내 캘린더에서
텔레그램 또는 이메일로

Make 가입 이메일을 교수에게 보내는 것부터 시작합니다. 알림 받을 방법을 하나 고른 뒤 본인 계정으로 연결합니다. 각 단계의 **완료 확인**에 체크하며 진행하세요.

교수 연락: ahn@dreamfurnace.com
함께 사용하는 봇: @make_dot_com_tel_bot 열기

> **텔레그램:** 1장 → 2장의 봇 준비 → 3~6장 순서로 진행합니다.
> **이메일:** 1장 → 3장으로 이동합니다. 3-5 대신 7장의 Gmail 설정을 마친 뒤 3-6으로 돌아옵니다. Telegram 앱과 Chat ID는 필요하지 않습니다.
> 두 방법 모두 테스트 후 8장의 폴더·라벨, 9장의 자동 실행을 설정합니다.

이 파일은 이미지, 두 버전의 블루프린트와 공통 스크립트를 포함해 따로 배포할 수 있습니다. Notion 실제 캡처를 해당 작업 바로 아래에 배치했습니다. 봇 대화·테스트 예시는 설명용 그림입니다. 캡처 속 이름·주소 대신 본인 값을 입력하세요.

자료 다운로드 · 블루프린트 1개 + 공통 스크립트

텔레그램·이메일 중 사용할 버전 하나와 공통 스크립트를 내려받습니다. 버튼은 세 개지만 실제 사용할 파일은 두 개입니다. 보통 다운로드 폴더에 저장됩니다.

함께 쓰는 것과 학생마다 다른 것

항목	누가 준비하나요?	학생이 사용할 값
Telegram 봇	교수 · 공용	@make_dot_com_tel_bot
봇의 Chat ID 자동 안내	교수 · 한 번 준비	학생은 Telegram에서 /start 또는 /myid
Telegram 수신 Chat ID	학생 개인 대화마다 다름	봇이 내 대화로 회신한 숫자
이메일 버전의 Gmail Connection	학생 본인	메일을 보낼 본인의 Gmail 계정

항목	누가 준비하나요?	학생이 사용할 값
이메일 버전의 To	학생 본인	본인이 알림을 받고 싶은 메일 주소
일정용 Custom webhook	학생마다 새로 생성	내 시나리오에서 Copy address로 복사
Apps Script WEBHOOK_URL	학생마다 다름	바로 위에서 복사한 내 Custom webhook URL
Calendar·Apps Script	학생별 수업용 Google 계정	내 기본 캘린더 + 내 새 스크립트 프로젝트

> **학생 A: A의 캘린더 → A의 스크립트 → A의 Webhook → A의 시나리오 → 공용 봇 → A의 개인 대화**
> 학생 B도 같은 순서로 진행하되 Webhook URL과 Chat ID는 B의 값을 씁니다. 교수 봇의 수신 설정과 학생의 일정용 Custom webhook은 서로 다른 설정입니다.

01 / 09

가입 이메일을 보내고 초대 수락

Make에 가입한 바로 그 이메일을 교수에게 보냅니다.

1-1. Make에 가입하고 이메일 확인

열 화면 · 컴퓨터 브라우저 · make.com

1. https://www.make.com/ 을 엽니다. 계정이 없으면 가입하고, 있으면 로그인합니다. Google로 계속하기를 썼다면 선택한 Google 계정을 확인합니다.
2. 프로필 또는 계정 설정에서 가입 이메일을 확인해 복사합니다. 복사는 Windows에서 Ctrl+C, Mac에서 ⌘C입니다.

완료 확인 · Make에 로그인했고 가입 이메일을 정확히 알고 있습니다.

1-2. 교수에게 초대 요청 메일 보내기

열 화면 · 평소 사용하는 메일 서비스



설명용 메일 화면 · 대괄호 부분은 본인 정보로 바꾸세요.

1. 받는 사람에 ahn@dreamfurnace.com을 입력합니다.
2. 제목은 [모델매니지먼트교육] Make 팀 초대 요청으로 씁니다. 본문에 이름과 Make 가입 이메일을 적고 보냅니다. 아래 예시의 대괄호 부분을 본인 정보로 바꿉니다.

```
받는 사람: ahn@dreamfurnace.com
제목: [모델매니지먼트교육] Make 팀 초대 요청

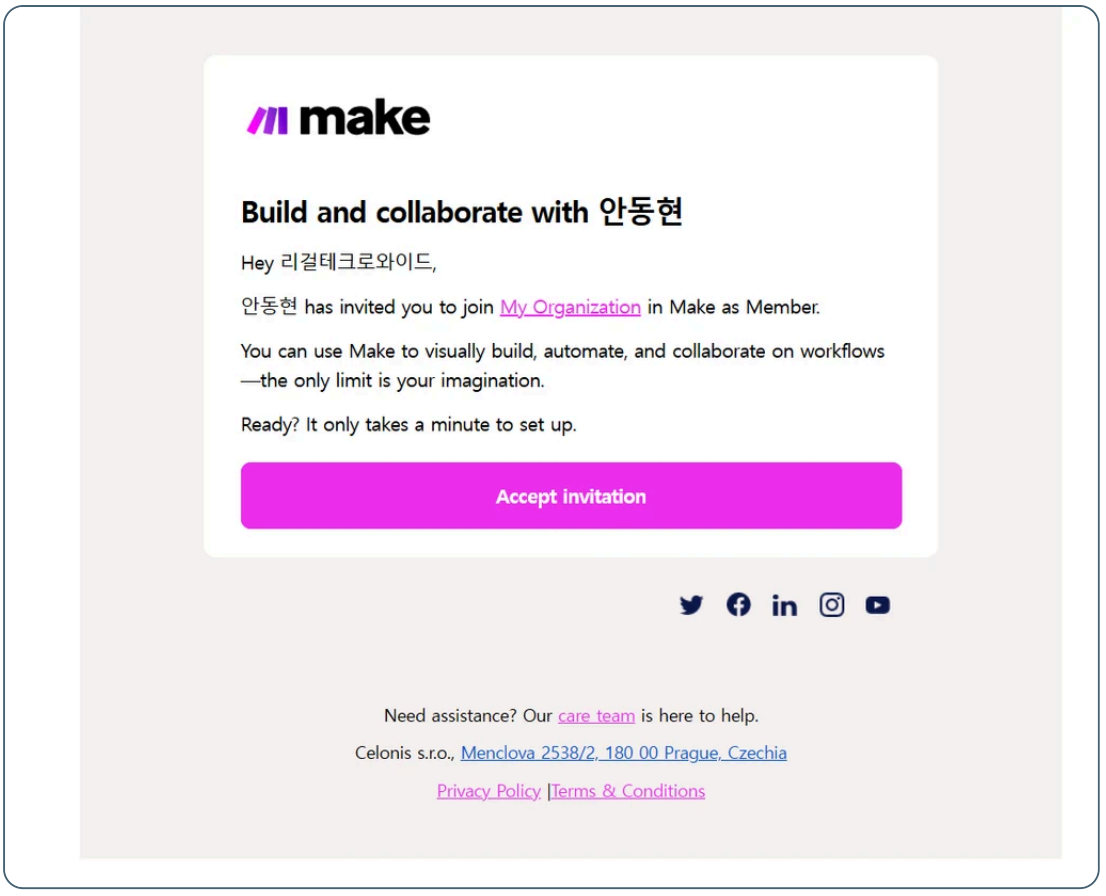
이름: [본인 이름]
Make 가입 이메일: [Make 프로필에 표시된 이메일]
```

완료 확인 · 보낸메일함에서 받는 주소와 Make 가입 이메일을 확인했습니다.

1-3. 초대 메일 수락 후 수업 팀으로 이동

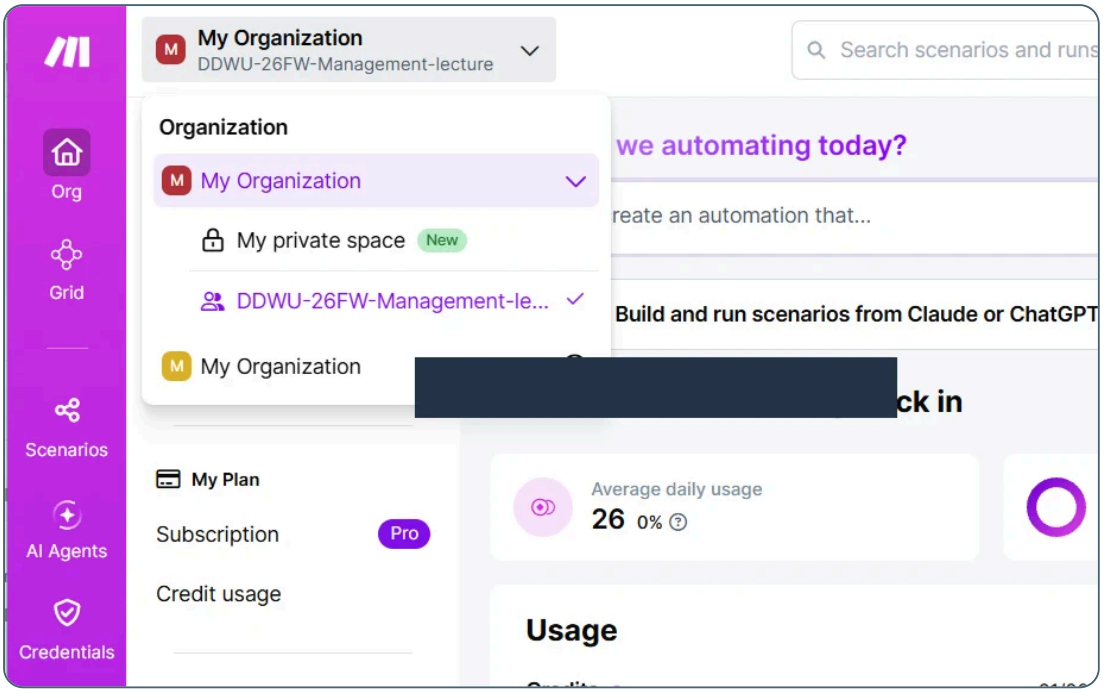
열 화면 · Make 가입 이메일의 받은편지함 → Make

1. 교수가 초대를 보내면 Make 초대 메일을 엽니다. Accept invitation(초대 수락)을 누릅니다.



실제 초대 메일 · 가운데 Accept invitation을 누릅니다.

2. 초대받은 이메일 계정으로 로그인합니다. 왼쪽 위 조직/팀 선택 메뉴를 열고 DDWU-26FW-Management-lecture 팀을 선택합니다. 개인 공간(My own space)이 아닌 수업 팀인지 확인합니다.



실제 팀 선택 메뉴 · DDWU-26FW-Management-lecture를 선택합니다. 계정 표시 일부는 가렸습니다.

완료 확인 · 수업 팀에서 새 시나리오를 만들 수 있습니다.

02 / 09

공용 봇에서 내 Chat ID 받기

봇은 함께 쓰고, 알림을 받는 개인 대화는 각자 사용합니다.

> 이메일 버전은 이 장을 건너뛰고 3장 · 내 시나리오와 연결 준비로 이동합니다.

2-1. 교수 봇과 개인 대화 시작

열 화면 · Telegram 앱



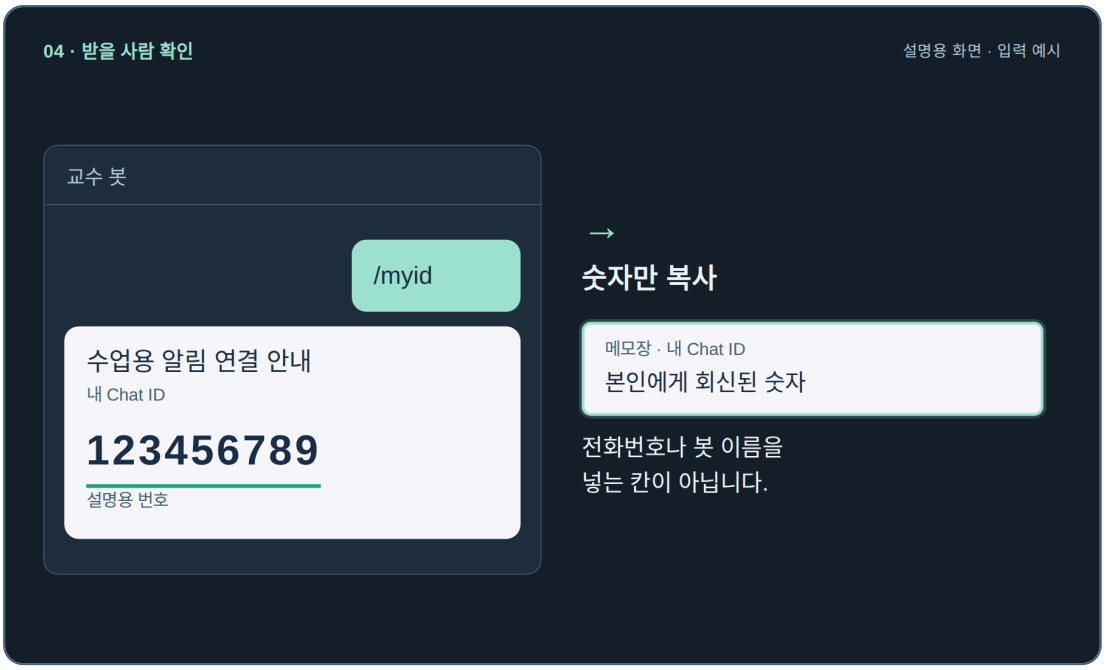
설명용 Telegram 화면 · 봇과 나의 개인 대화에서 진행합니다.

1. Telegram 앱을 열고 검색창(돋보기)에 @make_dot_com_tel_bot을 입력합니다. 검색 결과에서 사용자명이 정확히 일치하는 봇을 누릅니다.
2. 또는 https://t.me/make_dot_com_tel_bot 링크를 누르고 Telegram 열기를 선택합니다. 두 방법 모두 같은 봇의 개인 대화를 엽니다.
3. Start(시작)를 누릅니다. 이미 대화 중이면 입력창에 /start를 쓰고 전송합니다. 단체 대화방이 아닌 봇과 나의 개인 대화에서 진행합니다.

완료 확인 · 정확한 교수 봇의 개인 대화에 /start를 보냈습니다.

2-2. 자동 안내에서 숫자만 복사

열 화면 · 방금 연 교수 봇 대화



설명용 예시 번호 · 화면의 123456789를 그대로 쓰지 마세요.

1. 자동 회신의 내 Chat ID 줄을 찾습니다. 다시 확인하려면 /myid를 보냅니다.
2. 내 번호의 숫자만 복사합니다. 아래 123456789는 설명용 예시이므로 그대로 사용하지 않습니다. 메모장에 잠시 붙여 넣어 두면 편합니다.

```
수업용 알림 연결 안내
내 Chat ID: 123456789

Make의 Telegram 보내기 모듈에서
Chat ID 칸에 위 숫자만 붙여 넣으세요.
```

완료 확인 · 자동 회신에서 받은 본인의 Chat ID가 준비됐습니다.

내 시나리오와 연결 준비

수업 팀 안에 내 이름의 복제본을 만들고, 연결 세 가지를 확인합니다.

3-1. 기본형 파일 내려받기

열 화면 · 이 튜토리얼의 자료 다운로드

1. 이 페이지 위쪽의 자료 다운로드에서 텔레그램 버전 또는 이메일 버전 블루프린트 중 하나를 내려받습니다. 텔레그램은 calendar-conflict.blueprint.json, 이메일은 calendar-conflict-email.blueprint.json입니다.
2. 공통 파일인 GoogleAppsScript.template.js도 내려받습니다. 선택한 블루프린트 1개와 스크립트 1개, 총 두 파일을 사용합니다. 파일은 보통 다운로드 폴더에 저장됩니다.
3. 파일 이름과 확장자를 그대로 둡니다. .json은 다음 3-2에서 Make로 가져올 파일이고, .js는 4-1에서 Apps Script에 붙일 코드입니다.

