

1. 노코드 vs 로우코드 실습

노코드(No Code) 실습

Airtable 실습 - 간단한 직원 관리 시스템 만들기

1. 실습 준비

- airtable.com 가입 및 로그인
- Workspaces > Workspace 선택
- **Create** 클릭
- **Build on app on your own** 선택

2. 데이터 구조 생성

- Base 이름 변경: **직원 관리**
- 테이블 이름 변경: **직원**
- Grid 이름 변경: **직원목록**
 - 사번(Number)
 - 이름(Single line text)
 - 부서(Single select): 개발, 마케팅, 영업, 인사
 - 직위(Single line text)
 - 이메일>Email)
 - 시작일(Date)
 - 상태(Single select): 재직, 휴직, 퇴직
 - 프로필 사진(Attachment)

3. '직원' 테이블에 예제 데이터 넣기(AI 프롬프트)

- AI 프롬프트

'직원' 테이블에 예제 데이터를 10개 추가해줘

4. 뷰 추가

- Create new...
 - Grid: **직원 목록**(기본)
 - Gallery: **직원 카드**
 - Calendar: **입사일 달력**
 - Using date field: **시작일** 선택
 - Kanban: **부서별 직원 현황**

Zapier 실습 - 신규 직원 온보딩 업무 자동화

1. 실습 준비

- zapier.com 가입 및 로그인
- Create > Zaps 클릭
- Rename: 신규 직원 온보딩

2. Trigger 추가: 새 직원 데이터가 추가될 때

- Trigger 클릭 > Airtable 선택
- Setup
 - App: Airtable
 - Trigger Event: New Record 선택
 - Account: 새 계정 추가 > Base 추가 > 직원 관리 선택 > Grant access 클릭
- Configure
 - Base: 직원 관리 선택
 - Table: 직원 선택
- Test
 - 직원 테이블의 최신 레코드 3개 확인

3. Action 추가: 환영 이메일 발송

- Action 클릭 > Gmail 선택
- Setup
 - App: Gmail
 - Action Event: Send Email 선택
 - Account: 계정 연결
- Configure
 - To: 신규 직원 이메일 매핑
 - From: 계정 이메일 선택
 - Subject: 환영합니다, {이름}님!
 - Body: 환영 메시지 작성
- Test
 - Skip Test
- 중간 테스트
 - Airtable: 신규 레코드 추가
 - Zaps: Run 클릭
 - Gmail: 이메일 수신 확인 > 제목과 내용 확인

4. Action 추가: HR 채널에 신규 직원 알림

- Edit Zap > Add Step > Slack 선택
- Setup
 - App: Slack
 - Action event: Send Channel Message 선택
 - Account: Slack 계정 연결
- Configure
 - Channel: #채널명 선택
 - Add Zapier app to channel automatically: No 선택
 - 슬랙 채널에 Zapier 앱이 등록되어 있어야 메시지 전송 가능

- Message Text: 이메일 관련 정보 매핑
 - From, Subject, Snippet 등
- Send as a bot: Yes 선택
 - No 선택 시: 사용자의 이름으로 메시지 전송
- Test
 - Skip Test
- Publish 선택

5. 자동화 테스트

- Airtable의 '직원' 테이블에 새 직원 추가
 - Zapier가 자동으로 이메일 발송 및 슬랙 알림 전송 확인

Zapier 실습 - 퇴사 처리 업무 자동화

1. 실습 준비

- airtable.com 에 로그인
 - 직원 테이블에 변경일(Last modified time) 탑입의 필드 추가
 - 이 필드가 있어야만 레코드의 변경 상태에 반응하여 Zap이 작동하도록 설정할 수 있음
- zapier.com 에 로그인
 - Zaps > Create > New Zap 클릭
 - Rename: 퇴사 처리

