

인포그래픽(순서도) 캡셔닝 데이터 규격서

2024-11-04

v0.5

버전 정보

- v0.2
 - 색상 hex code로 표기하기로 논의된 바 있어 이를 반영함
 - 두께를 별도의 thickness 클래스로 구분함
- v0.3
 - sub_blocks → node
 - id를 문자열 표현으로 표기 (쌍따옴표로 묶어 문자열로 표현한 정수) ID가 123인 경우, "123"으로 변환되어 문자열로 표현됩니다.
 - 각 노드의 정보를 담은 title, text는 text로 병합
 - 모든 key에 대응하는 value는 not null이며 만약 해당하는 값이 없다면 비어있는 값으로 표기
 - direction은 논리자료형으로 표기
- v0.4
 - 예시 추가
- v0.5
 - 예시 수정

1. 파일 정보

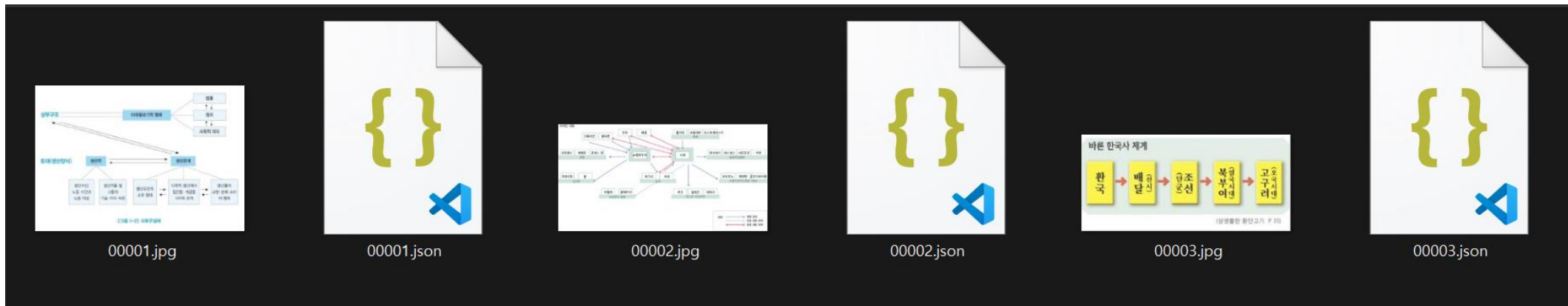
1. 파일 정보

1.1. 파일명

- 각 파일은 원천데이터(jpg)와 페어링된 라벨링데이터(json)로 구성됩니다.
- 페어링된 두 파일명은 동일합니다. (예: 00003.jpg -> 00003.json)
- 각 파일명은 00001부터 시작합니다. (예: 00001.jpg, 00002.jpg, ...)

이미지 포맷:
jpg, png, jpeg 등
이미지를 로드할 수 있는 범용적인 포맷은
모두 허용

(검토 필요)
해상도
눈에 글자만 보이면 됨(내부 수집기준)
숫자로 표현하는 방법을 고민해보자



2. 데이터의 타입

2. 데이터의 타입

No	Key		Description	Type	필수여부	종류	비고	
1	file_name		이미지 파일 이름	String	Y		jpg, png, jpeg 등 이미지를 로드할 수 있는 범용적인 포맷은 모두 허용	
2	summary		이미지에 해당하는 요약	String	Y			
3	components		구성 요소 파트 (노드 정보)	String	Y			
4		id	고유식별자	String	Y		Key(고유값), Not null (예: "1", "2", "12")	
5		text	해당 노드의 텍스트	String	Y			
6		node	하위 노드	JsonArray	Y		하위노드가 없다면 비어있는 배열로 표기 (예: "node": [])	
7	connections		연결 정보: 순서도 내 노드들 간의 연결 정보	JsonArray	Y			
8		from	출발 노드의 id	String	Y		간혹 노드가 없는 영역으로부터 출발할 경우 비어있는 문자열로 표기	
9		to	목적 노드의 id	String	Y		간혹 노드가 없는 영역으로 향해있을 경우 비어있는 문자열로 표기	
10		text	연결에 대한 설명	String	Y		설명이 없는 단순 연결이라면 비어있는 문자열로 표기	
11		type		연결 유형	String	Y	line	
							dashed	
							dotted	
							unknown	
12		color	연결선의 색상	String	Y	Hex code		
13		direction	연결의 방향	Boolean	Y	TRUE		
	FALSE							
14	thickness	연결의 두께 (연결의 강도)	String	Y	thin	default = medium if 선의 종류 2개 -> medium, thick 사용 if 3개 -> thin, medium, thick 사용		
					medium			
					thick			