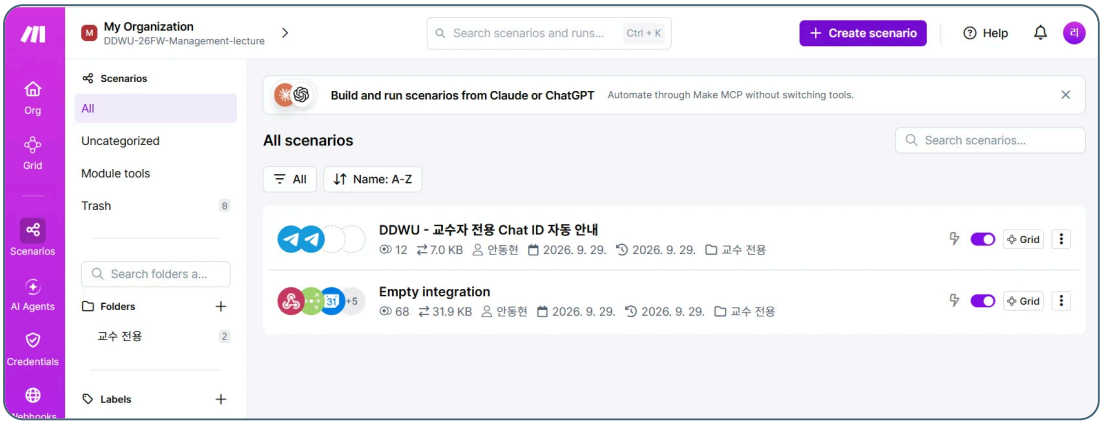
↑ 자료 다운로드로 이동

완료 확인 · 다운로드 폴더에서 두 파일을 찾을 수 있습니다.

3-2. 새 시나리오에 Blueprint 가져오기

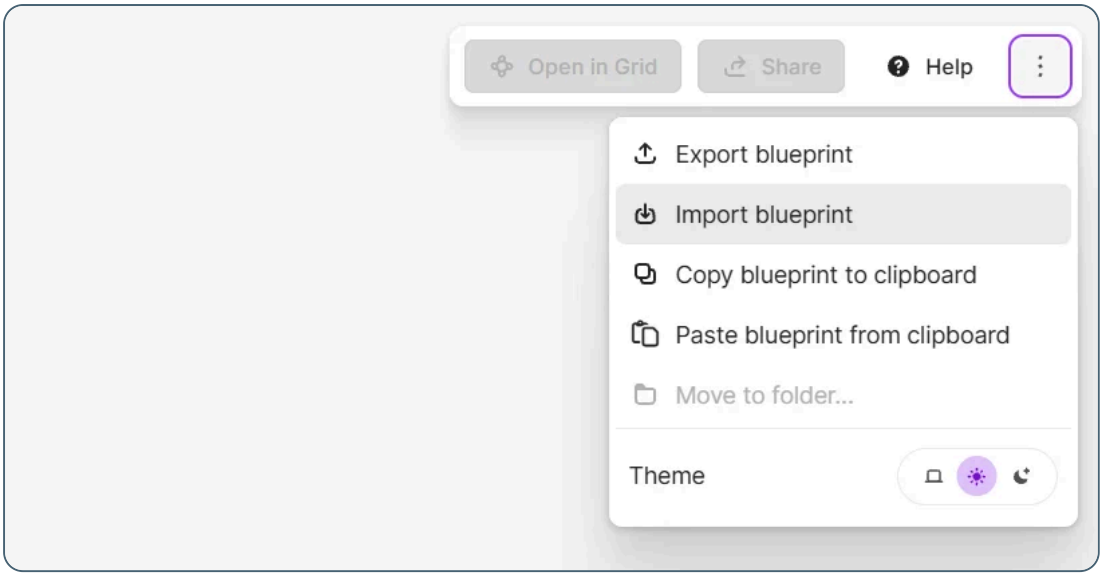
열 화면 · Make · 수업 팀 → Scenarios

1. Create scenario(새 시나리오)를 누르고 이름을 이름_과제1_일정알림으로 정합니다.

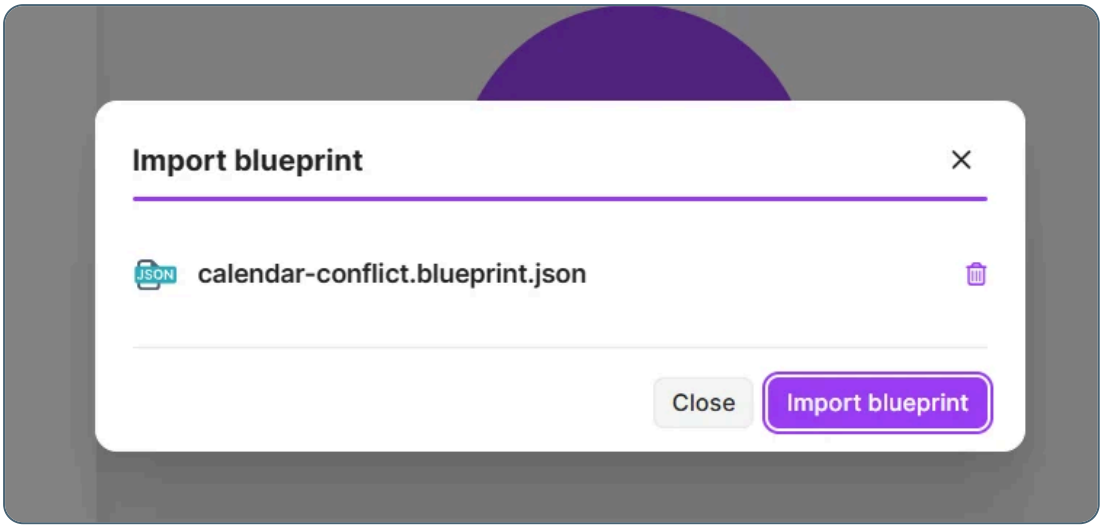


실제 Scenarios 화면 · 오른쪽 위 Create scenario를 누릅니다.

2. 편집 화면 오른쪽 위 ⋯ 메뉴에서 Import blueprint를 선택합니다. 파일 선택으로 3-1에서 내려받은 텔레그램 또는 이메일 버전의 .json 파일 하나를 고른 뒤 창 아래 Import blueprint를 누릅니다.

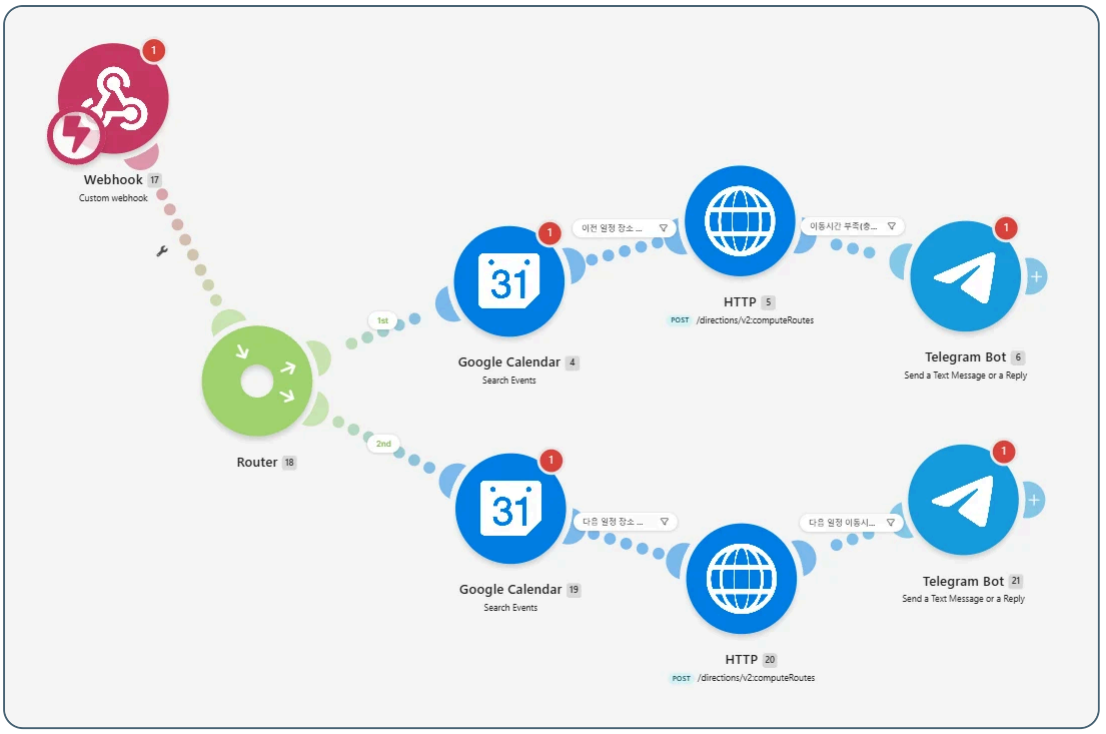


실제 ⋯ 메뉴 · Import blueprint 선택



실제 파일 선택 창 · 파일을 고른 뒤 아래 Import blueprint를 누릅니다.

3. Webhook 1개, Router 1개, 위쪽 3개, 아래쪽 3개가 있는지 확인합니다. 마지막 두 모듈은 선택한 버전에 따라 Telegram 또는 Gmail입니다. 아래 캡처는 텔레그램 버전입니다.



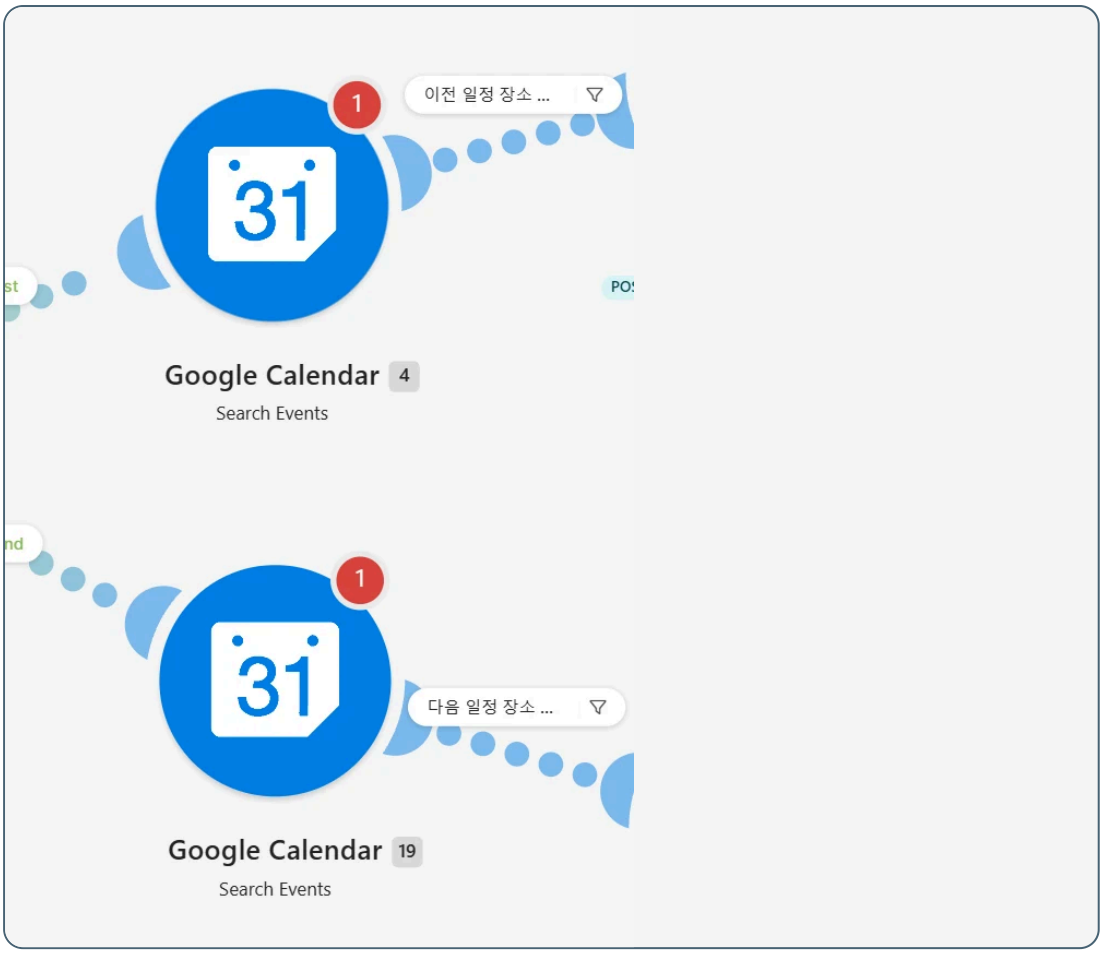
실제 가져오기 직후 화면 · 연결 전 빨간 표시가 있어도 아래 단계에서 차례로 설정합니다.

완료 확인 · 내 이름의 시나리오에 8개 모듈이 나타납니다.

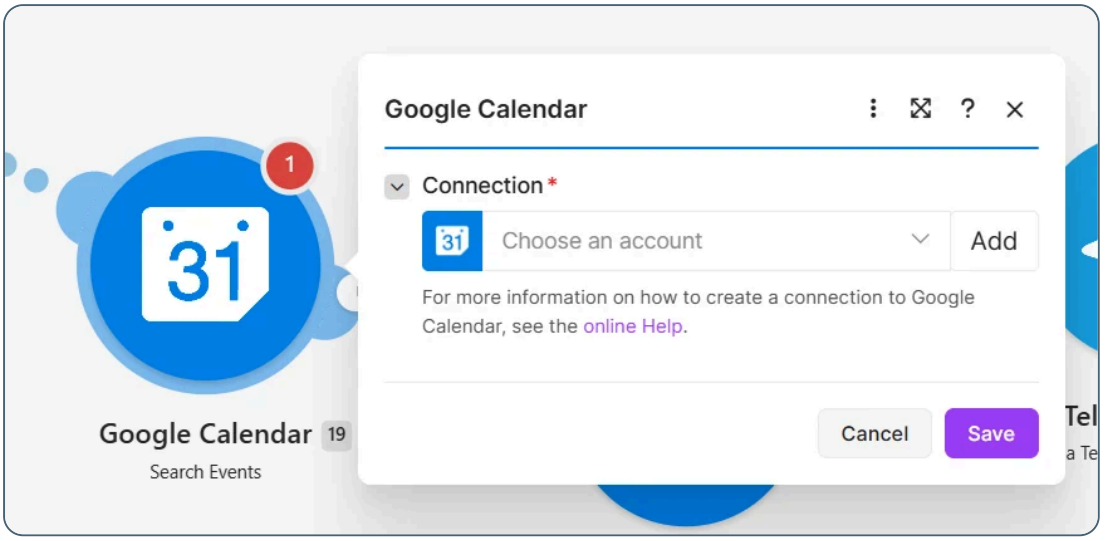
3-3. 2개의 캘린더 설정하기

열 화면 · 위쪽 A → 아래쪽 B의 Google Calendar 모듈

1. 위쪽 Google Calendar 원을 클릭합니다. Connection 옆 Add를 누르고 연결 이름을 본인이름_Google 로 정합니다. Google 로그인 창에서 수업에 사용할 본인 계정을 선택하고 연결을 허용합니다.

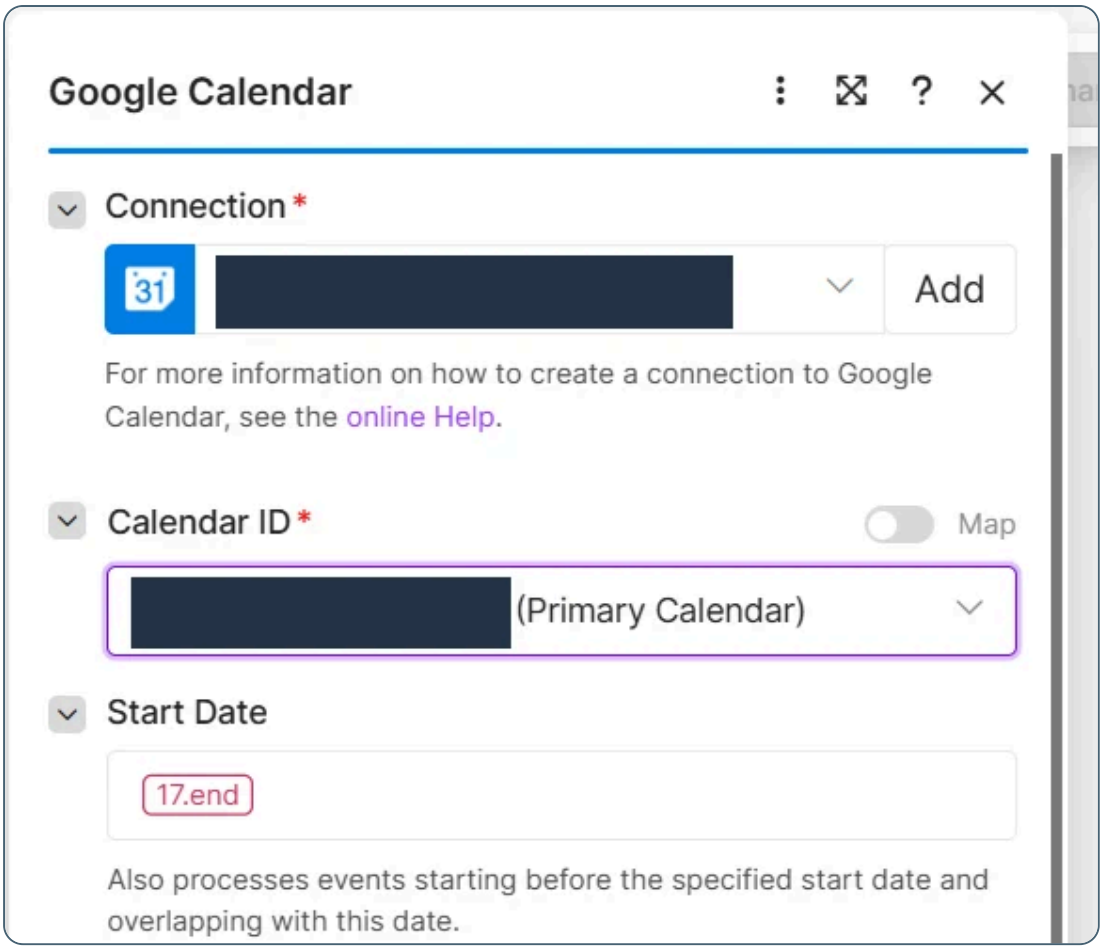


실제 Calendar 두 곳 · 위쪽과 아래쪽을 모두 설정합니다.



