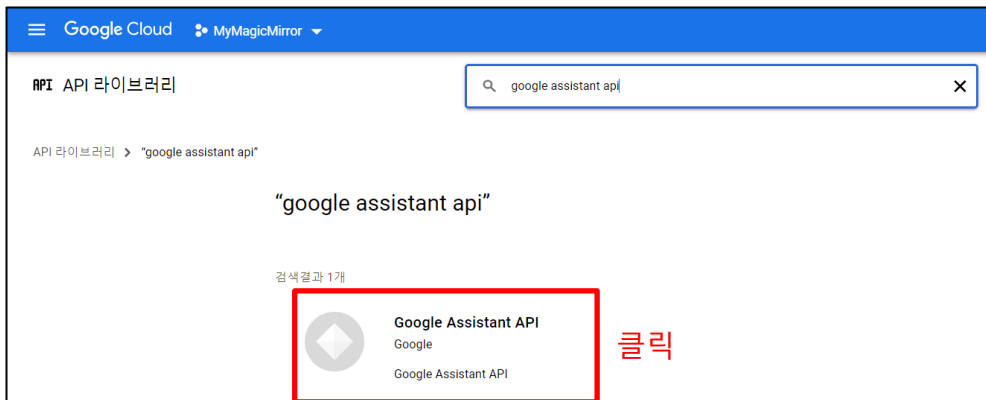
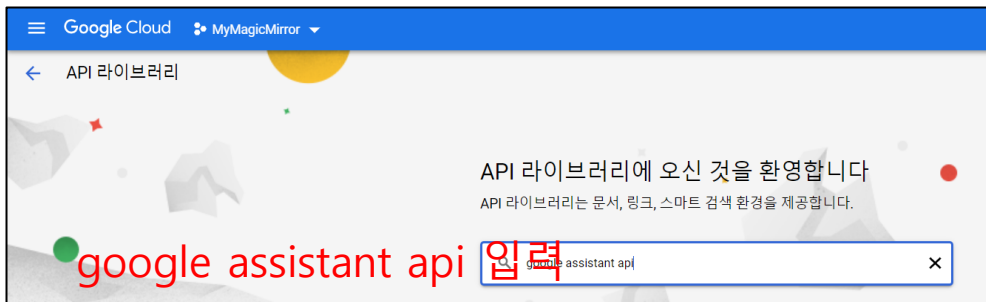
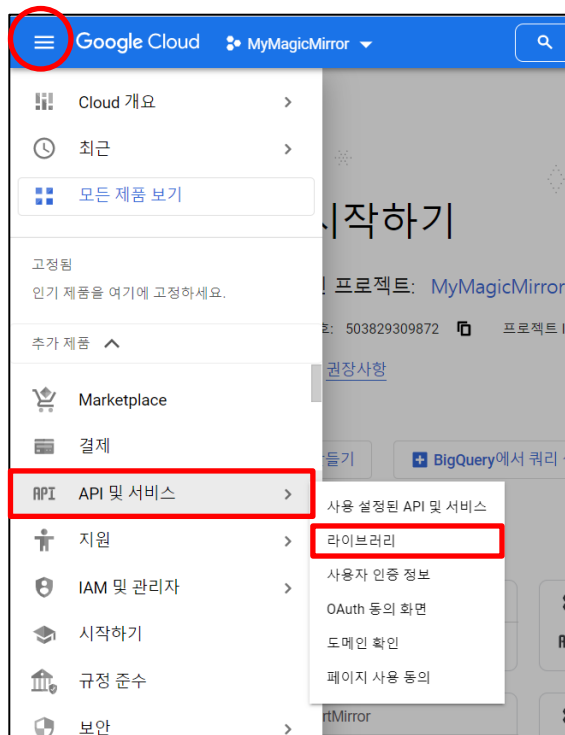