2. Trigger 추가: 직원의 레코드가 변경될 때

- Setup
 - App: Airtable 선택
 - Trigger Event: New or Updated Record 선택
 - Account: 계정 연결
- Configure
 - Base: 직원 관리 선택
 - Table: 직원 선택
 - Last modified time column: 변경일 선택
- Test
 - 직원 테이블의 최신 레코드 3개 확인

3. Filter 추가: 직원의 상태 필드가 퇴직으로 변경될 때

- Setup
 - App 변경: Filter 선택
- Configure & test
 - Field: 상태 선택
 - Condition: (Text) Exactly matches
 - Value: 퇴직

4. Action 추가: 퇴사 처리 체크리스트 이메일 발송

- Setup
 - App: **Gmail** 선택
 - Action Event: **Send Email** 선택
 - Account: 계정 연결
- Configure
 - To: 변경 직원 **이메일** 매핑
 - From: **계정 이메일** 선택
 - Subject: **{이름}님의 퇴사 절차를 안내드립니다.**
 - Body: 퇴사 절차 체크리스트 작성
- Test
 - Skip Test

5. Action 추가: IT팀에 계정 비활성화 요청

- Setup
 - App: **Slack** 선택
 - Action event: **Send Channel Message** 선택
 - Account: Slack 계정 연결
- Configure
 - Channel: **#채널명** 선택
 - Add Zapier app to channel automatically: **No** 선택
 - 슬랙 채널에 Zapier 앱이 등록되어 있어야 메시지 전송 가능
 - Message Text: 퇴사자의 **부서, 이름, 직위** 명시하여 작성
 - Send as a bot: **Yes** 선택
 - **No** 선택 시: 사용자의 이름으로 메시지 전송
- Test
 - Skip Test

6. Action 추가: 퇴사자 명단에 추가

- Setup
 - App: **Google Sheets** 선택
 - Action event: **Create Spreadsheet Row** 선택
 - Account: Google 계정 연결
- Configure
 - Drive: **My Google Drive** 선택
 - Spreadsheet: **퇴사자** 선택
 - Worksheet: **2025년** 선택
 - Row: 퇴사자의 **이름, 부서, 직위, 이메일, 퇴사일** 컬럼의 값 매핑
- Test
 - Skip Test
 - Publish 선택

7. 자동화 테스트

- Airtable의 **직원** 테이블에서 한 직원의 **상태**를 **퇴사**로 변경
 - Zapier가 실행되는 것을 확인

로우코드(Low Code) 실습

Appsmith 실습 - 직원 CRUD 앱 만들기

1. 실습 준비

- appsmith.com 회원 가입 및 무료 시작하기
 - Business plan trial: 15일 무료
- 워크스페이스 생성
 - 이름: **study**
- 애플리케이션 만들기
 - **Create New** 클릭
 - **Application** 선택
 - 이름 변경: (화면 상단) **study / Employee App**

2. 직원 목록 화면 만들기

- 데이터소스 연결하기
 - 원쪽 사이드바에서 **Datasources** 아이콘 클릭
 - **Bring your data** 클릭
 - **Sample Datasources / users** 선택
- 쿼리 구성하기
 - 오른쪽 상단 + **New query** 선택
 - **Datasources: users** 선택
 - **Run** 클릭하여 기본 생성된 쿼리 실행
 - 직원 목록 화면에 출력할 컬럼 지정

```
SELECT id, name, email, phone, image FROM public."users" LIMIT 10;
```

- **Run** 클릭하여 쿼리 실행
- 이름 오름차순으로 정렬

```
SELECT id, name, email, phone, image FROM public."users" ORDER BY name ASC LIMIT 10;
```

- **Run** 클릭하여 쿼리 실행
- Query 이름 변경: **getUsers**
- UI 구성하기
 - 원쪽 상단에서 **UI** 아이콘 클릭
 - **New UI element** 클릭
 - Table 위젯을 화면에 추가 및 크기 조정
 - 오른쪽 속성 창
 - **Table data: getUsers** 쿼리 선택
 - **Columns: id, image, name, email, phone** (드래그하여 순서 변경)