실제 연결 전 화면 · Choose an account 옆 Add로 본인의 Google 계정을 연결합니다.

2. Connection에 방금 만든 내 연결을 선택합니다. Calendar ID에서 본인의 기본 캘린더를 고릅니다. 직접 입력하는 화면이면 primary를 입력합니다. 교수나 친구의 이메일이 보이는 연결은 선택하지 않습니다.



실제 계정 연결 후 화면 · 계정 정보는 가렸습니다. 본인의 Primary Calendar를 고릅니다. 이 캡처는 아래쪽 B이므로 Start Date가 17.end입니다.

3. 아래 표와 검색 설정을 비교합니다. 색깔 있는 start·end 값과 addHours 계산식은 가져온 그대로 둡니다. Save(또는 OK)를 눌러 닫습니다.
4. 아래쪽 Google Calendar도 엽니다. Connection에는 방금 만든 동일한 내 Google 연결을 고르고, Calendar ID도 같은 기본 캘린더를 선택합니다. 아래쪽 검색 범위는 위쪽과 다르므로 복사해 덮어쓰지 않습니다. 저장합니다.

설정	위쪽 A · 이전 일정 검색	아래쪽 B · 다음 일정 검색
Connection / Calendar ID	내 Google 연결 / 내 기본 캘린더	위쪽과 동일
Time from (timeMin)	새 일정 start에서 4시간 전	새 일정 end
Time to (timeMax)	새 일정 start	새 일정 end에서 4시간 후
Single events / Order by / Limit	Yes / Start time / 10	Yes / Start time / 10

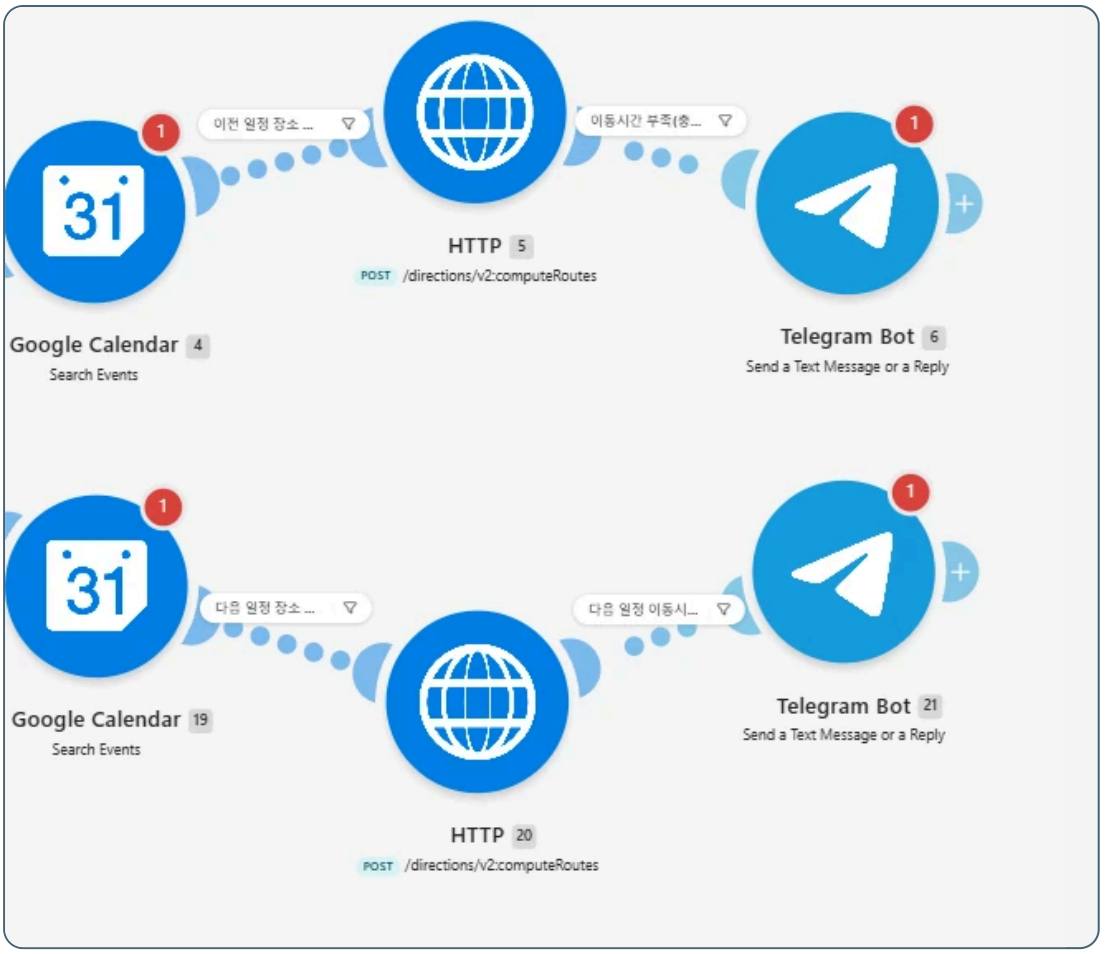
완료 확인 · Calendar 두 곳 모두 내 Google 연결과 같은 기본 캘린더를 사용합니다.

3-4. 2개의 경로 HTTP 노드 설정하기

열 화면 · 각 Calendar 바로 다음의 파란 HTTP 모듈

> **수업용 Google Routes API 키**
>
> 교수가 과제 실습용으로 제공한 키입니다. 복사해서 위쪽·아래쪽 HTTP의 Query parameters → key → Value에 동일하게 붙여 넣으세요.
>
> ```
> AIzaSyDUDK9UqkYkviLhtoJ5aImgc7icAuM78pE
> ```

1. Blueprint에는 HTTP가 이미 위아래 하나씩 있습니다. 위쪽의 파란 HTTP 원을 클릭합니다. 이 단계에서 바꿀 것은 두 HTTP의 key 값뿐입니다.



실제 시나리오에서 HTTP 위치 · 두 파란 지구본을 각각 엽니다. 제공된 캡처에는 HTTP 설정창이 없어 입력값은 아래 표로 안내합니다.

2. 위의 수업용 Routes API 키 복사 버튼을 누릅니다. 별도로 키를 신청하거나 Google Cloud 계정을 만들 필요가 없습니다.
3. Query parameters(또는 Query String)를 펼칩니다. 두 번째 항목에서 Name이 key인지 확인합니다. 바로 아래 Value의 YOUR_GOOGLE_ROUTES_API_KEY를 전체 선택해 지우고 복사한 키를 붙여 넣습니다. 따옴표나 key=는 붙이지 않습니다.
4. Save(또는 OK)를 누르고 아래쪽 HTTP도 엽니다. 똑같이 Name이 key인 항목의 Value를 같은 수업용 키로 바꾸고 저장합니다.
5. 시나리오 전체를 Save합니다. 두 HTTP에 같은 키가 들어 있으면 이 단계는 완료입니다. fields, URL, POST, JSON 본문과 색깔 있는 매핑은 제공된 값을 유지합니다. 아래 표는 설정을 확인할 때 참고합니다.

찾을 항목	두 HTTP에서 확인할 값
Authentication type	No authentication · 제공된 파일은 query key 방식
URL	https://routes.googleapis.com/directions/v2:computeRoutes
Method	POST

찾을 항목	두 HTTP에서 확인할 값
Query parameters → Name: fields	Value: routes.duration,routes.distanceMeters
Query parameters → Name: key	Value: 이 페이지의 수업용 Routes API 키
Body content type / Input method	application/JSON / JSON string
Parse response	Yes
본문의 travelMode / languageCode	TRANSIT / ko

본문에서 서로 달라야 하는 세 곳

JSON 항목	위쪽 A	아래쪽 B
origin.address · 출발 장소	이전 Calendar의 location	Webhook의 location
destination.address · 도착 장소	Webhook의 location	다음 Calendar의 location
departureTime · 출발 시각	이전 Calendar의 end	Webhook의 end

HTTP를 실수로 지웠다면

실행 전이라면 3-2의 기본형을 새 시나리오에 다시 불러오는 것이 가장 쉽습니다. 새 HTTP를 직접 추가하면 모듈 번호가 달라질 수 있어 뒤 Telegram의 duration 매핑과 충돌 필터도 다시 연결해야 합니다. 키만 붙여 넣는 것으로 복구되지는 않습니다.

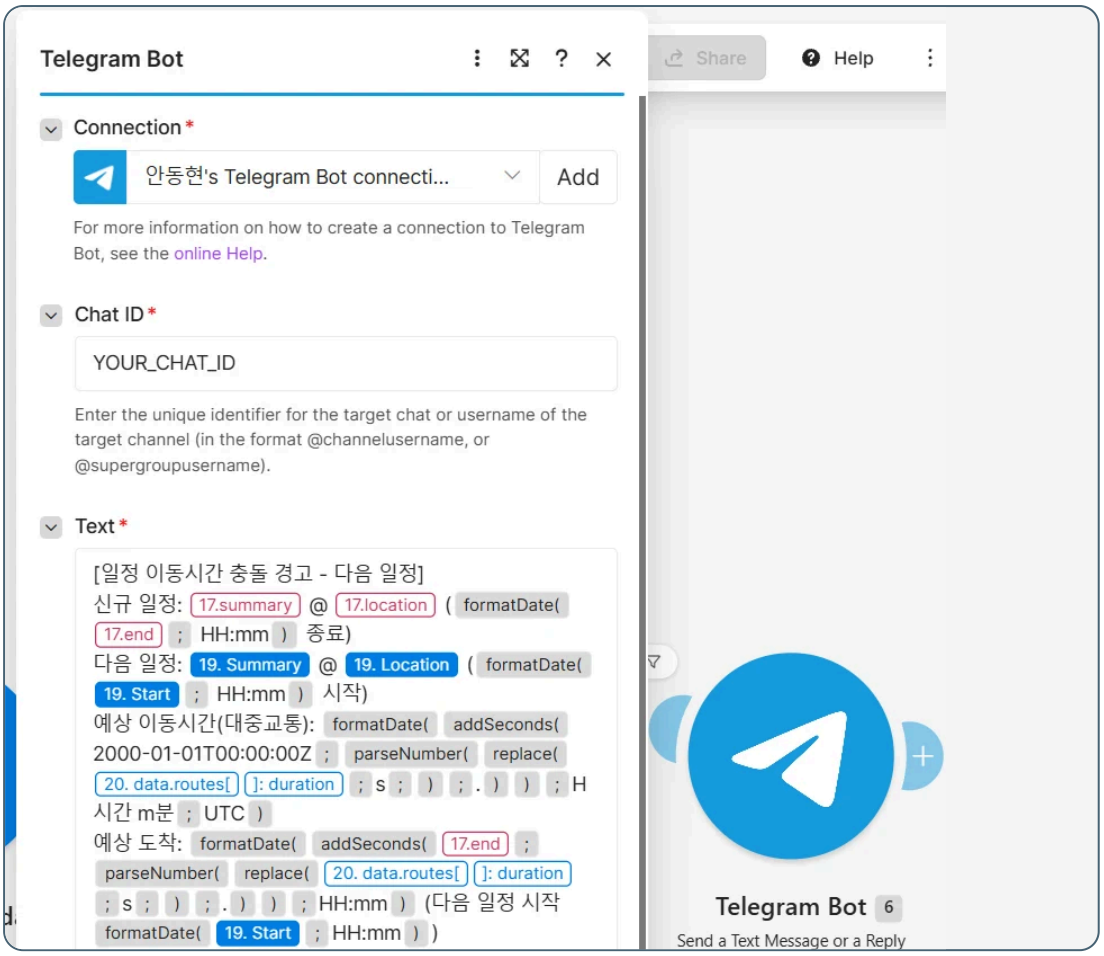
완료 확인 · 위쪽·아래쪽 HTTP의 key → Value에 이 페이지에서 복사한 동일한 키를 넣고 저장했습니다.

3-5. 2개의 텔레그램 노드 설정하기

열 화면 · 위쪽 A → 아래쪽 B의 마지막 Telegram Bot 모듈

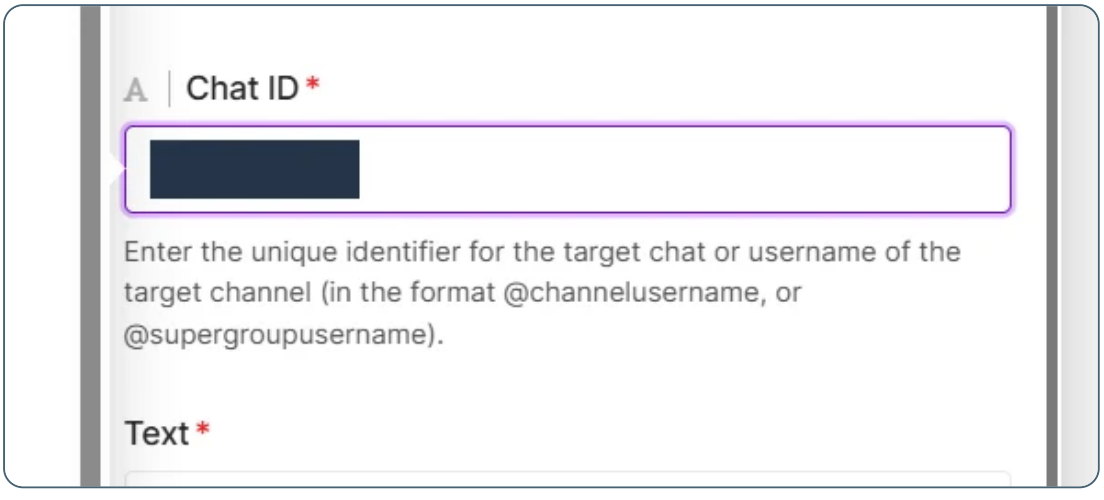
> 이메일로 받고 싶다면 이 단계 대신 7장의 Gmail 설정을 진행합니다. 설정을 마친 뒤에는 3-6으로 돌아와 내 Webhook을 만듭니다. 두 버전을 모두 만들 필요는 없습니다.

1. 위쪽 Telegram 원을 클릭합니다. Connection 목록에서 교수가 안내한 @make_dot_com_tel_bot용 기존 연결을 선택합니다. Connection은 메시지를 보내는 봇입니다. 학생은 Add로 새 봇 연결을 만들지 않습니다.



실제 Telegram 설정창 · Connection, Chat ID, Text를 위에서 아래로 확인합니다.

2. Chat ID 칸의 YOUR_CHAT_ID를 지우고 2-2에서 자동 회신으로 받은 내 숫자만 붙여 넣습니다. Chat ID는 알림을 받을 나의 개인 대화입니다. 봇 이름·전화번호·화면의 예시 번호를 쓰지 않습니다.



실제 Chat ID 입력 위치 · 캡처의 개인 번호는 가렸습니다. 2-2에서 본인에게 온 숫자를 넣습니다.