3. JSON 파일 생성 규칙

3. JSON 파일 생성 규칙

3.1. JSON 기본 구조

각 JSON 파일은 아래와 같은 기본 구조를 따라야 합니다

```
{
  "file_name": "<이미지 파일명>",
  "summary": "<이미지에 해당하는 요약>",
  "components": [
    {
      "id": "<고유 식별자, 자연수>",
      "text": "<구성 요소 설명>",
      "node": [
        {
          "id": "<고유 식별자, 자연수>",
          "text": "<구성 요소 설명>",
          "node": []
        }
      ]
    }
  ],
  "connections": [
    {
      "from": "<출발 node의 ID>",
      "to": "<목적 node의 ID>",
      "text": "<연결선 설명>",
      "type": "<연결선 유형>",
      "color": "<연결선 색상>",
      "direction": "<연결선 방향(arrow) 유무>",
      "thickness": "<연결선 두께>"
    }
  ]
}
```

NEXT) 부분 별 설명은 3.2.부터 >>

3. JSON 파일 생성 규칙

3.2 파일 정보

각 JSON 파일의 기본 정보

3.2.1. file_name (문자열)

- 해당 JSON 파일과 페어링된 이미지 파일의 이름 (예: "00003.jpg")

3.2.2. summary (문자열)

- 해당 이미지와 관련된 요약 정보. 순서도의 목적이나 기능을 간략하게 설명

```
{
  "file_name": "<이미지 파일명>",
  "summary": "<이미지에 해당하는 요약>",
  "components": [
    {
      "id": "<고유 식별자, 자연수>",
      "text": "<구성 요소 설명>",
      "node": [
        {
          "id": "<고유 식별자, 자연수>",
          "text": "<구성 요소 설명>",
          "node": []
        }
      ]
    }
  ],
  "connections": [
    {
      "from": "<출발 node의 ID>",
      "to": "<목적 node의 ID>",
      "text": "<연결선 설명>",
      "type": "<연결선 유형>",
      "color": "<연결선 색상>",
      "direction": "<연결선 방향(arrow) 유무>",
      "thickness": "<연결선 두께>"
    }
  ]
}
```

3. JSON 파일 생성 규칙

3.3 라벨링 정보 - 노드

이미지로부터 라벨링된 데이터

3.3.1. components (배열)

- 순서도의 주요 구성 요소를 포함, 각 요소는 여러 개의 서브 노드를 가짐
- id (문자열): 구성 요소의 고유 식별자
- text (문자열): 구성 요소의 텍스트, 비어 있을 경우 빈 문자열로 표시
- node (배열): 구성 요소의 하위 노드

3.3.2. ID 규칙

- id는 각 구성 요소와 하위 노드의 고유성을 보장하기 위해 반드시 유일해야 함.
- 정수형 ID를 문자열로 변환 표기 (예시: "1", "2", "12")
- ID가 123인 경우, "123"으로 변환되어 문자열로 표기

