Macro builder

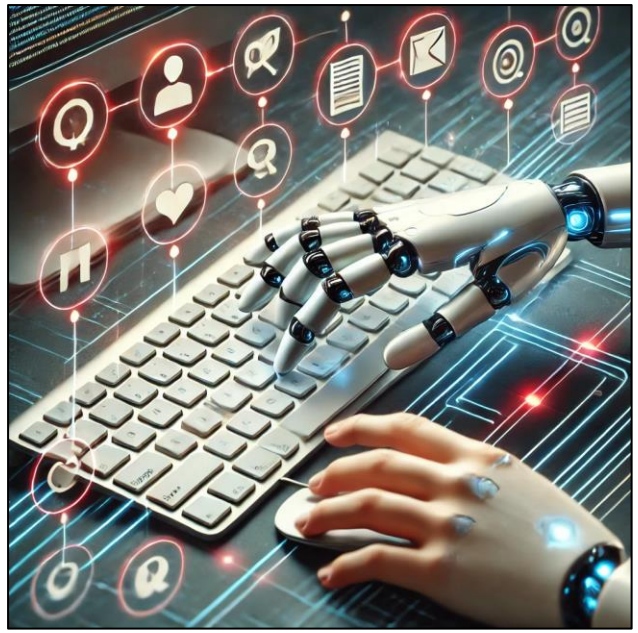
노코드 블록 코딩 기반 매크로 제작 플랫폼

팀명: 매크로빌더

디지털미디어학과 202021051 윤규민 (GUI, Backend)

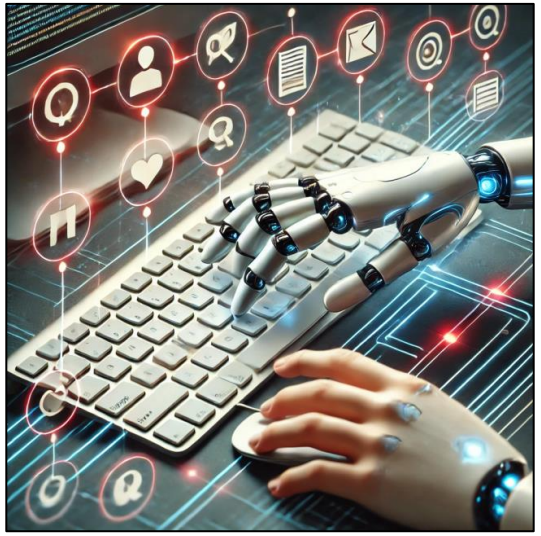
소프트웨어학과 202021511 조민재 (Frontend, Design)

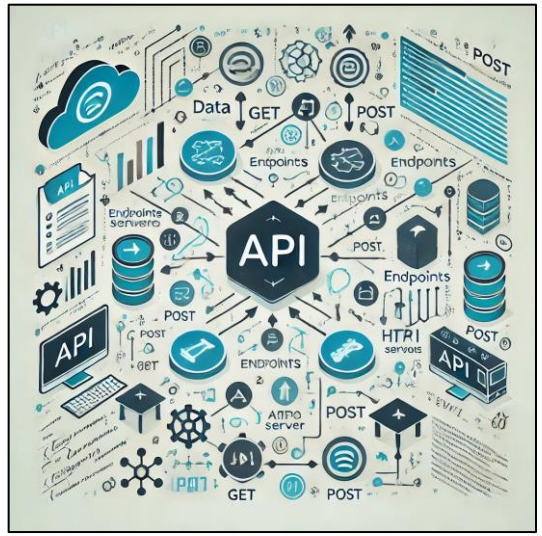
프로젝트 개요



매크로 빌더는 사용자가 자신의 컴퓨터로 수행하는 **반복적인 업무를 자동화**할 수 있도록 도와주는 웹 플랫폼으로, 제공되는 **블록 기반 인터페이스 노코드 툴**을 통해 사용자가 매크로를 직접 생성하고 다운받아 실행할 수 있게 함. **추후 마켓 플레이스와 플러그인**을 통해 추가적인 기능을 개발 및 공유할 수 있는 플랫폼 개발이 목적