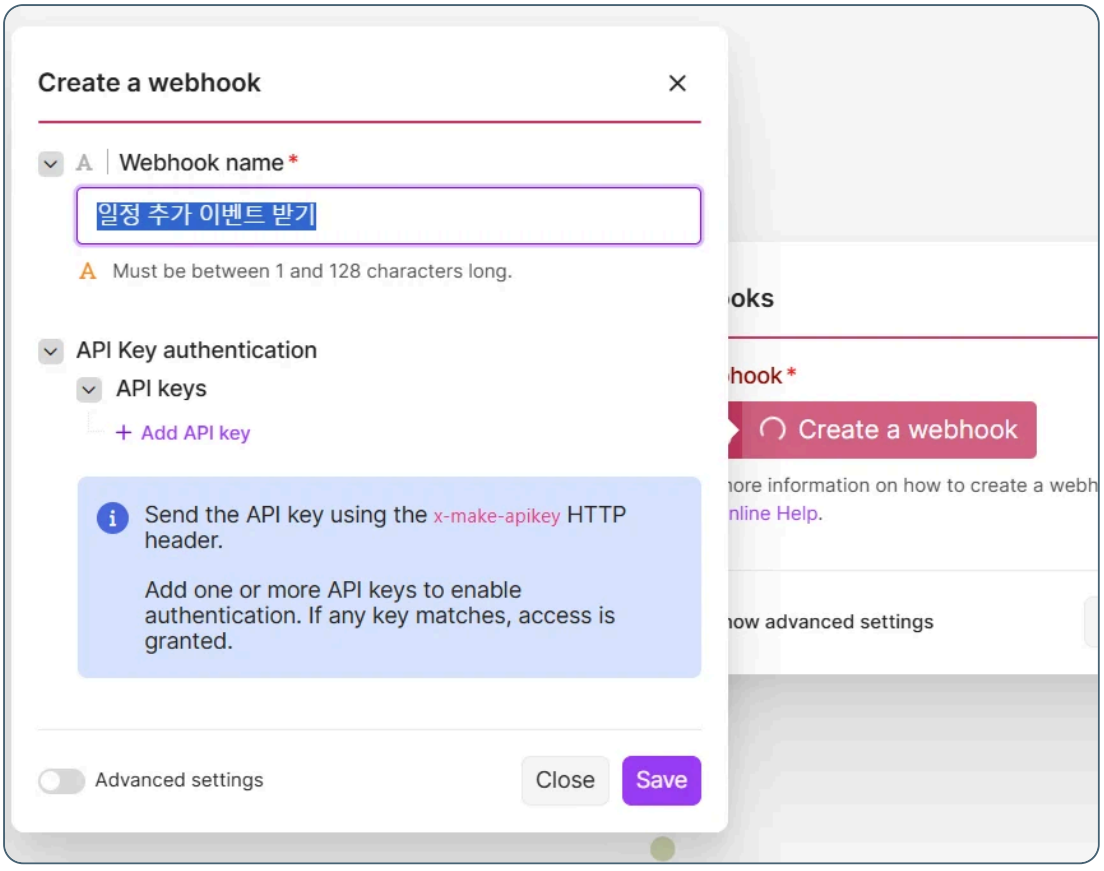
3. Text는 보낼 메시지 내용입니다. 제목·시간·이동시간의 색깔 있는 매핑은 그대로 둡니다. Save(또는 OK)를 눌러 닫습니다.
4. 아래쪽 Telegram도 열어 동일한 교수 봇 연결과 동일한 내 Chat ID를 넣고 저장합니다. 두 경로의 Text 내용은 서로 다르므로 위쪽 문구를 통째로 복사하지 않습니다.

완료 확인 · Telegram 두 곳의 보내는 봇은 교수 봇이고, 받는 Chat ID는 모두 내 번호입니다.

3-6. 내 Custom webhook 만들고 주소 복사

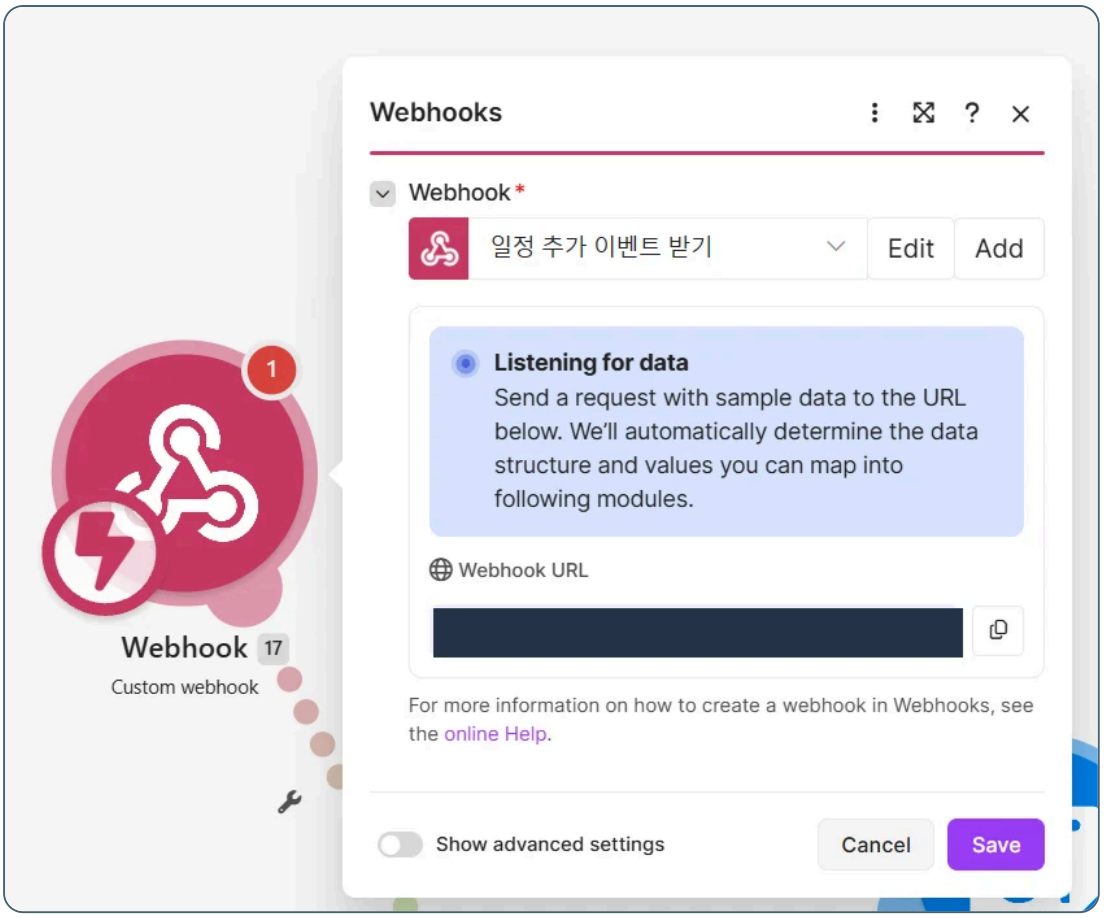
열 화면 · 가장 왼쪽 Webhooks 모듈

1. Custom webhook을 열고 Add(추가)를 눌러 이름_일정수신처럼 이름을 붙입니다. 목록에 교수나 다른 학생의 Webhook이 보여도 선택하지 않고 내 것을 새로 만듭니다.



실제 Create a webhook 창 · 이름을 입력하고 Save를 누릅니다. 이 기본형에서는 별도 API key authentication을 추가하지 않습니다.

2. 저장한 뒤 Copy address로 수신 URL을 복사합니다. 이 주소는 학생마다 다릅니다. 교수 주소나 친구 주소를 복사하지 않습니다.



실제 Webhook 창 · 주소 오른쪽 복사 아이콘을 누릅니다. 캡처의 수신 주소는 가렸습니다.

3. 시나리오도 Save로 저장합니다. 자동 실행 스위치는 아직 끈 상태로 두고, 다음 단계에서 입력 코드를 준비합니다.

완료 확인 · 내 이름의 Webhook에서 복사한 나만의 URL이 준비됐습니다.

04 / 09

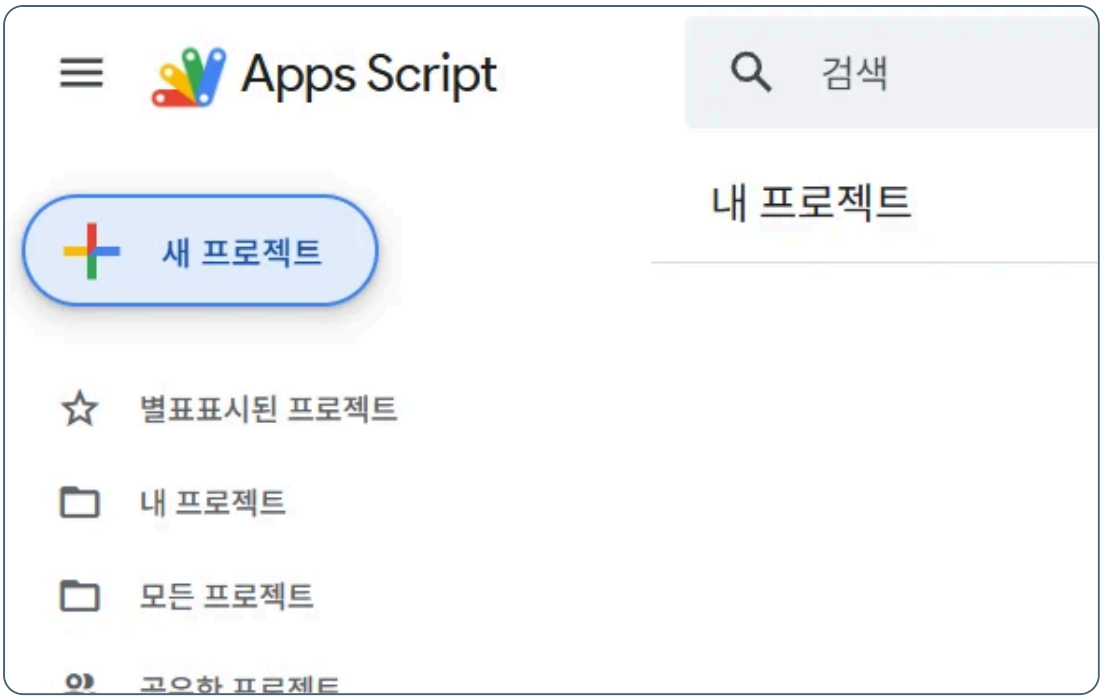
Apps Script에 캘린더 연결

수업용 Google 계정의 새 프로젝트에서 코드 한 곳을 바꾸고 초기화합니다.

4-1. 새 프로젝트에 템플릿 코드 붙이기

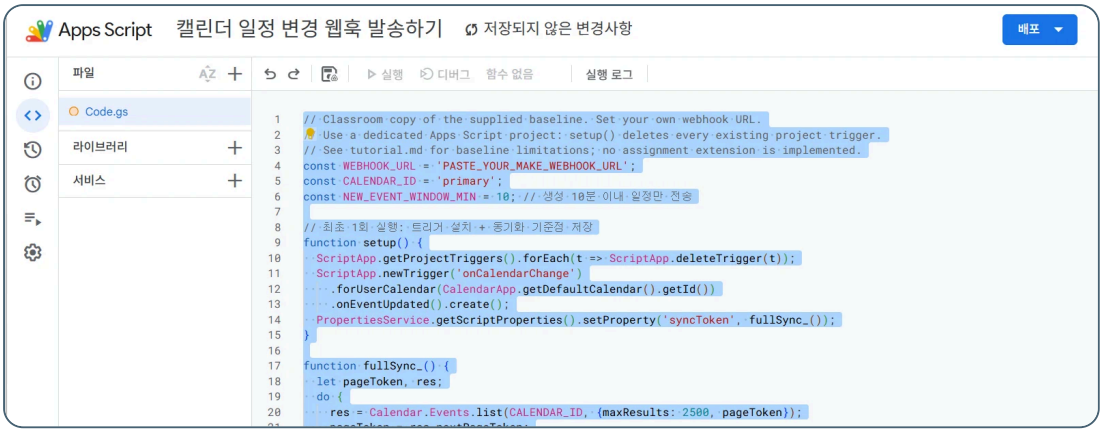
열 화면 · https://script.google.com/ · 수업용 Google 계정

1. 새 프로젝트를 만들고 이름_일정감지로 이름을 바꿉니다. 기존 업무 프로젝트는 사용하지 않습니다.



실제 Apps Script 첫 화면 · 새 프로젝트를 누릅니다.

2. 다운로드한 GoogleAppsScript.template.js를 메모장 또는 코드 편집기로 열어 전체 복사합니다. 또는 이 장 끝의 Apps Script 코드 보기 → 코드 전체 복사를 누릅니다. Code.gs 편집 영역을 클릭하고 기본 myFunction 코드를 모두 선택해 지운 뒤 붙여 넣습니다. 전체 선택은 Ctrl+A/⌘A, 붙여 넣기는 Ctrl+V/⌘V입니다.



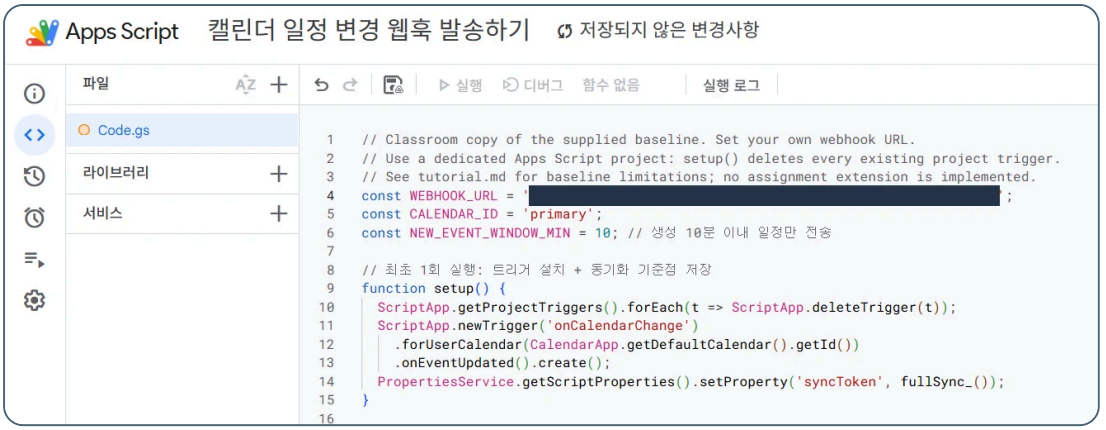
실제 Code.gs 편집 화면 · 기본 코드를 지우고 템플릿 전체를 붙여 넣습니다.

완료 확인 · Code.gs에 setup, onCalendarChange 등이 포함된 템플릿 전체가 들어 있습니다.

4-2. 내 Webhook URL 한 곳 바꾸기

열 화면 · Code.gs 위쪽의 WEBHOOK_URL

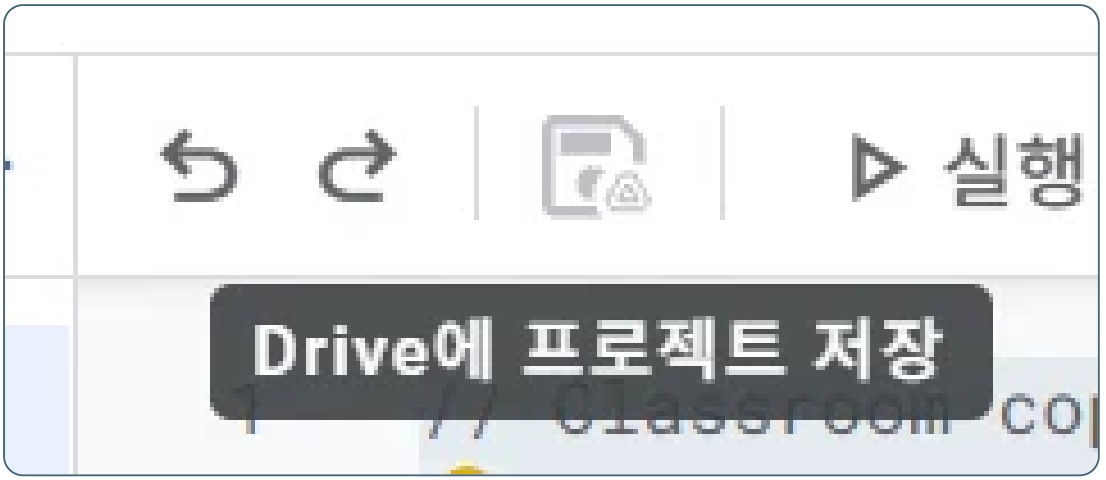
1. PASTE_YOUR_MAKE_WEBHOOK_URL만 지우고 3-6에서 내가 만든 Custom webhook의 수신 주소를 붙입니다. 양쪽 작은따옴표는 남깁니다.