3.3.3. text

- 해당 구성 요소에 해당하는 모든 텍스트를 포함

3.3.4. node

- node는 계층적인 구조로, 각 구성 요소는 필요에 따라 여러 하위 노드를 가질 수 있음
- node 배열 내 하위 노드가 없을 경우 빈 배열([])으로 표시

```
{
  "file_name": "<이미지 파일명>",
  "summary": "<이미지에 해당하는 요약>",
  "components": [
    {
      "id": "<고유 식별자, 자연수>",
      "text": "<구성 요소 설명>",
      "node": [
        {
          "id": "<고유 식별자, 자연수>",
          "text": "<구성 요소 설명>",
          "node": []
        }
      ]
    }
  ],
  "connections": [
    {
      "from": "<출발 node의 ID>",
      "to": "<목적 node의 ID>",
      "text": "<연결선 설명>",
      "type": "<연결선 유형>",
      "color": "<연결선 색상>",
      "direction": "<연결선 방향(arrow) 유무>",
      "thickness": "<연결선 두께>"
    }
  ]
}
```

3. JSON 파일 생성 규칙

3.4 라벨링 정보 - 연결(connections)

이미지로부터 라벨링된 데이터

3.4.1. connections (배열)

- 시스템 내 구성 요소들 간의 연결 정보
- from (문자열): 출발 구성 요소의 ID
- to (문자열): 연결될 목적 구성 요소의 ID
- text (문자열): 연결에 대한 설명
- type (문자열): 연결 유형 (예: "line", "dotted")
- color (문자열): 연결 선의 색상 (예: "#FF0000", "#000000")
- direction (논리 자료형(boolean)): 연결의 방향성 유무 (예: true, false)
- thickness (문자열): 연결선의 두께 (예: "thin", "thick")

type의 종류

- **line**: 기본 직선 연결.
- **dashed**: 점선 연결.
- **dotted**: 점으로 이루어진 연결.
- **unknown**: 분류되지 않음

color의 종류

- Hex code로 표현

direction의 종류

- **true**: 화살표가 있는 모든 연결
- **false**: 방향성 없음

thickness의 종류

- **thin**: 얇은 선
- **medium**: 중간 두께의 선
- **thick**: 두꺼운 선
- **extra-thick**: 매우 두꺼운 선.

```
{
  "file_name": "<이미지 파일명>",
  "summary": "<이미지에 해당하는 요약>",
  "components": [
    {
      "id": "<고유 식별자, 자연수>",
      "text": "<구성 요소 설명>",
      "node": [
        {
          "id": "<고유 식별자, 자연수>",
          "text": "<구성 요소 설명>",
          "node": []
        }
      ]
    }
  ],
  "connections": [
    {
      "from": "<출발 node의 ID>",
      "to": "<목적 node의 ID>",
      "text": "<연결선 설명>",
      "type": "<연결선 유형>",
      "color": "<연결선 색상>",
      "direction": "<연결선 방향(arrow) 유무>",
      "thickness": "<연결선 두께>"
    }
  ]
}
```

3. JSON 파일 생성 규칙

3.4 라벨링 정보 - 연결 - type

이미지로부터 라벨링된 데이터

3.4.1. connections (배열)

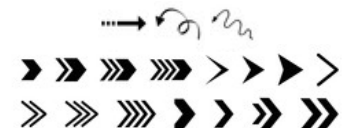
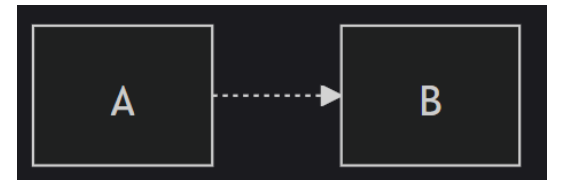
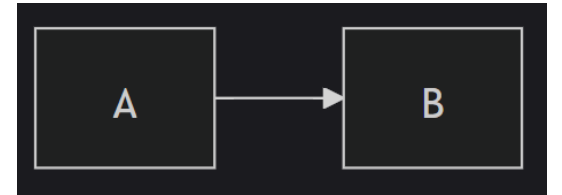
- 시스템 내 구성 요소들 간의 연결 정보
- from (문자열): 출발 구성 요소의 ID
- to (문자열): 연결될 목적 구성 요소의 ID
- text (문자열): 연결에 대한 설명