제공 기능





파일 관리 블록

파일, 폴더를 복사, 삭제, 압축, 이름변경 등 여러 작업을 할 수 있는 블록을 제공

마우스 키보드 블록

마우스 키보드의 액션을 녹화한후 이를 재생할 수 있는 블록을 제공

- 녹화의 경우 **별도의 gui를 제공함**

외부 API 블록

외부 API 데이터를 손쉽게 다룰 수 있게 블록을 제공함

Ex) 한국 법률 Api

제공 기능

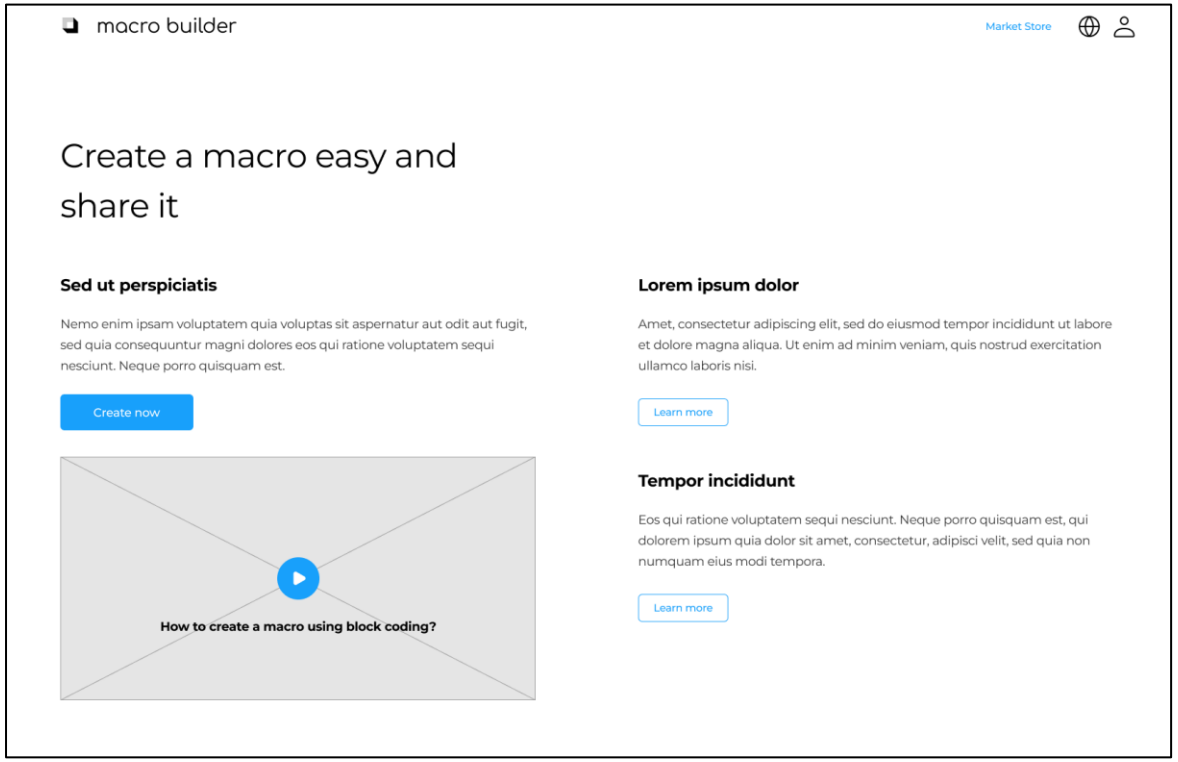
매크로 실행 GUI

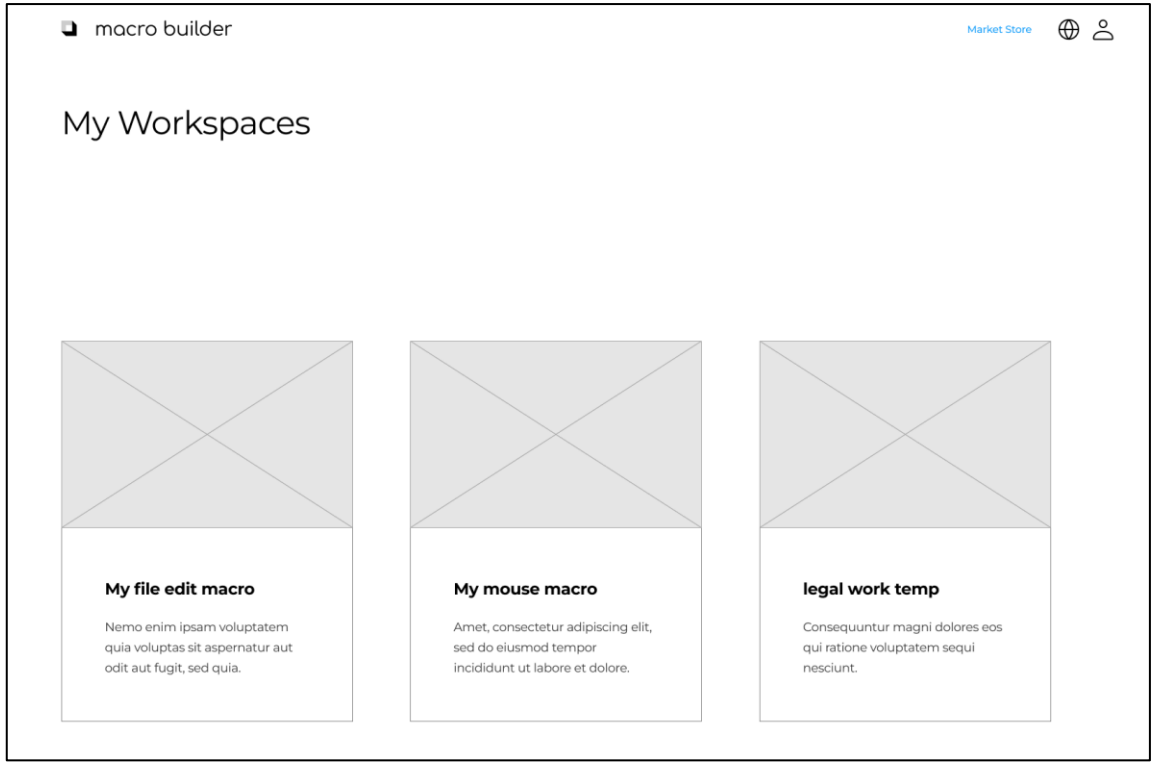
블록 코딩한 결과물을 **패키징 된 GUI 형태**로 다운받아서 로컬에서 실행 가능