- <https://console.cloud.google.com/>
- 방금 작성한 프로젝트 선택





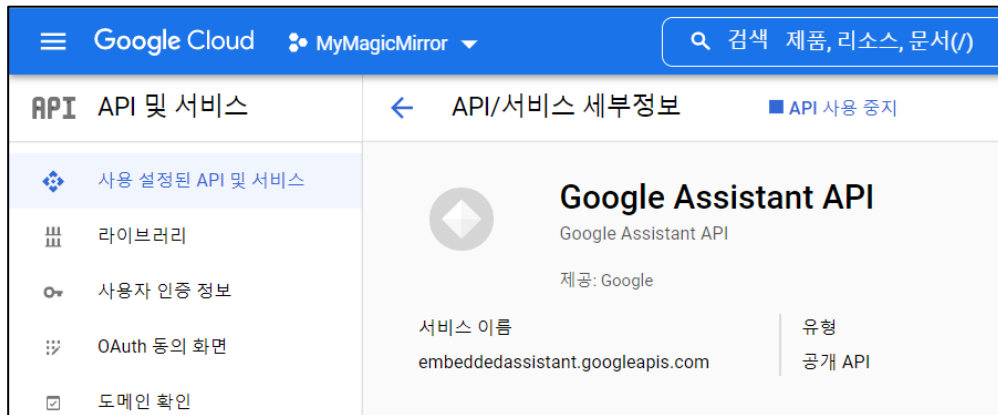
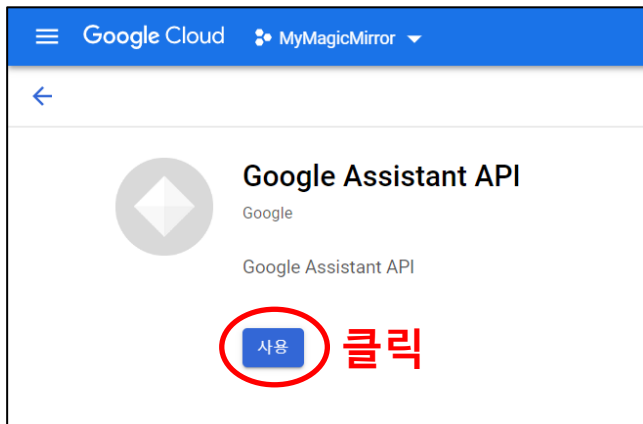
API 라이브러리 사용 설정하기





개발자 프로젝트 및 계정 구성 설정

- **Actions Console 프로젝트 구성**
 - 선택한 프로젝트에서 Google Assistant API 사용 설정
 - 추가 라이브러리 사용 설정 동일함





개발자 프로젝트 및 계정 구성 설정

OAuth 동의 화면 작성하기(1)

Google Cloud

MyMagicMirror

검색 제품, 리소스, 문서(/)

API	API 및 서비스	OAuth 동의 화면
	사용 설정된 API 및 서비스	
	라이브러리	
	사용자 인증 정보	
	OAuth 동의 화면	
	도메인 확인	
	페이지 사용 동의	

대상 사용자를 비롯해 앱을 구성하고 등록하려는 방식을 선택하세요. 프로젝트에는 하나의 앱만 연결할 수 있습니다.

User Type

☐ 내부 ⓘ

조직 내 사용자만 사용할 수 있습니다. 인증을 위해 앱을 제출할 필요는 없습니다. [사용자 유형 자세히 알아보기](#)

☒ 외부 ⓘ

Google 계정이 있는 모든 테스트 사용자가 사용할 수 있습니다. 앱이 테스트 모드로 시작되고 테스트 사용자 목록에 추가된 사용자에게만 제공됩니다. 앱을 프로젝트에 푸시할 준비가 되면 앱을 인증해야 할 수도 있습니다. [사용자 유형 자세히 알아보기](#)

OAuth 경험에 대한 [의견 보내기](#)



개발자 프로젝트 및 계정 구성 설정

OAuth 동의 화면 작성하기(2)

Google Cloud

MyMagicMirror

검색 제품, 리소스, 문서(/)

API 및 서비스

사용 설정된 API 및 서비스

라이브러리

사용자 인증 정보

OAuth 동의 화면

도메인 확인

페이지 사용 동의

앱 등록 수정

1 OAuth 동의 화면

2 범위

3 선택적 정보

4 요약

앱 정보

동의 화면에 표시되어 최종 사용자가 개발자를 확인하고 문의할 수 있습니다.