• **type (문자열): 연결 유형 (예: "line", "dotted")**

- color (문자열): 연결 선의 색상 (예: "#FF0000", "#000000")
- direction (논리 자료형(boolean)): 연결의 방향성 유무 (예: true, false)
- thickness (문자열): 연결선의 두께 (예: "thin", "thick")

type의 종류

- **line**: 기본 직선 연결.
- **dashed**: 점선 연결.
- **dotted**: 점으로 이루어진 연결.
- **unknown**: 분류되지 않음



3. JSON 파일 생성 규칙

3.4 라벨링 정보 - 연결 - color

이미지로부터 라벨링된 데이터

3.4.1. connections (배열)

- 시스템 내 구성 요소들 간의 연결 정보
- from (문자열): 출발 구성 요소의 ID
- to (문자열): 연결될 목적 구성 요소의 ID
- text (문자열): 연결에 대한 설명
- type (문자열): 연결 유형 (예: "line", "dotted")

• **color (문자열): 연결 선의 색상 (예: "#FF0000", "#000000")**

- direction (논리 자료형(boolean)): 연결의 방향성 유무 (예: true, false)
- thickness (문자열): 연결선의 두께 (예: "thin", "thick")

color의 종류

- Hex code로 표현

3. JSON 파일 생성 규칙

3.4 라벨링 정보 – 연결 – direction

이미지로부터 라벨링된 데이터

3.4.1. connections (배열)

- 시스템 내 구성 요소들 간의 연결 정보
- from (문자열): 출발 구성 요소의 ID
- to (문자열): 연결될 목적 구성 요소의 ID
- text (문자열): 연결에 대한 설명
- type (문자열): 연결 유형 (예: "line", "dotted")
- color (문자열): 연결 선의 색상 (예: "red", "blue")

direction의 종류

- **true**: 화살표가 있는 모든 연결
- **false**: 방향성 없음

• **direction (논리 자료형(Boolean))**: 방향성 유무 (예: true, false)

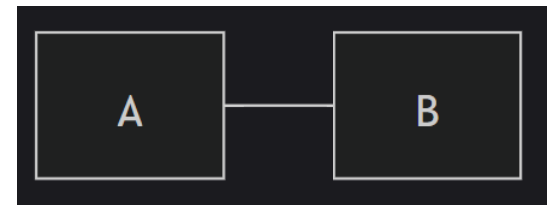
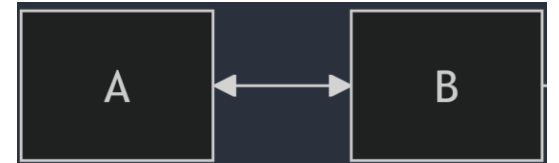
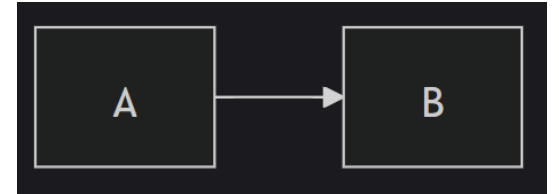
- thickness (문자열): 연결선의 두께 (예: "thin", "thick")

- X표시가 있는 연결선은 X를 텍스트(text)로 처리

예시: ✕

- 양방향 연결일 경우 2개의 node로 표기

예시: from A to B ("direction": true)
from B to A ("direction": true)



3. JSON 파일 생성 규칙

3.4 라벨링 정보 - 연결 - thickness

이미지로부터 라벨링된 데이터

3.4.1. connections (배열)

- 시스템 내 구성 요소들 간의 연결 정보
- from (문자열): 출발 구성 요소의 ID
- to (문자열): 연결될 목적 구성 요소의 ID
- text (문자열): 연결에 대한 설명
- type (문자열): 연결 유형 (예: "line", "dotted")
- color (문자열): 연결 선의 색상 (예: "red", "blue")
- direction (문자열): 연결의 방향 (예: "line", "arrow")