녹화 프로그램 GUI

웹을 통해 마우스 키보드 블록을 사용할 때 자동으로 다운받아지며, **로컬**에서 발생하는 마우스. 키보드 액션을 **녹화해서 웹으로 보냄**

사용자 인터랙션 시나리오





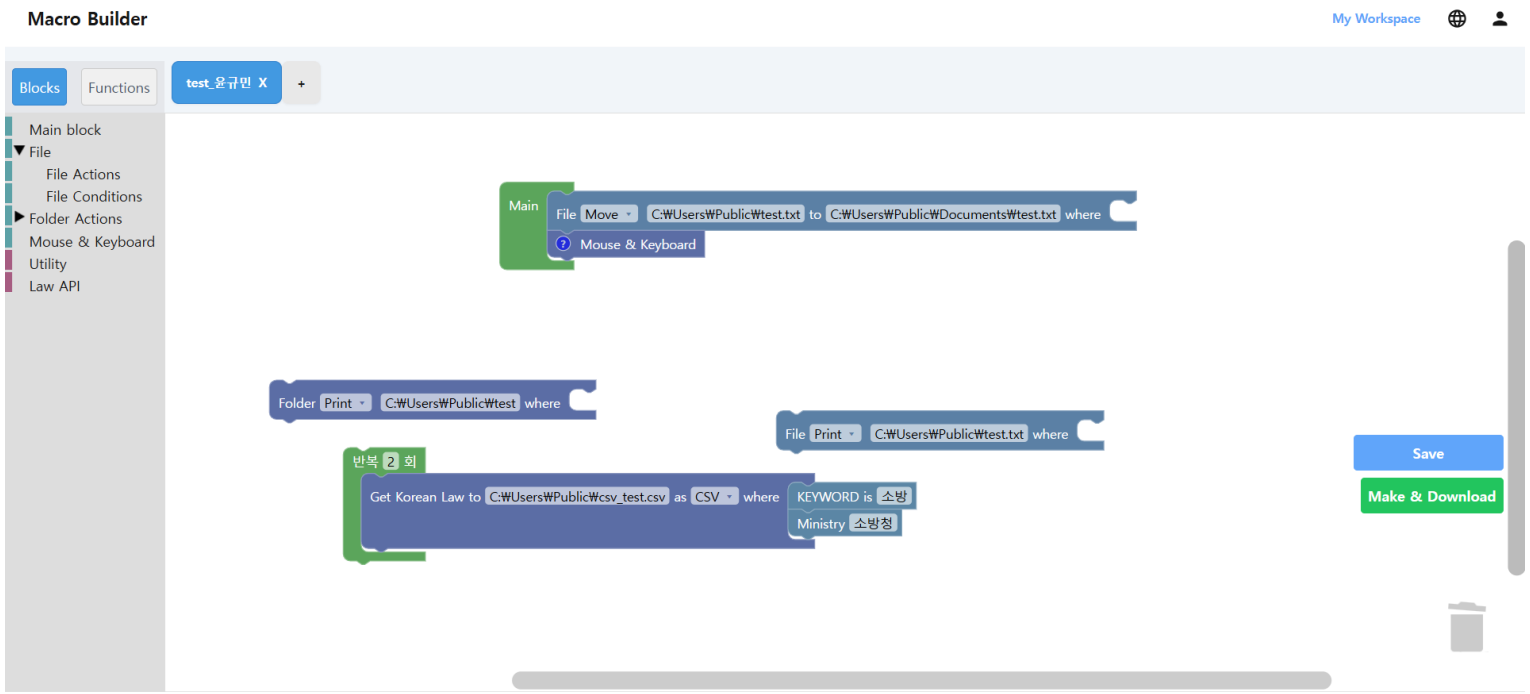
랜딩 페이지

랜딩 페이지를 통해 사이트에 처음으로 접속한 사용자에게 사용방법을 안내.

내 워크스페이스

유저가 블록코딩으로 매크로 로직을 설계할 수 있는 매크로 제작 공간을 불러오거나 새로 만들 수 있도록 함.