앱 이름 *

project-503829309872

동의를 요청하는 앱의 이름

사용자 지원 이메일 *

이메일 입력

사용자가 동의 관련 질문을 위해 문의할 때 이용합니다.

앱 로고

찾아보기

사용자가 앱을 알아보는 데 도움이 되도록 동의 화면에 대한 이미지(1MB 이하 크기)를 업로드합니다. 허용되는 이미지 형식은 JPG, PNG, BMP입니다. 최적의 결과를 위해서는 로고가 120x120픽셀 크기의 정사각형이어야 합니다.

승인된 도메인

동의 화면 또는 OAuth 클라이언트 구성에서 도메인이 사용되면 여기에서 사전 등록해야 합니다. 앱이 인증을 거쳐야 하는 경우 [Google Search Console](#)로 이동하여 도메인이 승인되었는지 확인하세요. 승인된 도메인 한도에 대해 [자세히 알아보세요](#).

승인된 도메인 1 *

mymagicmirror-6com

+

도메인 추가

개발자 연락처 정보

이메일 주소 *

이메일 입력

이 이메일 주소는 Google에서 프로젝트 변경사항에 대해 알림을 보내기 위한 용도입니다.

저장 후 계속

취소

다음 단계 계속 [저장 후 계속]



개발자 프로젝트 및 계정 구성 설정

OAuth 동의 화면 작성하기(3)

범위

API ↑	범위	사용자에게 표시되는 설명
표시할 행이 없습니다.		

테스트 사용자

사용자 0명(테스트 0명, 기타 0명)/사용자 한도100명

?

필터 속성 이름 또는 값 입력

사용자 정보

표시할 행이 없습니다.

대시보드로 돌아가기

OAuth 동의 화면

주기에서 계산됩니다. [자세히 알아보기](#)

사용자 0명(테스트 0명, 기타 0명)/사용자 한도100명

?

테스트 사용자

+ ADD USERS

필터 속성 이름 또는

사용자 정보

표시할 행이 없습니다.

×

사용자 추가

⚠

계시 상태가 '테스트'로 설정된 동안에는 테스트 사용자만 앱에 액세스할 수 있습니다. 앱 인증 전에 허용되는 사용자 한도는 100명이며 앱의 전체 수명 주기에서 계산됩니다.

LEARN MORE

접속중인 구글 이메일 주소 입력

1 / 100

저장



개발자 프로젝트 및 계정 구성 설정

사용자 인증 정보 만들기(1)

Google Cloud MyMagicMirror

검색 제품, 리소스, 문서(/)

API API 및 서비스

사용자 인증 정보

+ 사용자 인증 정보 만들기 삭제

사용 설정된 API에 액세스하려면

API 키

이름

Browser key (auto created by Google Service)

OAuth 2.0 클라이언트 ID

이름

New Actions on Google App

Web client (auto created by Google Service)

생성일

2022. 7. 3.

2022. 7. 3.

유형

웹 애플리케이션

웹 애플리케이션

사용 설정된 API 및 서비스

라이브러리

사용자 인증 정보

OAuth 동의 화면

도메인 확인

페이지 사용 동의

API 키

OAuth 클라이언트 ID

서비스 계정

사용자 인증 정보 선택 도움말



개발자 프로젝트 및 계정 구성 설정

사용자 인증 정보 만들기(2)

[←](#) OAuth 클라이언트 ID 만들기

클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 [OAuth 2.0 설정](#)을 참조하세요. OAuth 클라이언트 유형을 [자세히 알아보세요](#).

애플리케이션 유형 *

웹 애플리케이션 ▼

이름 *

MagicMirror

OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.

i

아래에 추가한 URI의 도메인이 [승인된 도메인](#)으로 [OAuth 동의 화면](#)에 자동
으로 추가됩니다.