• **thickness (문자열): 연결선의 두께 (예: "thin", "thick")**

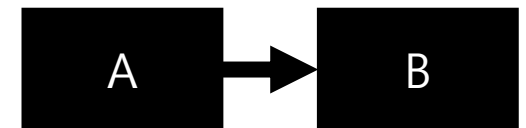
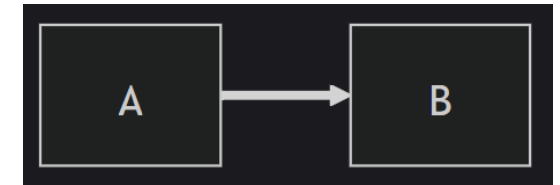
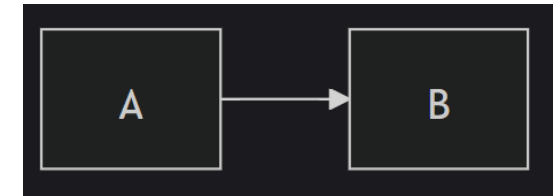
thickness의 종류

- thin: 얇은 선
- medium: 중간 두께의 선
- thick: 두꺼운 선

Default = medium

2가지일 경우 medium, thick을 사용

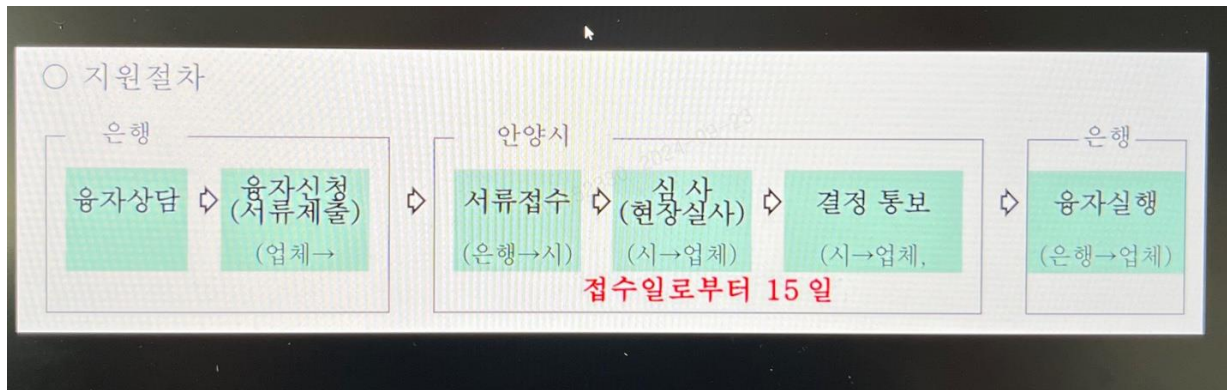
3가지일 경우 thin, medium, thick을 사용



4. 예시

4. 예시 JSON 파일

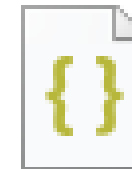
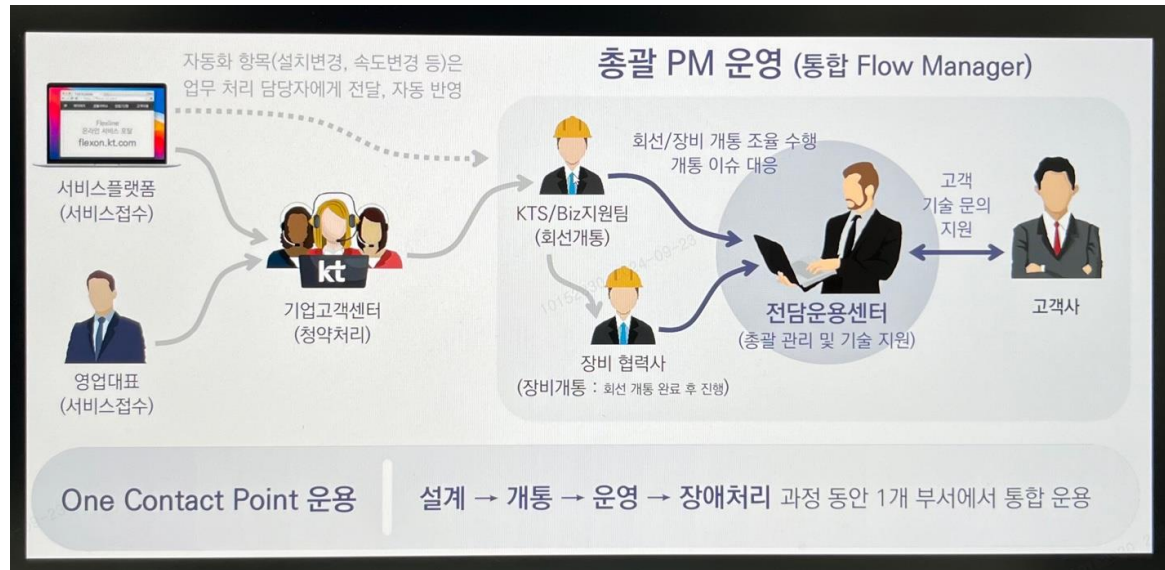
- 예시 1 (파일첨부: 10001.jpg)