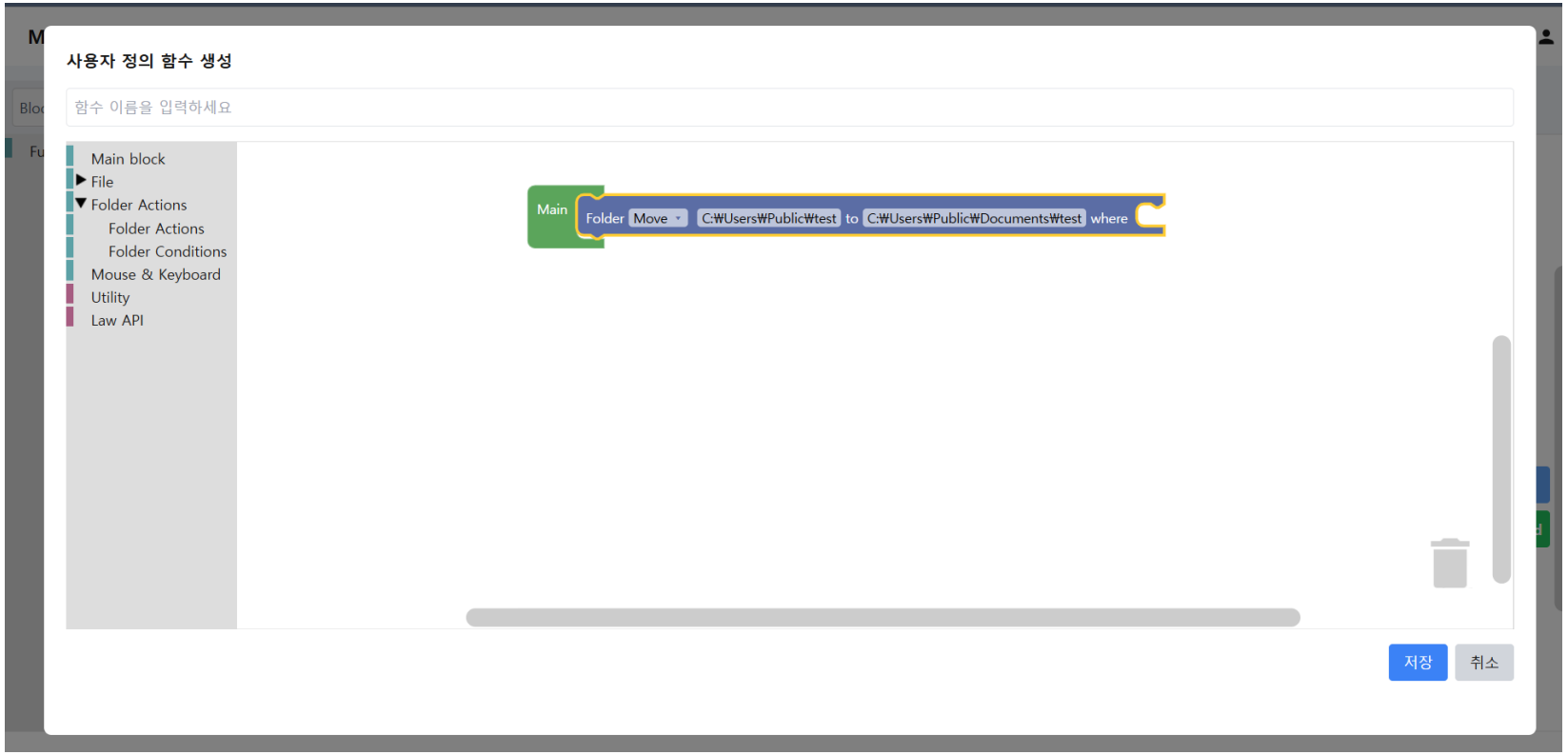
사용자 인터랙션 시나리오



워크시트 공간

- 유저가 **블록 코딩**을 이용해서 매크로 프로그램을 작성할 수 있도록 하는 공간.
- 파일 관련 기능, 마우스 & 키보드 조작, api 활용 등의 블록이 존재