승인된 리디렉션 URI **?**

웹 서버의 요청에 사용

URI 1 *

<https://googleassistant.bugsounet.fr>

+ URI 추가

참고: 설정이 적용되는 데 5분에서 몇 시간이 걸릴 수 있습니다.

만들기

취소



개발자 프로젝트 및 계정 구성 설정

사용자 인증 정보 만들기(3)

OAuth 클라이언트 생성됨

API 및 서비스의 사용자 인증 정보에서 언제든지 클라이언트 ID와 보안 비밀에 액세스할 수 있습니다.



OAuth 액세스는 [OAuth 동의 화면](#)에 나열된 [테스트 사용자](#)로 제한됩니다.

클라이언트 ID

503829309872-s9buurg2kou32t774ko6k7vgi47aeu0m.apps.gc



클라이언트 보안 비밀번호

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



↓ JSON 다운로드

확인

OAuth 2.0 클라이언트 ID

<input type="checkbox"/> 이름	생성일 ↓	유형	클라이언트 ID	
<input type="checkbox"/> MagicMirror	2022. 7. 3.	웹 애플리케이션	503829309872-s9bu...	
<input type="checkbox"/> New Actions on Google App	2022. 7. 3.	웹 애플리케이션	503829309872-qqq9...	
<input type="checkbox"/> Web client (auto created by Google Service)	2022. 7. 3.	웹 애플리케이션	503829309872-18np...	

JSON 파일을 다운로드하여 라즈베리파이에 저장



개발자 프로젝트 및 계정 구성 설정

사용자 인증 정보 만들기(4)

Google Cloud MyMagicMirror

검색 제품, 리소스

API	API 및 서비스	OAuth 동의 화면
⚙️	사용 설정된 API 및 서비스	project-503829309872 앱 수정
📖	라이브러리	
🔑	사용자 인증 정보	게시 상태 ?
📄	OAuth 동의 화면	테스트
🔍	도메인 확인	앱 게시
🔄	페이지 사용 동의	사용자 유형
		외부 ?
		내부로 설정

Google Cloud MyMagicMirror

검색 제품, 리소스, 문서(/)

API	API 및 서비스	OAuth 동의 화면
⚙️	사용 설정된 API 및 서비스	project-503829309872 앱 수정
📖	라이브러리	
🔑	사용자 인증 정보	확인 상태
📄	OAuth 동의 화면	확인이 필요하지 않음
🔍	도메인 확인	동의 화면이 표시되고 있지만 앱이 검토되지 않았으므로 특정 OAuth 범위를 요청할 수 없으며 사용자에게 일부 정보가 표시되지 않을 수 있습니다. 자세히 알아보기
🔄	페이지 사용 동의	게시 상태 ?
		프로덕션 단계
		테스트로 돌아가기



MMM-GoogleAssistant 모듈 설치

```
$ cd ~/MagicMirror/modules
```

```
$ git clone https://github.com/bugsounet/MMM-GoogleAssistant
```

```
$ cd MMM-GoogleAssistant
```

```
$ npm install
```