- 애플리케이션 배포
 - (화면 왼쪽 상단) Deploy 클릭
 - 화면 동작 확인
- 직원 목록 화면에서 사진 표시하기
 - Table 위젯 선택
 - 오른쪽 속성 창에서 Columns 설정
 - image 컬럼 > 설정 > Column type: Image 선택
- 테이블 위젯 이름 변경
 - 왼쪽 페이지 위젯 목록에서 Table1 선택
 - 설정 > Rename: employeeTable
- 페이지 이름 변경
 - 왼쪽 상단 페이지 정보에서 Page1 설정 클릭
 - 이름 변경: Employee List
- 페이지 제목 위젯 추가
 - 페이지 UI 편집 화면 > + New UI element > Text 드래그하여 페이지 화면 상단에 추가
 - 속성 창:
 - Content > Text: 직원 목록
 - Style > Font size: XL 선택
 - 위젯 이름 변경: pageTitle
- 페이지 이동 처리하기
 - employeeTable > Pagination 속성
 - Show pagination: True
 - Server side pagination: True
 - Query 변경: Queries > getUsers 선택

```

SELECT id, name, email, phone, image
FROM public."users"
ORDER BY name ASC LIMIT {{employeeTable.pageSize}}
OFFSET {{(employeeTable.pageNo - 1) * employeeTable.pageSize}};

```

3. 직원 상세 정보 화면 만들기

- 새 페이지 추가
 - 왼쪽 상단 페이지에서 [+]를 클릭하여 새 페이지 추가: New blank page 선택
 - 페이지 이름 변경: (상단 페이지 정보에서) Employee Details
- 쿼리 구성하기
 - 오른쪽 상단 + New query 선택
 - Datasources: users 선택
 - Query 이름 변경: getUserId
 - 목록에서 선택한 직원의 id로 상세 정보 조회

```

SELECT * FROM public."users" WHERE id=1;

```

- Run 클릭하여 쿼리 실행: id가 1번인 직원 정보 확인

- UI 구성하기
 - 원쪽 상단에서 **UI** 아이콘 클릭
 - **New UI element** 클릭
 - JSONForm 위젯을 화면에 추가 및 크기 조정
 - 위젯 이름 변경: **employeeForm**
 - 오른쪽 속성 창
 - General > Title: 직원 상세 정보
 - Data > Source data: `{{getUserById.data[0]}}` 입력
 - **Generate form** 클릭
 - 항목 감추기: **Latitude, Longitude**
 - 날짜 컬럼 포맷 변경
 - **Dob** 컬럼
 - Date format: YYYY-MM-DD
 - **Created At** 컬럼:
 - Field Type: **Datepicker**
 - Date format: YYYY-MM-DD HH:mm
 - **Updated At** 컬럼:
 - Field Type: **Datepicker**
 - Date format: YYYY-MM-DD HH:mm
 - **Image** 컬럼
 - General Visible: **false**
 - 사진 출력 위젯 추가
 - UI > Media > **Image** 드래그하여 Form 왼쪽에 추가
 - 위젯 이름 변경: **profileImage**
 - Data > **Image** 속성: `{{employeeForm.formData.image}}`
 - **employeeForm** 위젯 > 속성 창
 - **Image** 컬럼
 - General Visible: **false**
 - **profileImage** 위젯 > 속성 창
 - Data > **Image** 속성 변경: `{{getUserById.data[0].image}}`
 - **employeeForm**에서 **Image** 컬럼을 삭제했기 때문

4. 목록 화면과 상세 정보 화면을 연결하기

- Employee List 페이지: 액션 추가
 - **employeeTable** 위젯 > **name** 컬럼 설정
 - Column type: **Button** 선택
 - Text: `{{currentRow.name}}`
 - onClick 추가
 - Action: **Navigate to**
 - Type: **Page name**
 - Choose page: **Employee Details**
 - Query params
 - `id: currentRow.id`
 - Target: **Same window** 선택
 - Employee Details 페이지: Query 변경
 - Queries > **getUserById** 변경:

```
SELECT * FROM public."users"
WHERE id={{appsmith.URL.queryParams.id}};
```