00001.json

4. 예시 JSON 파일

- 예시 2 (파일첨부: 10002.jpg)

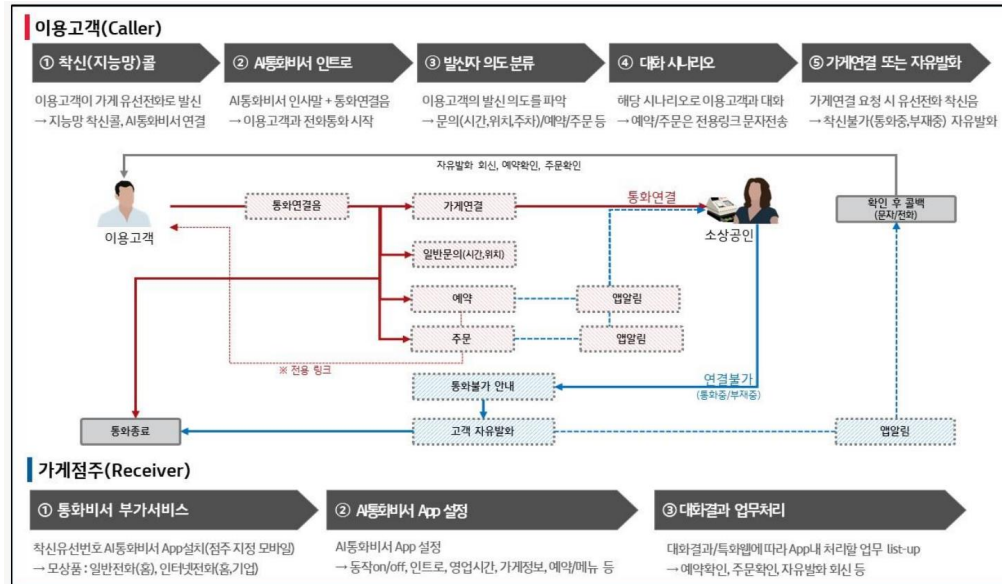


00002.json

열기: 더블클릭

4. 예시 JSON 파일

- 예시 3 (파일첨부: 10003.jpg)