자격 증명 설정

- <https://console.cloud.google.com/> - API 및 서비스 - 사용자 인증정보에서 OAuth json 파일 다운로드
- JSON 파일을 credentials.json으로 이름을 변경하고, MMM-GoogleAssistant 폴더로 이동
- 터미널에서 npm run token 명령 실행 -> y입력

```
pi@raspberrypi: ~/MagicMirror...odules/MMM-GoogleAssistant
파일(F) 편집(E) 탭(T) 도움말(H)
found 0 vulnerabilities
pi@raspberrypi:~/MagicMirror/modules/MMM-GoogleAssistant $ cd MagicMirror/modules/MMM-GoogleAssistant
bash: cd: MagicMirror/modules/MMM-GoogleAssistant: 그런 파일이나 디렉터리가 없습니다
pi@raspberrypi:~/MagicMirror/modules/MMM-GoogleAssistant $ npm run token

> MMM-GoogleAssistant@4.0.5 token
> installer/token.sh

Welcome to MMM-GoogleAssistant Token generator!

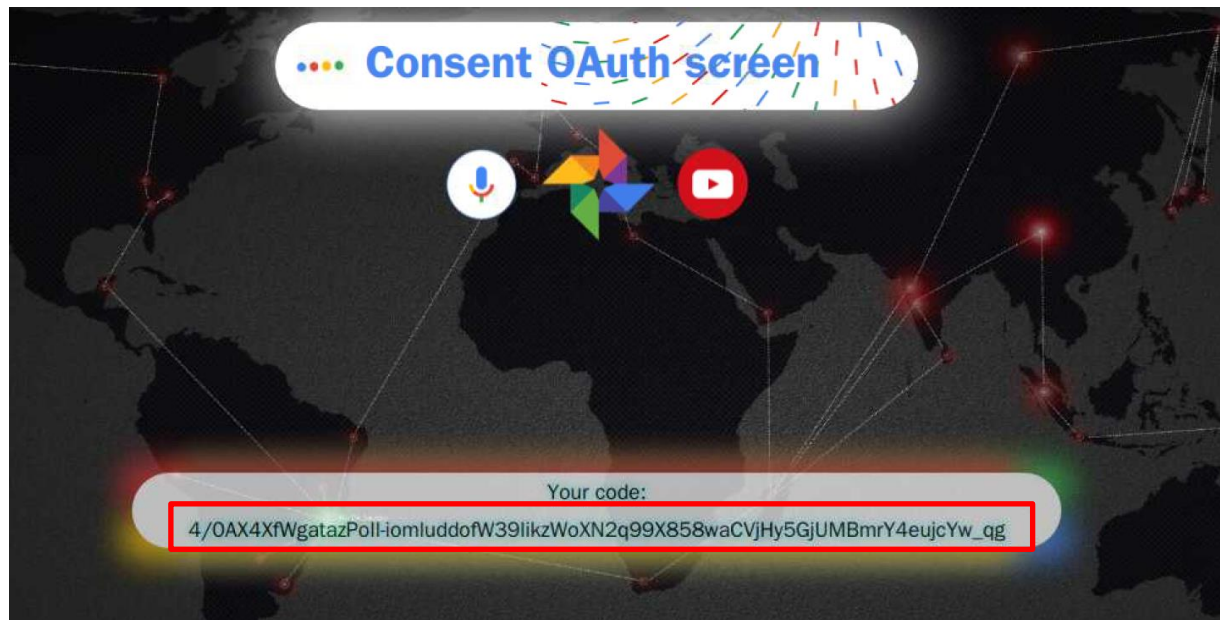
Do you want to install/reinstall MMM-GoogleAssistant token? [Y/n]
Your choice: 
```





자격 증명 설정

화면에 표시된 코드를 복사





자격 증명 설정

터미널에 복사한 코드를 붙여넣기(아래 화면처럼 표시되면 성공)

```
pi@raspberrypi: ~/MagicMirror...odules/MMM-GoogleAssistant
파일(F) 편집(E) 탭(T) 도움말(H)
otypedata=consent&response_type=code&client_id=503829309872-s9b
uurg2kou32t774ko6k7vgi47aeu0m.apps.googleusercontent.com&redirect_
uri=https%3A%2F%2Fgoogleassistant.bugsounet.fr
Paste your code: 4/0AX4XfwgatazPoIl-iomluddofW39likzWoXN2q99X858wa
CVjHy5GjUMBmrY4eujcYw_qg
[GA] Testing Conversation start...

[GA] Assistant Question: What time is it?
[GA] Assistant Response: It's 4:12 PM.
[GA] Conversation Complete

[GA] Testing Conversation ended.

Done.
pi@raspberrypi:~/MagicMirror/modules/MMM-GoogleAssistant $
```