실제 WEBHOOK_URL 수정 위치 · 따옴표 안의 개인 수신 주소는 가렸습니다. 3-6에서 복사한 내 주소를 넣습니다.

2. Make의 내 Webhook에서 Copy address를 다시 눌러 Code.gs의 주소와 같은지 비교합니다. Notion 캡처 속 주소·교수 주소·친구 주소는 복사하지 않습니다.

3. CALENDAR_ID = 'primary'는 유지합니다. Code.gs 편집창 안을 클릭한 상태에서 Windows는 Ctrl+S, Mac은 ⌘S를 누릅니다. 상단 코드 편집 도구줄의 디스크 모양 아이콘도 같은 프로젝트 저장 기능입니다. 오른쪽 위의 파란 배포 버튼과는 위치·역할이 다릅니다.



실제 저장 아이콘 확대 · 되돌리기·다시 실행 화살표 오른쪽, 실행 버튼 왼쪽입니다. 마우스를 올리면 “Drive에 프로젝트 저장”이라고 나옵니다.

4. 저장되지 않은 변경사항 표시가 사라지고 함수 목록에 setup이 나타나는지 확인합니다. 저장 오류가 뜨면 따옴표나 괄호를 확인합니다. 이 과제는 캘린더 트리거로 실행하므로 배포 버튼은 누르지 않습니다.

```
const WEBHOOK_URL = '여기에 내 Make 수신 URL';
const CALENDAR_ID = 'primary';
```

완료 확인 · Code.gs의 WEBHOOK_URL이 내 Make Custom webhook 주소와 일치하고 저장됐습니다.

4-3. setup 선택·실행 → Google 권한 승인 → 트리거 확인

열 화면 · Apps Script · Code.gs 편집기 → Google 승인 창 → 트리거

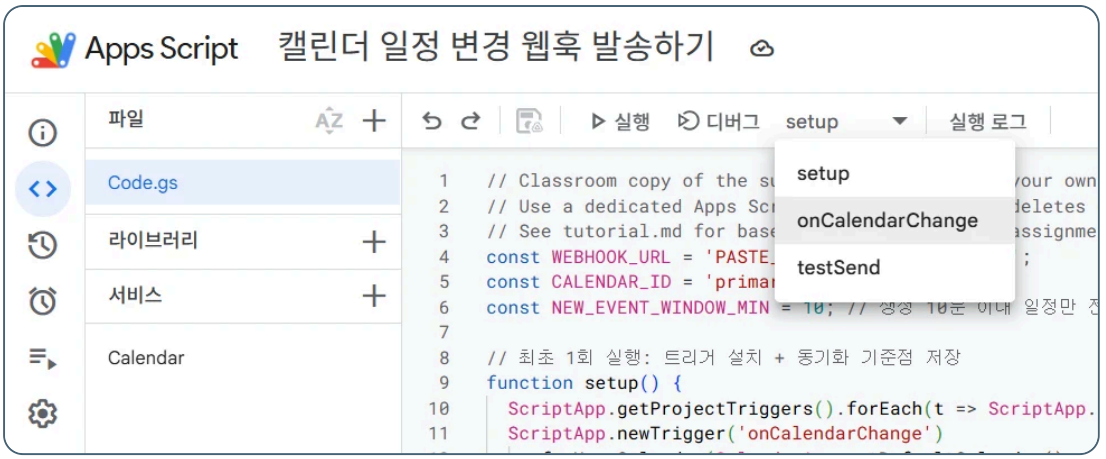
1. ① Calendar 서비스 추가: 왼쪽 서비스 옆 +를 누릅니다. 목록에서 Google Calendar API를 선택하고 식별자가 Calendar인지 확인한 뒤 추가합니다. 이미 서비스 아래 Calendar가 보이면 다시 추가하지 않습니다.



실제 서비스 추가 창 · 캡처는 창을 막 연 상태라 AdSense가 선택돼 있습니다. 그대로 추가하지 말고 목록을 내려 Google Calendar API를 선택하고 식별자 Calendar를 확인합니다.

2. ② 함수 선택: Code.gs 편집기 위쪽의 함수 드롭다운(실행·디버그 버튼 오른쪽)을 엽니다. 목록에서 setup을 클릭합니다. 드롭다운에 setup이라는 글자가 표시된 상태로 둡니다. 이번 단계는 testSend나

onCalendarChange를 선택하는 단계가 아닙니다.



실제 함수 목록 · 최초 설정에는 맨 위 setup을 선택하고 실행합니다. onCalendarChange는 이후 캘린더 변경 때 자동 실행될 함수입니다.

3. ③ 실행: 화면 상단의 버튼 [▶ 실행]을 클릭합니다. 처음 실행하면 권한 필요 또는 승인 필요 창이 나타날 수 있습니다. 권한 검토(Review permissions)를 누릅니다. 이미 같은 권한을 승인했다면 승인 창 없이 실행될 수 있으므로 ⑦로 넘어갑니다.

4. ④ 계정 선택: Google 계정 선택 창에서 Make의 Calendar 노드 두 곳에서 입력한 것과 동일한 본인 수업용 Google 계정을 선택합니다. 여러 계정이 보이면 이메일을 확인합니다.

5. ⑤ 확인되지 않은 앱 경고가 나온 경우: 고급(Advanced)을 펼친 뒤 “[내 프로젝트 이름](으)로 이동(안전하지 않음)” 또는 “Go to [project name] (unsafe)”을 클릭합니다. 경고가 없으면 다음으로 넘어갑니다.

6. ⑥ 권한 허용: 화면의 프로젝트 이름·계정을 다시 확인하고 요청하는 권한을 읽습니다. 이 코드는 캘린더를 조회하고, 캘린더 변경 시 자동 실행하며, Make로 외부 요청을 보냅니다. 권한 선택 체크박스가 나오면 이 기능에 필요한 권한을 선택하고 화면 아래 허용(Allow) 또는 계속(Continue)을 누릅니다. 내가 만든 프로젝트가 아니거나 코드와 관계없는 권한을 요구하면 진행하지 않고 교수에게 확인합니다.

7. ⑦ 실행 완료 확인: 승인 창이 닫히면 Apps Script로 돌아옵니다. 화면 상단의 [실행 로그] 버튼을 클릭해 로그 패널을 엽니다. 이미 로그 패널이 열려 있으면 그대로 확인합니다. 실행 시작(Execution started) 뒤에 실행 완료(Execution completed)가 표시되는지 확인합니다. 실행 중이면 완료될 때까지 기다립니다. 승인만 끝나고 실행이 시작되지 않았다면 함수 드롭다운이 setup인 상태에서 [▶ 실행]을 클릭하고 로그를 다시 확인합니다. 빨간 오류가 나오면 오류 문구를 확인합니다.

8. ⑧ 트리거 확인: 왼쪽 시계 모양의 트리거를 엽니다. 이벤트가 Calendar - 변경됨이고, 함수가 onCalendarChange인 행이 있어야 합니다. 트리거가 없으면 다음 단계로 넘어가지 말고 setup의 실행 내역을 확인합니다.

9. ⑨ 다음 단계로 이동: setup은 자동 감지를 설치하고 현재 캘린더를 기준점으로 저장하는 초기 설정입니다. 여기에서 Make로 일정이 전송되지 않아도 정상입니다. 4-4에서 새 일정을 만든 뒤 자동 수신을 확인합니다. 함수를 setup으로 설정하고 [▶ 실행] 버튼을 클릭하는 것은 한 번만 하면 됩니다.

함수 이름	언제 사용하는가	Make로 전송하는가
setup	최초 설정 · 권한 승인과 트리거 설치	전송하지 않음
testSend	수신 주소가 맞는지 테스트할 때 수동 실행	샘플 일정 1건 전송
onCalendarChange	setup 이후 캘린더 변경 때 자동 실행	기본형 조건에 맞는 일정만 전송

“이 앱은 차단되었습니다”는 다른 경우입니다

학교·조직의 관리자 정책으로 앱 접근이 차단됐거나 고급 → 이동 링크가 없는 경우에는 위의 ⑤ 경로로 진행할 수 없습니다. 교수에게 화면의 정확한 문구를 알립니다. 브라우저 보안 기능이나 조직 정책을 끄는 단계는 없습니다.

승인 창이 열리지 않거나 실행이 실패한다면

권한 검토를 눌러도 새 창이 안 보이면 브라우저 주소창의 팝업 차단 표시를 확인하고 Apps Script의 Google 승인 창을 허용합니다. Calendar is not defined 오류는 왼쪽 서비스에 Calendar가 있는지 확인합니다. 트리거 생성 뒤 오류가 난 경우에도 setup이 실행 완료될 때까지 초기 설정이 끝난 것으로 보지 않습니다.

Google 공식 안내

최초 실행의 권한 검토·계정 선택·허용: https://developers.google.com/workspace/tasks/quickstart/apps-script · 직접 개발한 앱의 확인되지 않음 경고: https://developers.google.com/apps-script/api/troubleshoot-authentication-authorization

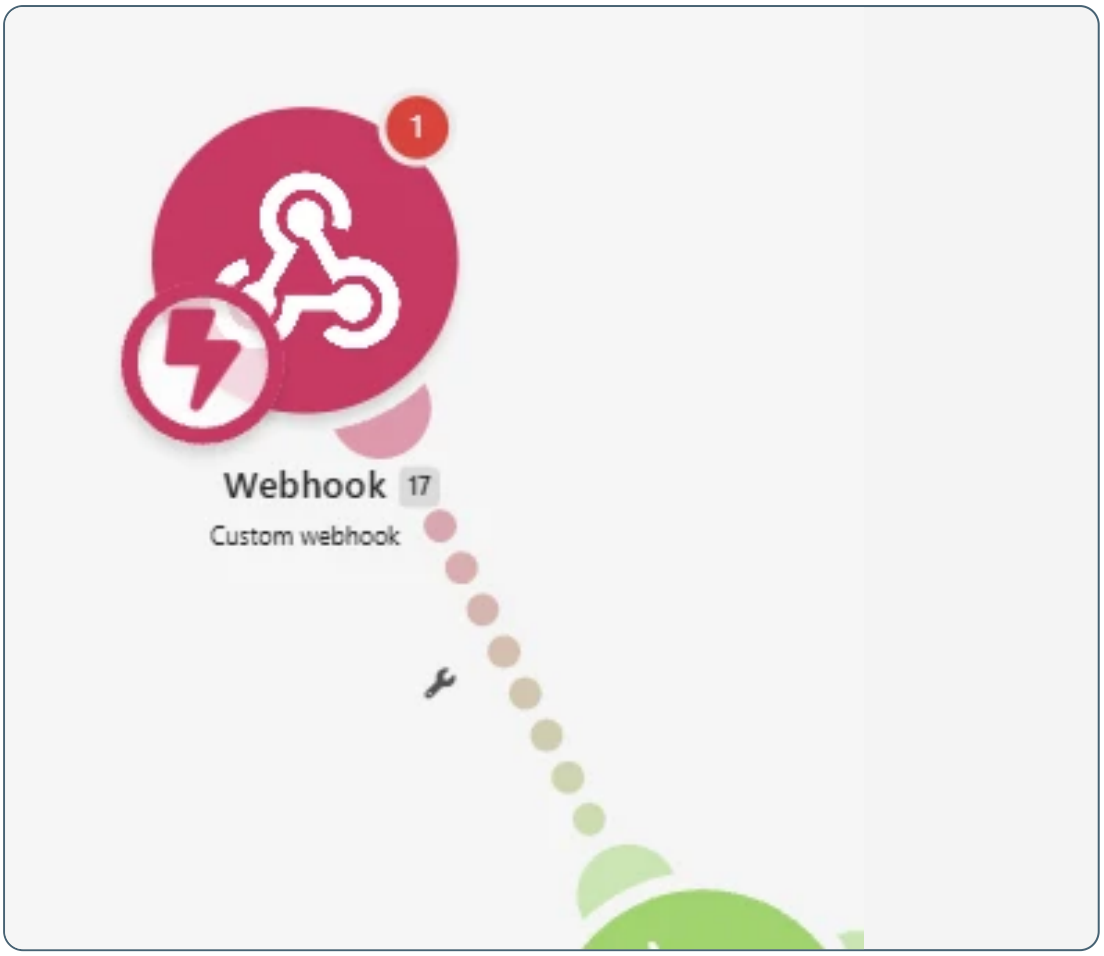
완료 확인 · setup의 실행 완료와 onCalendarChange 캘린더 트리거를 모두 확인했습니다. 권한 승인 창만 닫힌 상태와 구분합니다.

4-4. 새 일정 하나가 Make에 도착하는지 확인

열 화면 · Make → 수업용 Google Calendar

1. 4-3에서 setup 실행 완료와 트리거를 확인한 다음 진행합니다. Make 시나리오 편집기에서 Run once를 누르고, 기존 데이터 사용/새 데이터 대기 중 고르는 창이 나오면 새 데이터 대기(Wait for new data)를 선택합니다. 같은 Google 계정의 https://calendar.google.com/ 을 열고 기본 캘린더의 미래 날짜에 새 일정을 만듭니다. 제목은 이름_수신확인, 장소는 강남역처럼 입력하고 시작·종료 시각을 정해 저장합니다. 종일 체크는 끕니다. 이번에는 Apps Script의 실행 버튼을 누르지 않고 자동 실행을 기다립니다.

2. Make로 돌아와 맨 왼쪽 Webhook 원 위에 나타난 실행 결과 숫자를 누릅니다. Output → Bundle을 펼쳐 id, summary, location, start, end를 확인합니다. summary가 이름_수신확인인지, location과 시간이 방금 입력한 값인지 대조합니다. testSend로 전송한 “테스트 일정”과 구분합니다.



실제 Webhook 위치 확대 · 새 일정 수신 뒤 원 위의 실행 결과 숫자를 엽니다. 이 이미지는 모듈 위치 안내이며 수신 성공 결과 캡처는 아닙니다.

3. 이 단계는 새 일정이 입구까지 왔는지 확인하는 단계입니다. 주변 일정이 없거나 이동 충돌이 없으면 Telegram 메시지나 이메일이 오지 않아도 됩니다. A/B 경고는 6-1의 테스트로 확인합니다.

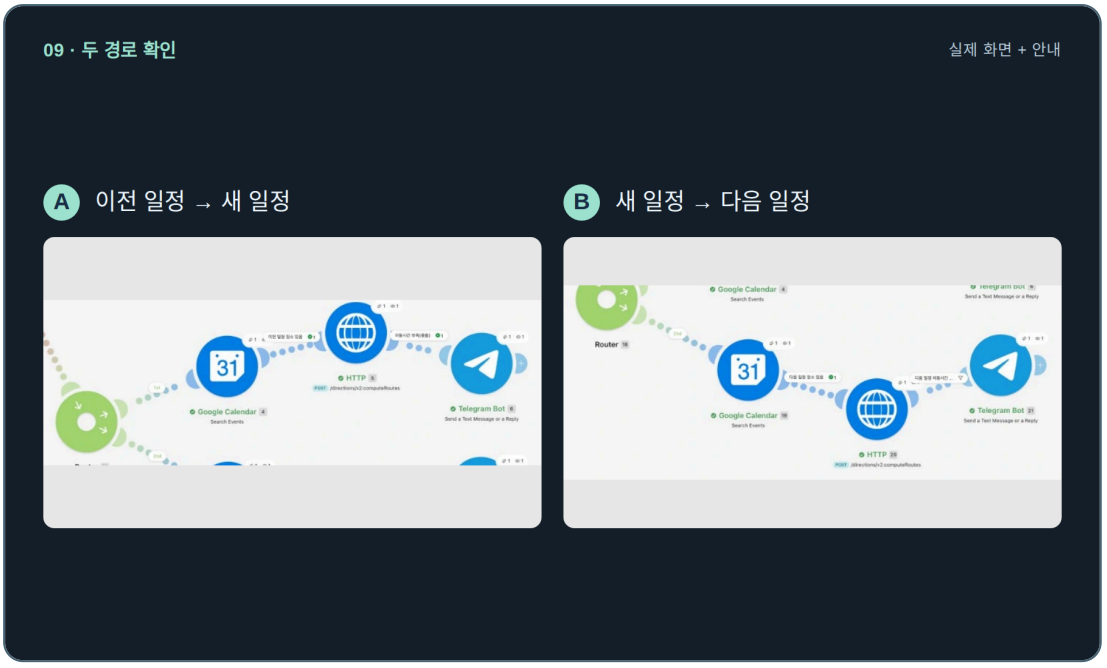
완료 확인 · testSend 샘플이 아니라 내가 방금 캘린더에 등록한 새 일정의 다섯 값이 Webhook 실행 결과에 있습니다.