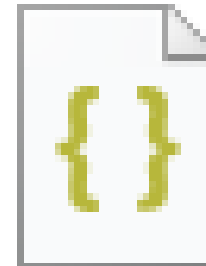


00003.json

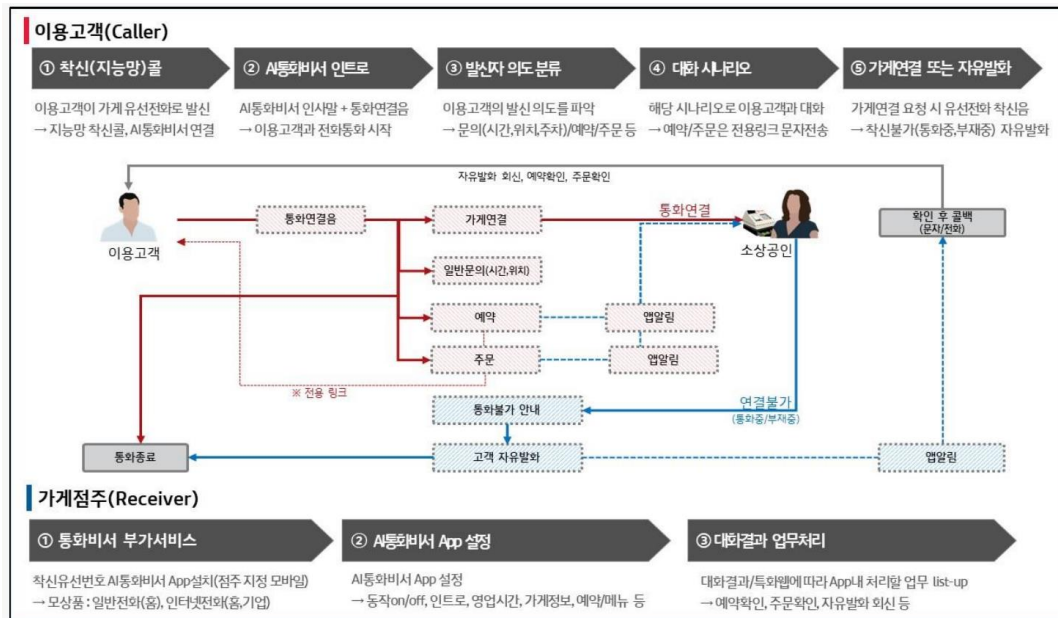
열기: 더블클릭

4. 예시 JSON 파일

- 예시 4 (파일첨부: 10004.jpg)

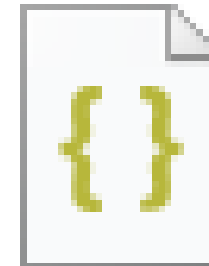
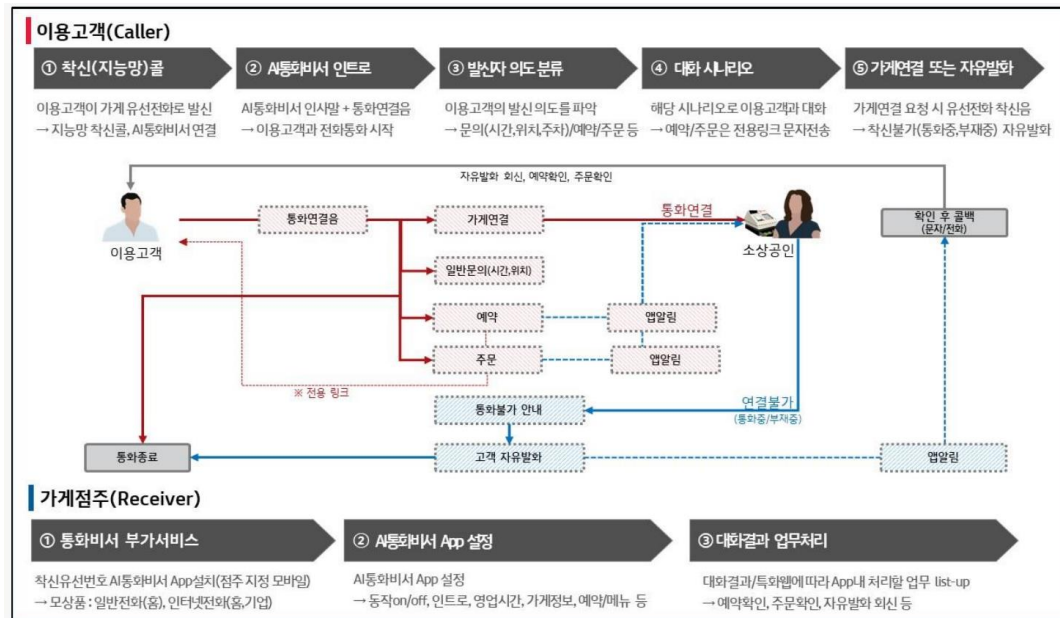


00003.json



4. 예시 JSON 파일

- 예시 5 (파일첨부: 10005.jpg)



00003.json

끝.

질문, 피드백 받습니다.