MagicMirror에 구글 어시스턴트 모듈 설정

\$ **cd** ~ 로 홈 폴더로 이동

\$ **geany MagicMirror/config/config.js**

모듈리스트 맨 하단에 아래 코드 추가

```
{  
    module: "MMM-GoogleAssistant",  
    configDeepMerge: true,  
    config: {  
        assistantConfig: {  
            lang: "ko-KR",  
            latitude: 37.3460,  
            longitude: 126.550,  
        },  
    },  
},
```




MagicMirror에 구글 어시스턴트 모듈 설정

```
config.js x
94     position: "bottom_bar",
95     config: {
96       feeds: [
97         {
98           title: "New York Times",
99           url: "https://rss.nytimes.com/services/xml/rss/nyt/HomePage",
100         }
101       ],
102       showSourceTitle: true,
103       showPublishDate: true,
104       broadcastNewsFeeds: true,
105       broadcastNewsUpdates: true
106     }
107   },
108   ],
109 ],
110 };
111
112
113 /***** DO NOT EDIT THE LINE BELOW *****/
114 if (typeof module !== "undefined") {module.exports = config;}
115
```

이곳에 코드 붙여넣고, 저장



MagicMirror에 gateway 설치

```
$ cd ~/MagicMirror/modules
```

```
$ git clone https://github.com/bugsounet/Gateway
```

```
$ cd Gateway
```

```
$ npm install
```

```
$ cd ~
```

```
$ geany MagicMirror/config/config.js
```



MagicMirror에 gateway 설치

다음은 모듈 리스트 하단에 코드 추가 후, 저장

```
{  
    module: 'Gateway'  
},
```

The screenshot shows the MagicMirror configuration file (config.js) in a code editor. The file is being edited to add the 'Gateway' module. The code is as follows:

```
104 broadcastNewsFeeds: true,  
105 broadcastNewsUpdates: true  
106 }  
107 },  
108 {  
109     module: "MMM-GoogleAssistant",  
110     configDeepMerge: true,  
111     config: {  
112         assistantConfig: {  
113             lang: "ko-KR",  
114             latitude: 37.3460,  
115             longitude: 126.550,  
116         },  
117     },  
118 },  
119 {  
120     module: 'Gateway'  
121 },  
122 ]  
123 };  
124  
125 /***** DO NOT EDIT THE LINE BELOW *****/
```

The new module entry at line 119-121 is highlighted with a red box.



MagicMirror에 EXT-detector 설치

```
$ cd ~/MagicMirror/modules
```

```
$ git clone https://github.com/bugsounet/EXT-Detector
```

```
$ cd EXT-Detector
```

```
$ npm install
```

```
$ cd ~
```

```
$ geany MagicMirror/config/config.js
```



다음은 모듈 리스트 하단에 코드 추가 후, 저장

```
{
    module: "EXT-Detector",
    position: "top_left",
    configDeepMerge: true
},
```

```

111 config: {
112   assistantConfig: {
113     lang: "ko-KR",
114     latitude: 37.3460,
115     longitude: 126.550,
116   },
117 }
118 },
119 {
120   module: 'Gateway'
121 },
122 {
123   module: "EXT-Detector",
124   position: "top_left",
125   configDeepMerge: true
126 },
127 ]
128 };
129
130 /***** DO NOT EDIT THE LINE BELOW *****/
131 if (typeof module !== "undefined") {module.exports = config;}

```



MagicMirror 실행

```
$ cd ~/MagicMirror
```

```
$ npm run start
```

마이크가 사용가능한 환경에서 호출음 '자비스' 입력

(예시) "자비스", 오늘 날씨 어때?

MagicMirror 종료

Ctrl + q 입력



참고 자료

- MagicMirror² 설치 및 사용 방법
 - <https://docs.magicmirror.builders/>
- MMM-GoogleAssistant 설치 및 사용 방법
 - <https://wiki.bugsounet.fr/en/MMM-GoogleAssistant>

Raspberry Pi를 활용한 IoT 프로젝트

Thank You