두 경로와 알림 설정 읽기

가져온 설정을 위·아래로 나누어 확인합니다. 번호를 외울 필요는 없습니다.

5-1. A/B의 출발과 도착 구분

열 화면 · 위쪽 A와 아래쪽 B의 Calendar·HTTP



제공된 실제 A/B 화면 확대 · 원본 경로를 함께 확인하세요.

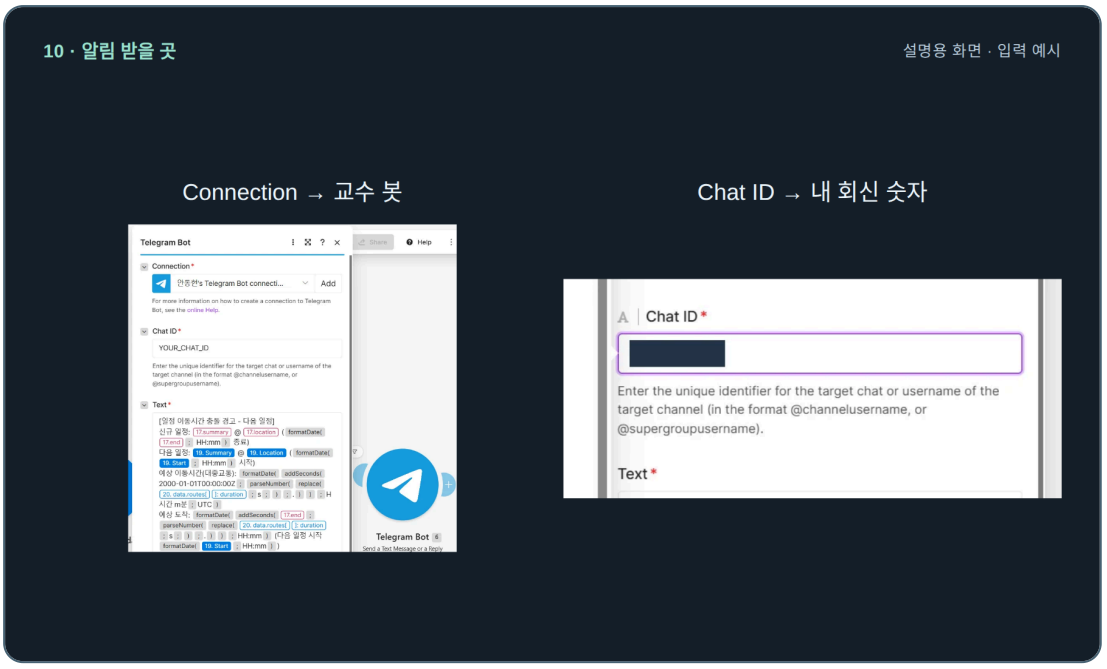
1. 아래 A/B 매핑 표를 옆에 두고 모듈을 하나씩 엽니다. A는 이전 일정 장소에서 새 일정 장소로, B는 새 일정 장소에서 다음 일정 장소로 이동합니다.
2. HTTP의 URL·POST·TRANSIT·출발 시각을 확인합니다. 기본 파일에 들어 있는 계산식은 유지하고, 연결과 교수 지도 설정을 확인합니다.

완료 확인 · A와 B에서 어느 일정의 종료 시각을 쓰는지 말할 수 있습니다.

5-2. 충돌 필터와 보내기 모듈 확인

열 화면 · HTTP → 알림 모듈 사이 연결선 · Telegram 또는 Gmail 2곳

> 이메일 버전은 HTTP → Gmail 사이의 충돌 필터를 확인합니다. Gmail 두 곳의 Connection은 본인 Gmail 연결, To는 본인이 알림을 받을 주소여야 합니다. Subject와 Content의 기존 매핑은 유지합니다.



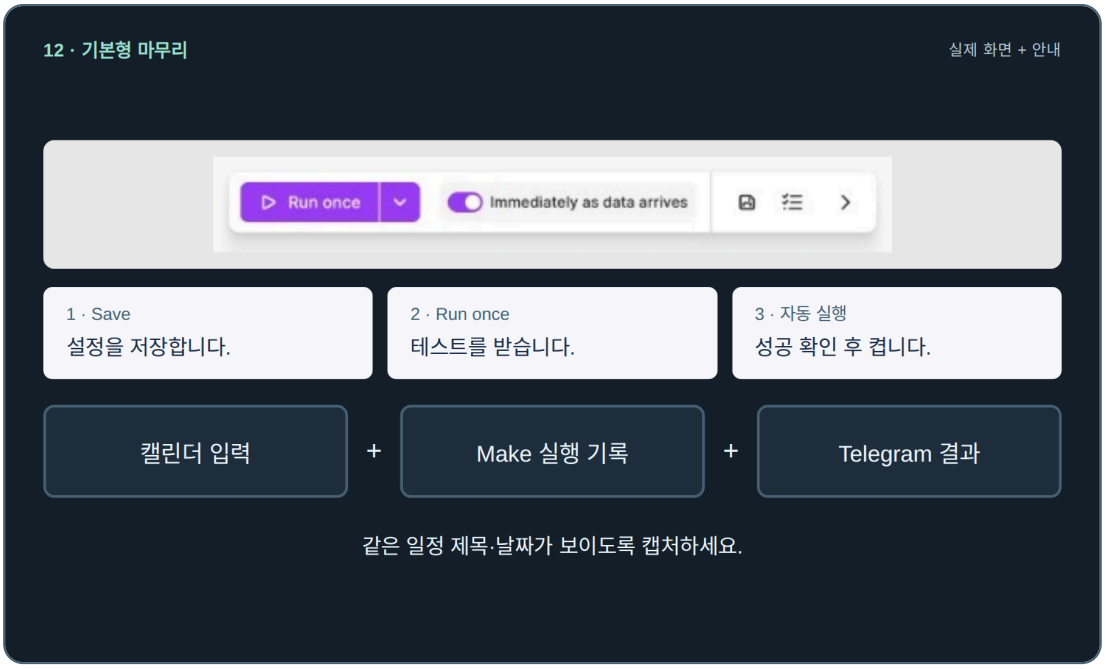
실제 Telegram 설정창 · Chat ID는 본인의 회신 숫자를 사용하세요.

1. 연결선의 필터를 열고 예상 도착 시각이 다음 시작 시각보다 늦을 때 통과하는지 확인합니다. 같거나 빠르면 기본형에서는 경고를 보내지 않습니다.
2. 텔레그램 버전은 두 곳의 Connection이 교수 봇용 기존 연결, Chat ID가 내 숫자인지 확인합니다. 이메일 버전은 두 Gmail의 Connection과 To를 확인합니다. 알림 본문의 일정 제목·시간 매핑은 지우지 않습니다.

완료 확인 · 조건 통과 후 선택한 개인 대화 또는 메일함으로 충돌 경고를 보내도록 되어 있습니다.

5-3. 저장하고 테스트 대기

열 화면 · Make 편집기 · Save → Run once



실제 실행 버튼 확대 · 기본형 완료 후 직접 수정한 결과도 캡처하세요.

1. 수정한 모듈의 저장/확인을 누른 뒤 시나리오 전체를 Save합니다.
2. 다음 장의 A/B/C 테스트를 하나씩 진행합니다. Run once가 끝나면 다음 입력 전에 다시 누릅니다. 저장했다는 것과 자동 실행이 켜졌다는 것은 다릅니다.

완료 확인 · 저장된 내 시나리오에서 테스트 입력을 받을 준비가 됐습니다.

확인할 값	A · 이전 → 신규	B · 신규 → 다음
일정 검색 범위	신규 시작 − 4시간 → 신규 시작	신규 종료 → 신규 종료 + 4시간
출발 장소	검색된 이전 일정의 장소	신규 일정의 장소
도착 장소	신규 일정의 장소	검색된 다음 일정의 장소
출발 시각	이전 일정 종료	신규 일정 종료
비교 시각	신규 일정 시작	다음 일정 시작

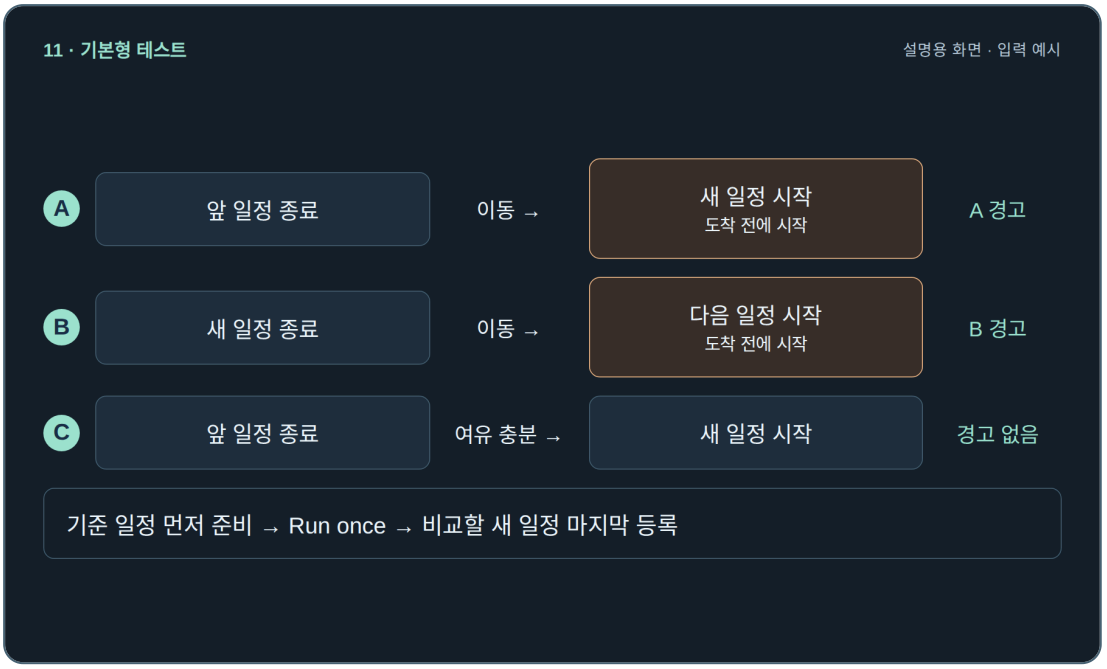
충돌 알림 확인과 과제 제출

충돌 알림을 확인하고 PPT로 정리합니다. 충돌 없는 일정의 안전 안내는 가산점 과제입니다.

6-1. A·B·C를 각각 새 일정으로 테스트

열 화면 · 수업용 Calendar · Make 실행 기록 · 알림 수신처

> 텔레그램 또는 7장에서 지정한 이메일로 결과를 확인합니다. 두 버전의 충돌 판정 기준은 같습니다.



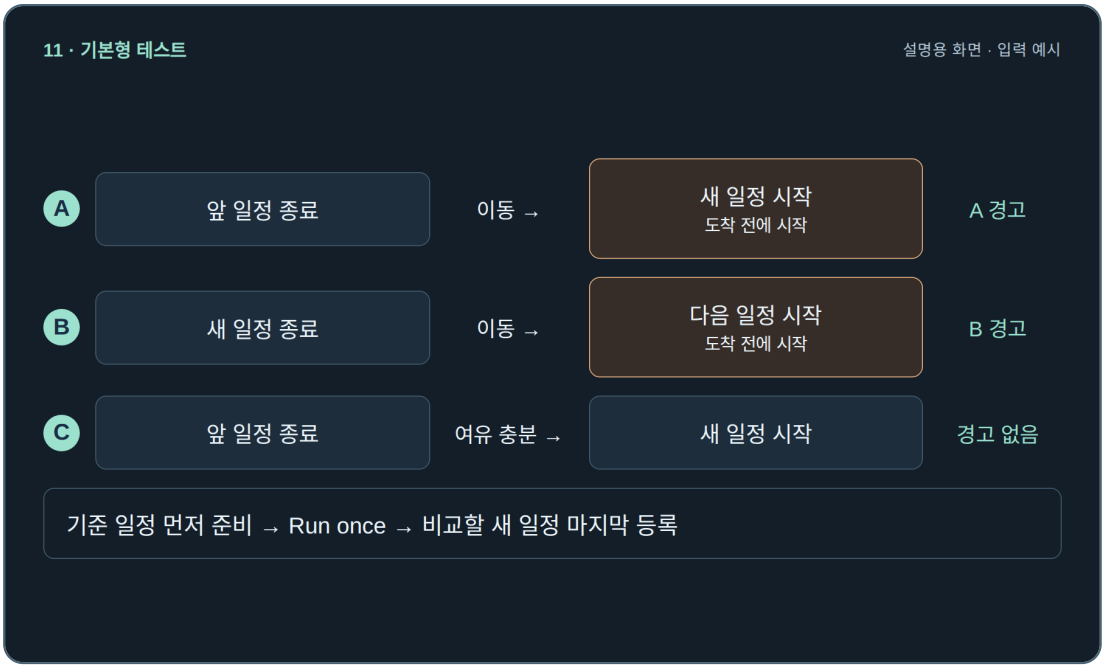
테스트 설명 그림 · 실제 경고 여부는 Routes 응답으로 확인합니다.

1. 아래 테스트 표에서 미래 날짜를 정합니다. A, B, C는 서로 다른 날짜로 나누고 제목 앞에 이름_A, 이름_B, 이름_C를 붙입니다.
2. 주변 기준 일정은 먼저 준비합니다. Make에서 수신 대기한 뒤 비교 대상이 될 새 일정을 마지막에 만듭니다. 일정 제목과 시각이 보이도록 생성 화면을 캡처합니다.
3. A와 B에서는 실제 이동시간으로 계산한 도착이 늦는지와 경고가 왔는지를 확인합니다. 충돌 없는 C는 기본형에서 경고가 없습니다. 충돌 알림이 도착한 메일함 또는 텔레그램 화면을 캡처합니다.

완료 확인 · 충돌하는 일정의 생성 화면과 그 일정에 대한 알림 수신 화면을 확보했습니다.

6-2. 가산점 · 충돌 없을 때 안전 일정 안내

열 화면 · 내 시나리오 · 선택 과제



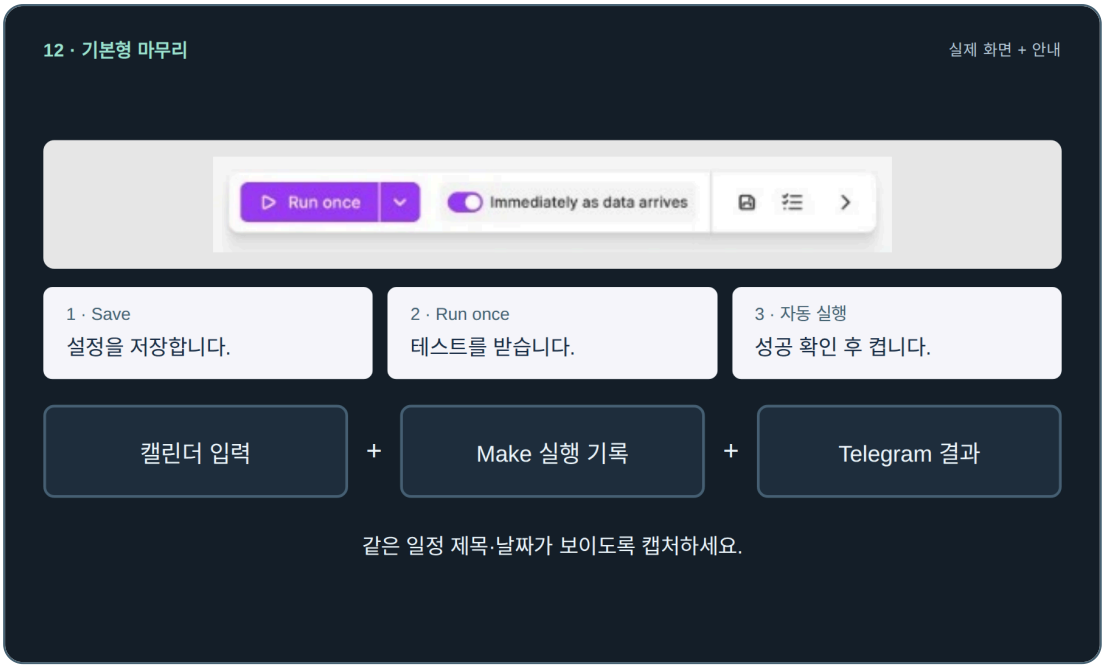
테스트 설명 그림 · 실제 경고 여부는 Routes 응답으로 확인합니다.