- Deploy 및 테스트

5. 직원 등록 화면 만들기

- 새 페이지 추가
 - 왼쪽 상단 페이지에서 [+]를 클릭하여 새 페이지 추가: **New blank page** 선택
 - 페이지 이름 변경: (상단 페이지 정보에서) **Employee Form**
- 쿼리 구성하기
 - 오른쪽 상단 + **New query** 선택
 - Datasources: **users** 선택
 - Query 이름 변경: **getUserById**
 - 목록에서 선택한 직원의 id로 상세 정보 조회

```
SELECT name, gender, dob, phone, email, image, country
FROM public."users"
WHERE id=451;
```

- **Run** 클릭하여 쿼리 실행: id가 1번인 직원 정보 확인

- UI
 - 왼쪽 상단에서 **UI** 아이콘 클릭
 - **New UI element** 클릭
 - 제목 위젯 추가
 - Text 드래그하여 페이지 화면 상단에 추가
 - 속성 창:
 - Content > Text: **직원 등록**
 - Style > Font size: **XL** 선택
 - 위젯 이름 변경: **pageTitle**
 - 폼 위젯 추가
 - UI > Forms > JSONForm 드래그하여 페이지에 추가
 - 위젯 이름 변경: **employeeForm**
 - 오른쪽 속성 창:
 - General > Title: **기본 정보**
 - Data > Source data: **{{getUserById.data[0]}}**
 - **Generate form** 클릭
 - 컬럼 정렬: **name, email, phone, image, gender, dob, country**
 - 컬럼 타입 설정
 - Name(Text Input, Required)
 - Email>Email Input, Required)
 - Phone(Phone Number Input, Required)
 - Image(Text Input, Required)
 - Gender(Select)

- [{"label": "남자","value": "male"}, {"label": "여자","value": "female"}]
- Dob(Datepicker)
 - Date format: YYYY-MM-DD
- Country(Text Input)

6. 직원 목록 화면에서 직원 등록 화면으로 이동하기

- Employee List 페이지: 액션 추가
 - UI > Buttons > Button 드래그하여 페이지에 추가
 - 위젯 이름 변경: addEmployeeButton
 - 속성 창:
 - Label: 직원 등록
 - onClick 추가
 - Action: Navigate to
 - Type: Page name
 - Choose page: Employee Form
 - Target: Same window 선택
- Deploy 및 테스트
- Employee Form 페이지: 예제 데이터 제거
 - EmployeeForm 위젯 > 속성 창
 - Data > Source data 값 삭제
 - Deploy 및 테스트
 - 직원 등록 폼의 각 항목 값이 비어 있는지 확인

7. 직원 등록 폼 입력 값 저장하기

- 쿼리 구성하기
 - 오른쪽 상단 + New query 선택
 - Datasources: users 선택
 - Query 이름 변경: insertUser
 - 입력폼의 값을 DB에 저장

```
INSERT INTO public."users" (name, gender, dob, phone, email,
image, country)
VALUES ({{employeeForm.formData.name}},
{{employeeForm.formData.gender}},
{{employeeForm.formData.dob}}::date,
{{employeeForm.formData.phone}}, {{employeeForm.formData.email}},
{{employeeForm.formData.image}},
{{employeeForm.formData.country}});
```

- employeeForm: onSubmit 이벤트 추가
 - employeeForm 위젯 선택
 - 오른쪽 속성 창:
 - Events > onSubmit 추가
 - Action: Execute a query 선택 > insertUser 선택
 - Callbacks action 추가: On success 이벤트

- Action: **Navigate to**
- Type: **Page name**
- Choose page: **Employee List**
- Target: **Same window**
- Callbacks action 추가: On failure 이벤트
 - Action: **Show alert**
 - Message: 입력 실패!
 - Type: **Error**
- Deploy 및 테스트