1. 기본형의 충돌 경고를 유지하면서, 충돌이 없을 때 안전 일정 안내가 오도록 직접 수정합니다. 텔레그램 또는 이메일 중 선택한 방법으로 받으면 됩니다.
2. 앞뒤 이동시간이 충분한 새 일정을 만들어 안전 일정 안내가 도착하는지 확인합니다. 구현했다면 일정 화면과 알림 수신 화면을 PPT 뒤에 추가합니다.

완료 확인 · 가산점에 도전했다면 충돌 없는 일정의 안전 일정 안내를 확인했습니다.

6-3. 시나리오 정리 + PPT 한 파일 메일 제출

열 화면 · Make 학생용 폴더 · PowerPoint · 메일



실제 실행 버튼 확대 · 기본형 완료 후 직접 수정한 결과도 캡처하세요.

1. 8장의 방법으로 내 시나리오를 학생용 폴더 과제#1 시나리오 위치에 넣고 자기 이름 라벨을 설정합니다.
2. Google Apps Script 코드 입력 화면 → Make 시나리오 완성 화면 → 충돌하는 일정 생성 화면 → 알림이 도착한 이메일 또는 텔레그램 화면 순서로 PPT 한 파일(.ppt 또는 .pptx)을 만듭니다.
3. 받는 사람을 ahn@dreamfurnace.com으로 입력하고 PPT 한 파일을 첨부해 보냅니다. 권장 파일명은 이름_과제1.pptx, 권장 메일 제목은 [모델매니지먼트교육 과제1] 이름입니다.
4. 마감은 2026년 10월 5일(월) 23:59:59 한국시간입니다. 교수 메일함에 표시된 도착 시각 기준이므로 여유 있게 보냅니다. 보내기 전에 받는 주소와 첨부 파일을 확인합니다.

완료 확인 · 학생용 폴더·이름 라벨을 설정했고 PPT 한 파일을 교수 메일로 보냈습니다.

테스트	입력 예시 · 날짜는 현재 이후	기본형의 기대 결과
A · 이전 이동 부족	미래 날짜 D: 강남역 14:00–15:00를 먼저 준비. 홍대입구역 15:01–16:00를 새로 등록.	A 경고. 실제 이동 응답으로 예상 도착 > 15:01인지 확인.
B · 다음 이동 부족	다른 미래 날짜: 홍대입구역 16:01–17:00를 먼저 준비. 강남역 15:00–16:00를 새로 등록.	B 경고. 실제 예상 도착 > 16:01인지 확인.

테스트	입력 예시 · 날짜는 현재 이후	기본형의 기대 결과
C · 여유 충분	다른 미래 날짜: 같은 장소의 앞뒤 일정 사이에 충분한 간격을 두고 새 일정을 등록.	경고 없음. 실제 이동 응답이 간격보다 짧고 필터가 멈춘 기록 확인.

07 / 09

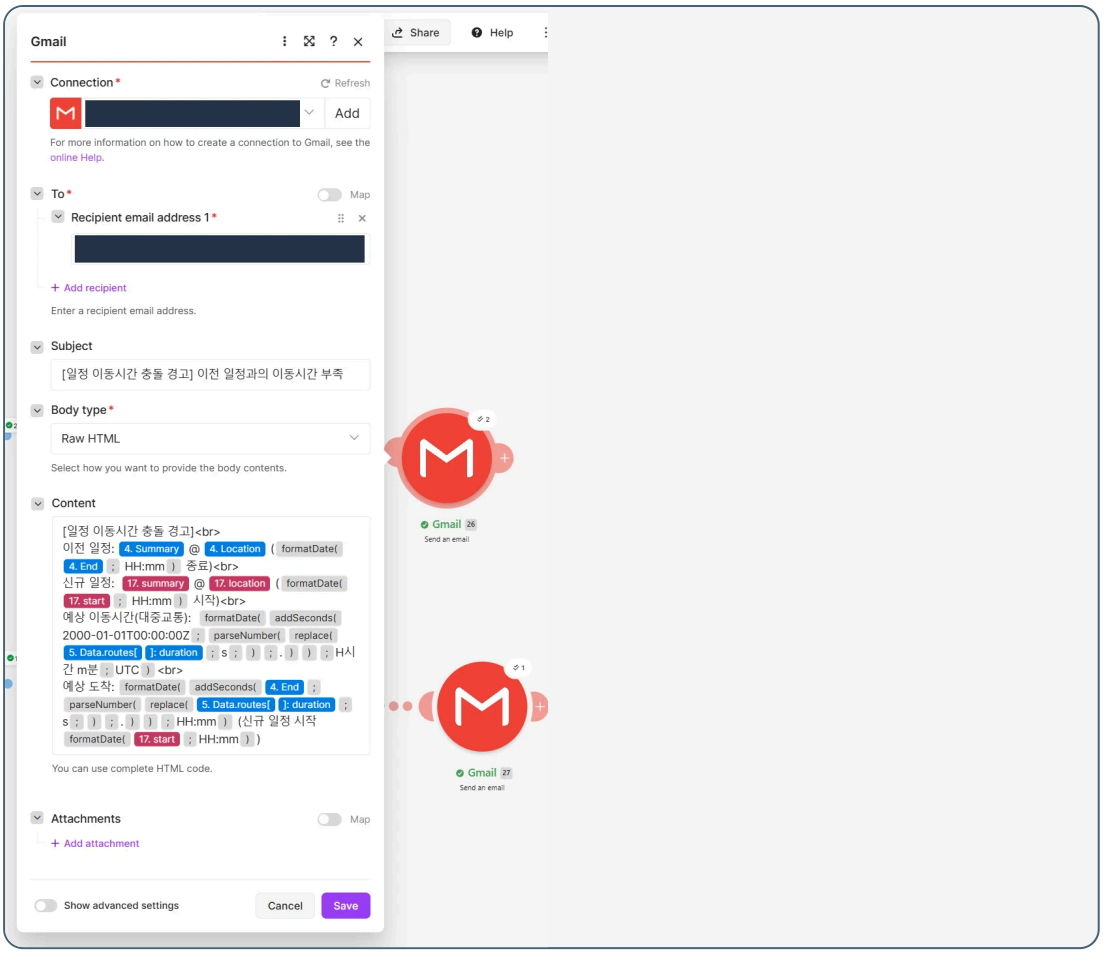
선택 · 텔레그램 대신 이메일로 받기

이메일 버전 블루프린트에는 마지막 두 곳에 Gmail이 들어 있습니다. Connection은 보내는 계정, To는 알림을 받는 주소입니다.

7-1. 이메일 버전의 Gmail 두 곳 찾기

열 화면 · Make · 이메일 버전 시나리오

1. 상단의 이메일 버전 블루프린트를 3-2 방법으로 새 시나리오에 가져옵니다. 이미 이메일 버전을 가져왔다면 다시 가져오지 않습니다.
2. 위쪽 A와 아래쪽 B의 마지막 모듈이 모두 Gmail · Send an email인지 확인합니다. 두 HTTP와 충돌 필터는 이미 들어 있습니다.



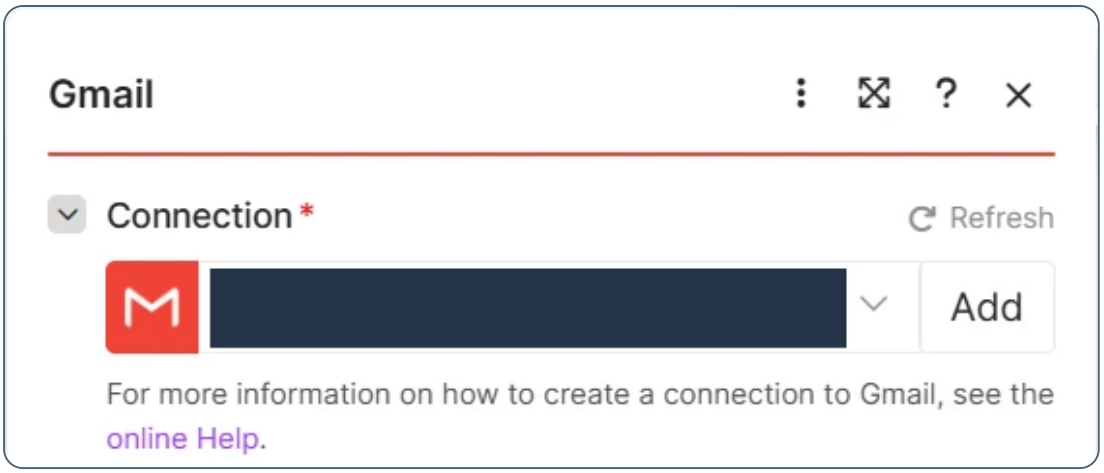
Notion 12번 실제 화면 · 위아래 마지막 모듈이 Gmail입니다. 예시 연결과 수신 주소는 가렸습니다. 각자 본인 값으로 설정합니다.

완료 확인 · 마지막 두 모듈이 Gmail인 내 시나리오를 열었습니다.

7-2. 본인 Gmail 계정 연결과 권한 승인

열 화면 · 위쪽 Gmail → Connection → Add

1. 위쪽 Gmail 원을 클릭합니다. Connection 옆 Add 또는 Create a connection을 누릅니다. 연결 이름을 본인이름_Gmail로 입력합니다.



같은 실제 화면의 Connection 확대 · 오른쪽 Add를 눌러 본인 Gmail을 연결합니다. Google 승인 창은 뒤이어 열립니다.

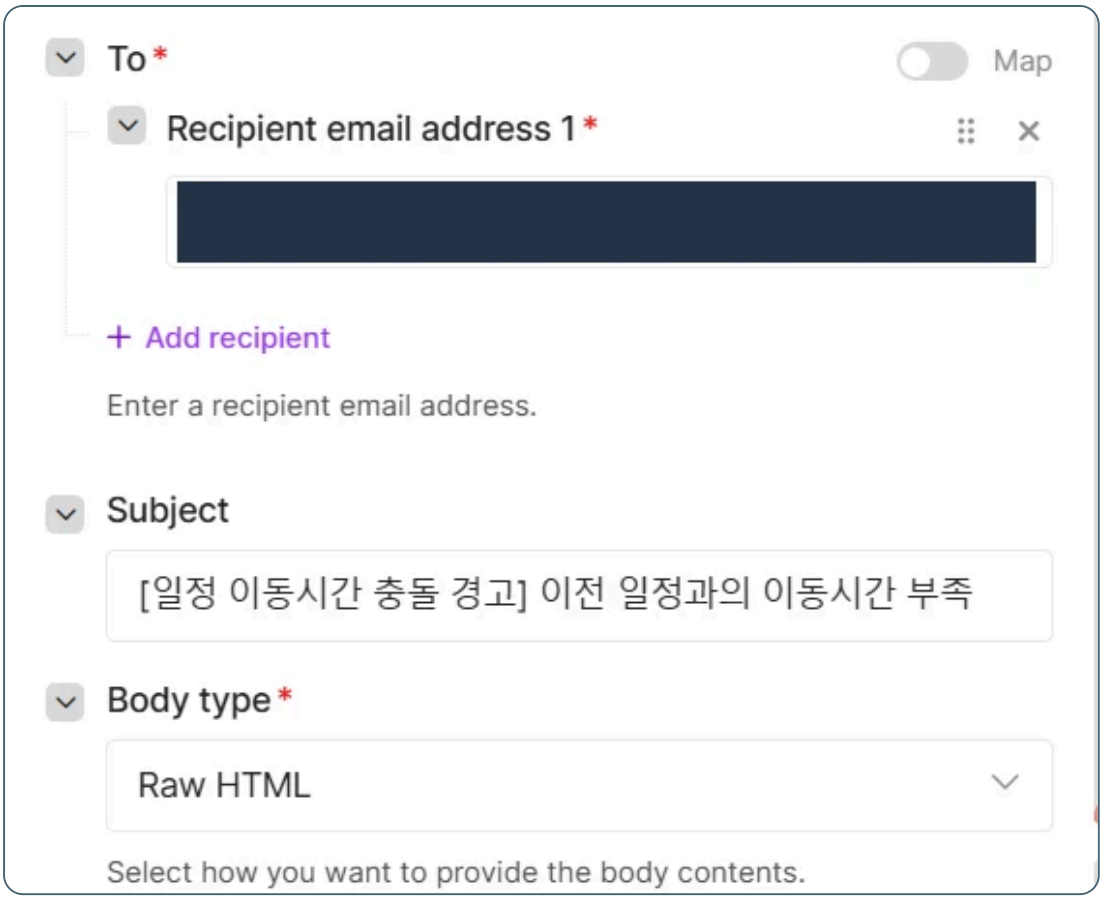
2. Sign in with Google을 클릭합니다. Google 계정 선택 창에서 메일을 보낼 본인 Gmail 계정을 선택합니다. Calendar 연결 때와 비슷하지만 Gmail에 대한 연결을 따로 승인하는 과정입니다.
3. Make에 연결할 계정의 이메일을 확인하고 계속(Continue)을 누릅니다. 요청하는 Gmail 권한을 확인하고, 권한 선택 항목이 나오면 메일 발송에 필요한 권한을 선택한 뒤 허용(Allow) 또는 계속을 누릅니다.
4. Make로 돌아오면 Connection에 방금 만든 본인의 Gmail 연결이 선택됐는지 확인합니다. 이 실습에서는 Additional scopes와 Advanced settings를 따로 입력하지 않습니다.

완료 확인 · 위쪽 Gmail의 Connection에 본인의 Gmail 계정이 연결됐습니다.

7-3. 받을 메일 주소 입력하고 아래쪽도 설정

열 화면 · Gmail 두 곳 · To / Subject / Body type / Content

1. To를 펼칩니다. Recipient email address 칸에 본인이 알림을 받고 싶은 메일 주소를 입력합니다. 입력칸이 없으면 Add a recipient를 누릅니다. 보내는 Gmail과 같은 주소로 받아도 되고, 본인이 사용하는 다른 메일 주소로 받아도 됩니다.



실제 To 입력 위치 · Recipient email address에 본인이 받을 주소를 넣습니다. Subject와 Raw HTML은 제공된 값을 유지합니다.

2. Subject는 메일 제목입니다. 위쪽은 이전 일정과의 이동시간 부족, 아래쪽은 다음 일정까지의 이동시간 부족으로 구분돼 있습니다. Body type은 Raw HTML, Content는 제공된 본문을 유지합니다.
는 줄바꿈 표시입니다.
3. Save(또는 OK)를 누릅니다. 아래쪽 Gmail도 열어 같은 본인의 Gmail Connection을 선택하고 To에 받을 주소를 입력합니다. 아래쪽 Content는 위쪽과 계산에 쓰는 값이 다르므로 복사해 덮어쓰지 않습니다.
4. 시나리오 전체를 Save합니다. 3-6으로 돌아가 내 Webhook을 만들고, 4장의 Apps Script 연결을 진행합니다. 이미 연결했다면 6-1의 테스트로 이동합니다.

항목	넣을 값
Connection	메일을 보낼 본인의 Gmail 연결
To	본인이 알림을 받고 싶은 메일 주소
Subject	제공된 A / B 경고 제목
Body type	Raw HTML
Content	각 경로의 일정·시간 매핑이 있는 기존 본문

완료 확인 · Gmail 두 곳에 본인 연결과 수신 주소를 입력하고 저장했습니다.

7-4. 충돌 경고 이메일 수신 확인

열 화면 · Make · Run once → Calendar → 수신 메일함

1. 6-1의 A와 B 테스트를 각각 진행합니다. Run once를 누른 뒤 충돌 조건에 맞는 새 일정을 만들고, 마지막 Gmail까지 실행됐는지 확인합니다.
2. To에 입력한 메일함을 엽니다. 받은편지함에서 [일정 이동시간 충돌 경고] 제목을 찾습니다. 보이지 않으면 스팸함도 확인합니다. 제목·장소·시작과 종료·예상 도착 시각이 내 테스트 일정과 맞는지 대조합니다.
3. C처럼 이동시간이 충분한 경우에는 기본형에서 경고 메일이 발송되지 않는 것이 맞습니다. Make 실행 기록에서 충돌 필터가 통과하지 않았는지 확인합니다.

완료 확인 · A/B 경고 메일을 받았고, C에서는 충돌 필터가 멈춘 것을 확인했습니다.

설정 참고: Make 공식 Gmail 연결 안내 · Send an email 필드 안내

08 / 09

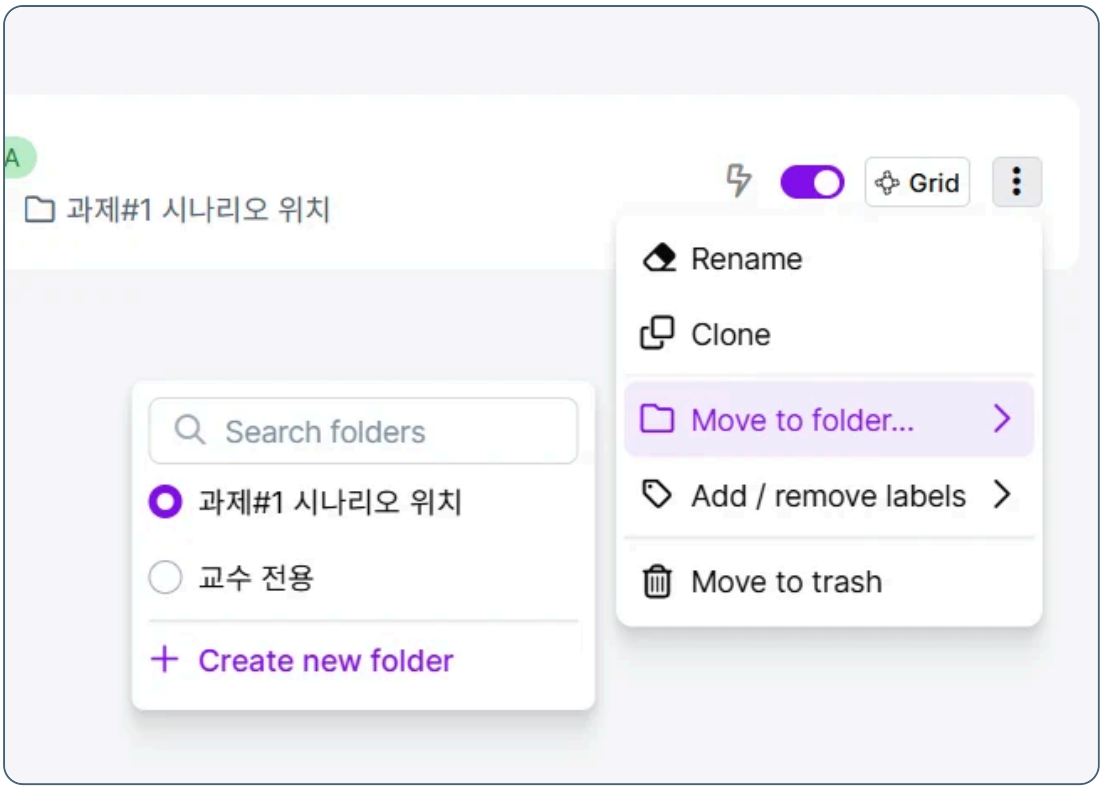
시나리오를 제출용 폴더에 넣고 이름 라벨 붙이기

텔레그램·이메일 버전 모두 진행합니다. Make의 폴더와 라벨은 수업 팀 안에서 본인 시나리오를 찾기 위한 표시입니다.

8-1. 제출용 폴더로 이동

열 화면 · Make · 수업 팀 → Scenarios 목록

1. 편집 화면에서 시나리오를 Save한 뒤 Scenarios 목록으로 돌아갑니다. 이름_과제1_일정알림처럼 본인 이름이 들어간 행을 찾습니다.
2. 본인 시나리오 행의 가장 오른쪽 ⋯ 버튼을 누릅니다. Move to folder…를 선택한 뒤 과제#1 시나리오 위치 폴더를 선택합니다.



Notion 13번 실제 메뉴 · Move to folder… → 과제#1 시나리오 위치를 선택합니다.

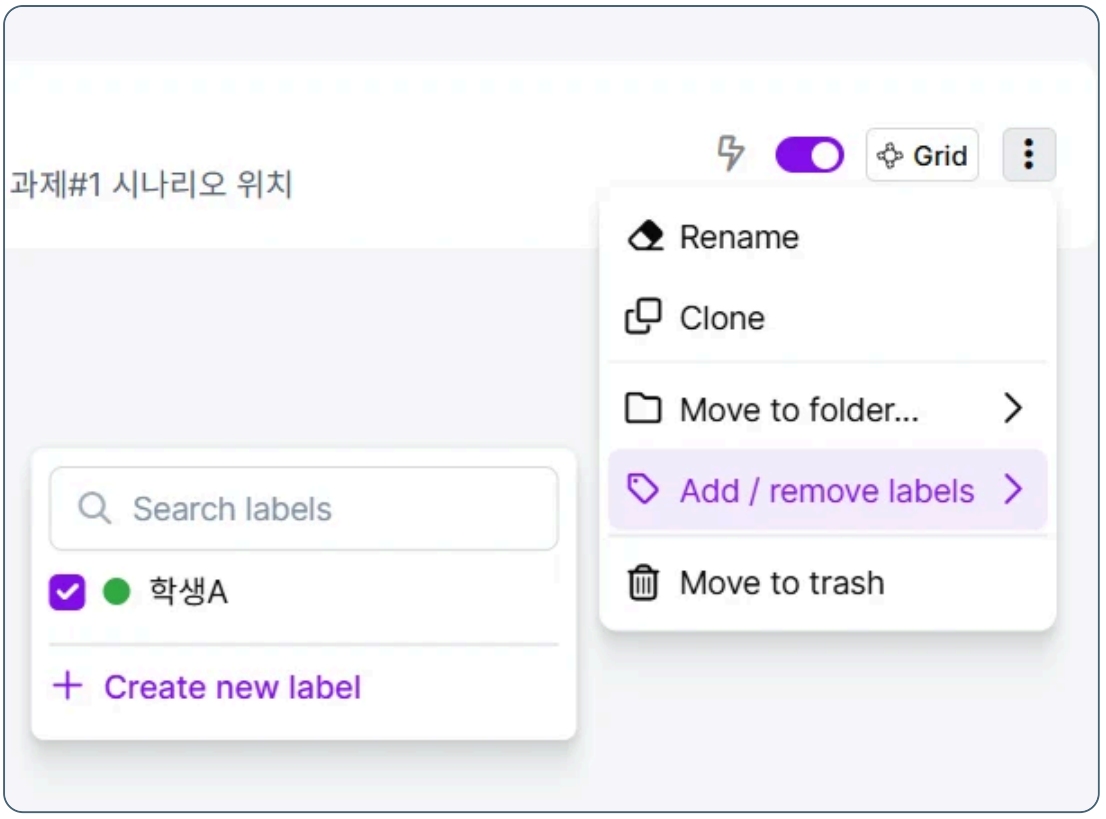
3. 왼쪽 폴더 목록에서 과제#1 시나리오 위치를 열고 방금 이동한 내 시나리오가 있는지 확인합니다.

완료 확인 · 내 시나리오가 과제#1 시나리오 위치 폴더에 있습니다.

8-2. 본인 이름 라벨 만들고 적용

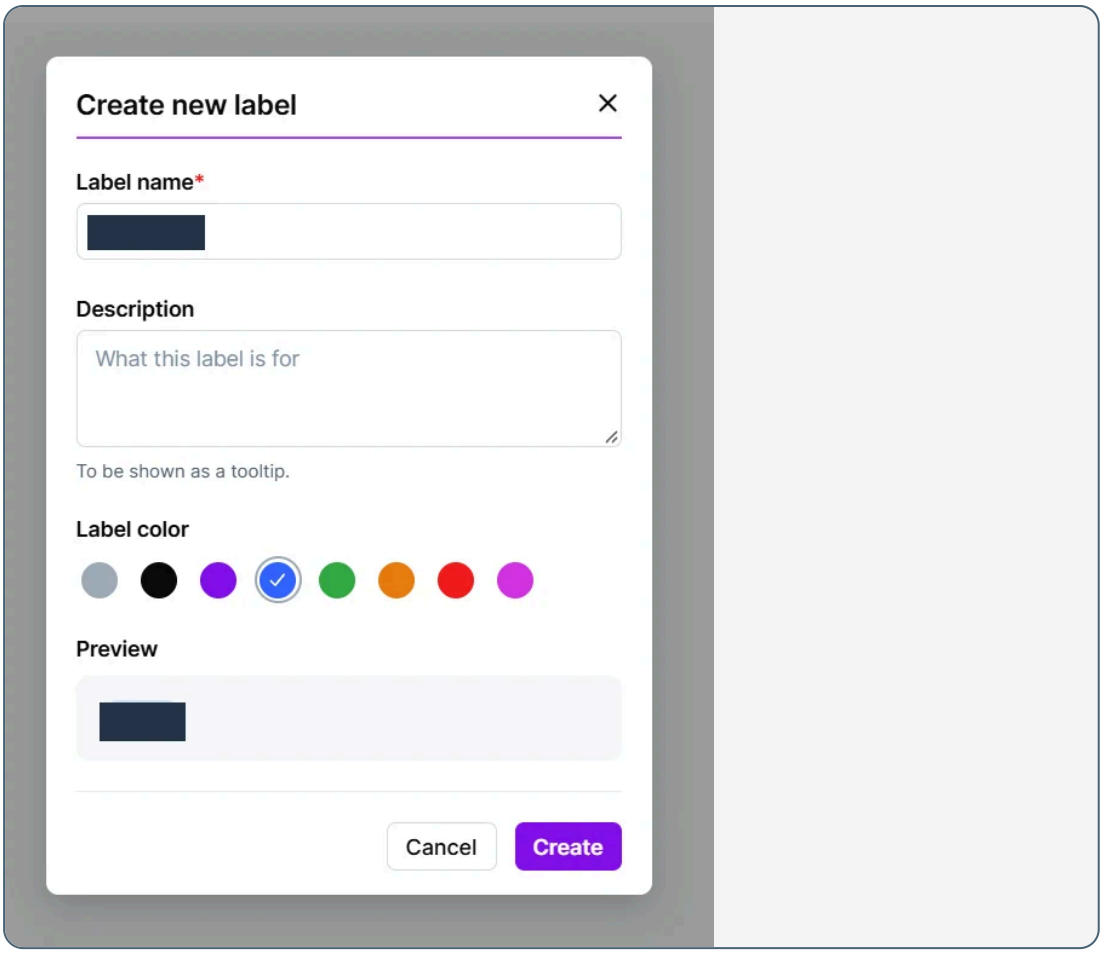
열 화면 · Make · 내 시나리오 행의 ⋯ 메뉴

1. 본인 시나리오 행 오른쪽 ⋯를 다시 누릅니다. Add / remove labels를 선택합니다.



실제 Add / remove labels 메뉴 · 캡처의 학생A 대신 본인 이름 라벨을 선택합니다. 없으면 Create new label을 누릅니다.

2. Search labels에서 본인 이름을 찾습니다. 본인 라벨이 없으면 Create new label을 누릅니다. 열린 창의 Label name에 본인 이름을 입력하고 색을 고른 뒤 Create를 누릅니다. Description은 비워 두어도 됩니다.



실제 Create new label 창 · Label name에 본인 이름을 쓰고 색을 고른 뒤 Create를 누릅니다. Description은 비워 두어도 됩니다. 예시 개인 이름은 가렸습니다.

3. 본인 이름 라벨을 체크해 적용합니다. 목록에서 내 시나리오에 이름 라벨이 표시되는지 확인합니다.

완료 확인 · 제출용 폴더 안의 내 시나리오에 본인 이름 라벨이 붙었습니다.

09 / 09

Active 켜기 · Run once 없이 자동 실행

테스트를 마친 뒤 자동 실행을 켭니다. Apps Script는 캘린더 변경을 전달하고, Make는 Webhook에 새 데이터가 들어올 때 실행합니다.

9-1. 즉시 실행 설정과 Active 스위치 확인

열 화면 · Make · 내 시나리오 편집기 → Scenarios 목록

1. Calendar·HTTP·알림 모듈의 설정과 A/B 테스트를 마친 내 시나리오를 Save합니다.

2. 편집기 아래 스케줄 설정이 Immediately as data arrives인지 확인합니다. 다른 값이라면 스케줄 설정을 열어 즉시 실행(Immediately)을 선택하고 저장합니다. 이 시나리오는 Custom webhook으로 받은 데이터가 실행 시작점입니다.
3. Scenarios 목록에서 본인 시나리오 행의 활성화 스위치를 켭니다. 상세 화면에서는 Inactive 스위치를 켜서 Active로 바꿀 수도 있습니다. 화면에 Active 또는 켜진 스위치가 표시되는지 확인합니다.



Notion 14번 실제 활성화 스위치 · 왼쪽 회색은 꺼짐, 오른쪽 보라색은 켜짐입니다. 본인 시나리오의 스위치를 켭니다.

완료 확인 · 내 시나리오의 즉시 실행 설정과 Active 상태를 확인했습니다.

9-2. Run once를 누르지 않고 새 일정으로 확인

열 화면 · 수업용 Google Calendar → Make History → 알림 수신처

1. 이번에는 Run once를 누르지 않습니다. 6-1의 A 또는 B 조건을 사용해 미래 날짜에 새 충돌 테스트 일정을 등록합니다. 제목에는 이름_자동실행확인을 넣습니다.
2. Make의 해당 시나리오 History에서 새 실행 기록을 열어 방금 등록한 일정 제목과 실행 결과를 확인합니다. 텔레그램 또는 이메일에서 그 일정의 충돌 경고가 왔는지도 확인합니다.
3. 사용 중인 브라우저 창을 계속 열어 둘 필요는 없습니다. 실습을 중단하려면 내 시나리오의 Active 스위치를 끕니다.

완료 확인 · Run once 없이 새 일정이 처리됐고, 선택한 수신처로 경고가 도착했습니다.

실행 설정 참고: Make 공식 스케줄 안내

과제 제출과 평가

기본 2점 + 가산점 0.5점 · 최대 2.5점

마감: 2026년 10월 5일(월) 23:59:59 한국시간 · 교수 메일함에 표시된 도착 시각 기준

제출할 것 두 가지

1. Make 시나리오를 학생용 폴더 **과제#1 시나리오 위치**로 이동하고, **자기 이름 라벨**을 설정합니다.
2. 각 단계의 스크린샷을 순서대로 넣은 **PPT 한 파일(.ppt 또는 .pptx)**을 교수 메일 ahn@dreamfurnace.com 으로 보냅니다.

PPT에 넣을 화면 순서

순서	캡처할 화면
1	Google Apps Script 코드 입력 화면
2	Make 시나리오 완성 화면
3	충돌하는 일정을 생성한 Google Calendar 화면
4	충돌 알림이 도착한 이메일 또는 텔레그램 화면

가산점 기능을 구현했다면 충돌 없는 일정과 안전 일정 안내 수신 화면을 뒤에 추가합니다. 캡처의 글자와 일정 제목·시각이 읽히게 넣고, Webhook URL·API 키·토큰은 가립니다.

권장 파일명: **이름_과제1.pptx**
권장 메일 제목: **[모델매니지먼트교육 과제1] 이름**

평가 기준

평가 항목	배점	확인 기준
기한 내 PPT 첨부 메일 도착	1	교수 메일함에 표시된 도착 시각 기준
Make 시나리오 파일 평가	0.5	학생용 폴더에 있는 본인 이름 라벨의 시나리오
충돌 알림 수신 스크린샷	0.5	PPT에 포함된 충돌 이메일 또는 텔레그램 수신 화면
안전 일정 안내 · 가산점	0.5	충돌 없을 때 '안전 일정 안내' 알림이 오도록 구현

텔레그램과 이메일은 같은 기준으로 평가합니다. 보내기 전에 받는 주소와 PPT 첨부를 확인하고, 교수 메일함에 기한 내 도착하도록 여유 있게 보냅니다.

기본형이 처리하는 범위

기본 코드는 생성 10분 이내이면서 장소와 시작·종료 시각이 있는 일정을 전달합니다. 장소 없는 일정과 종일 일정은 제외됩니다. 기본형은 충돌할 때만 경고하며, 충돌 없는 일정의 안전 일정 안내는 선택 가산점 기능입니다.

앞뒤 4시간 검색은 최대 10개이므로 직전·직후 한 건만 찾는 것은 아닙니다. 지도 응답이 없거나 API 오류가 난 경우를 안전한 일정으로 판단하지 않습니다.

수업 자료실: ddwu.dreamfurnace.com · 문의: ahn@dreamfurnace.com

실제 캡처: 교수 제공 Notion 자료 · 설정 근거: Make 팀과 연결 · Blueprint 가져오기 · Telegram 모듈 · Telegram 수신 방식 · Google 고급 서비스

2026-10-01 개정 · 컴퓨터의 체크 표시는 편의를 위한 개인 기록이며 교수에게 전송되지 않습니다.