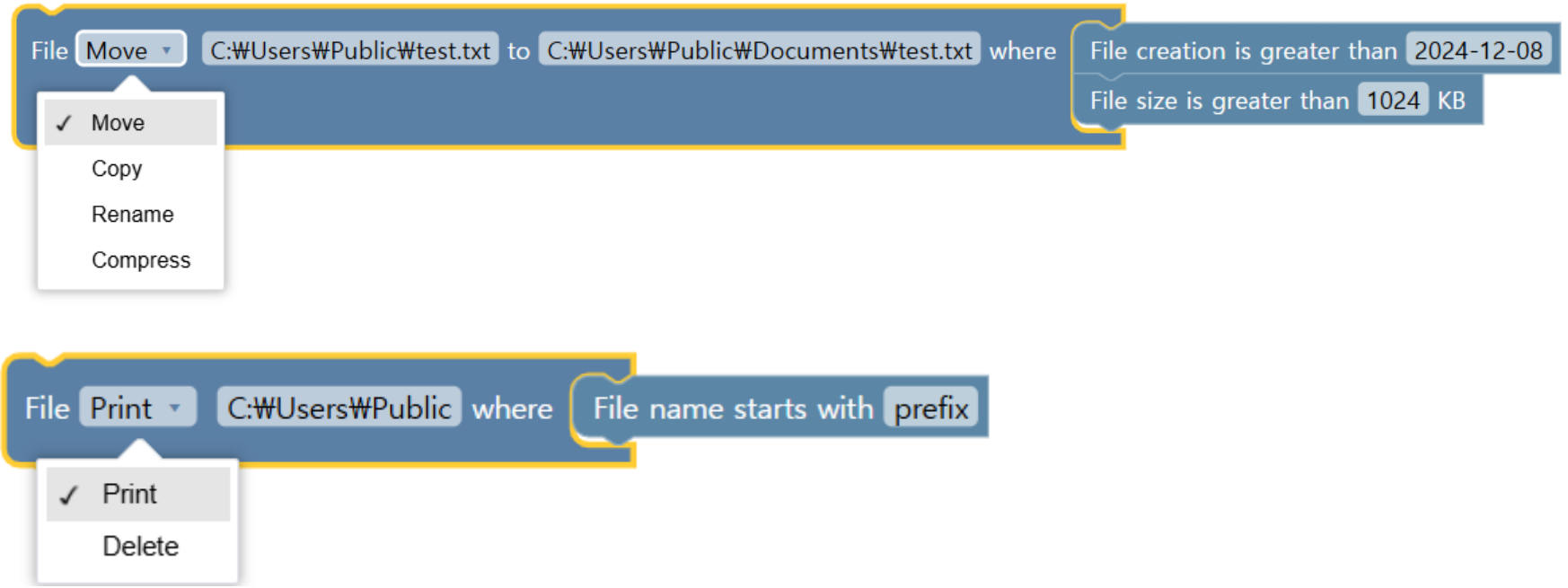
사용자 인터랙션 시나리오



함수 시트

- 블록을 별도의 함수로 만들 수 있음.
- 만든 함수를 워크시트에 블록으로 가져올 수 있음.
- 함수블록 내에 다른 함수 집어넣을 수 있음.

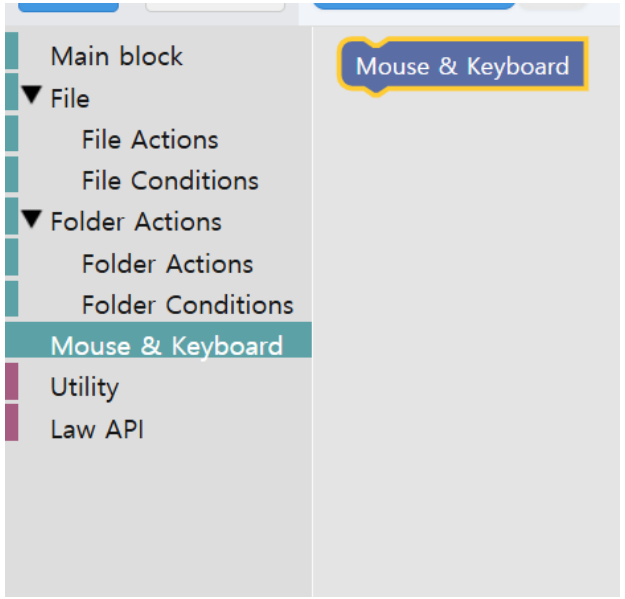
사용자 인터랙션 시나리오



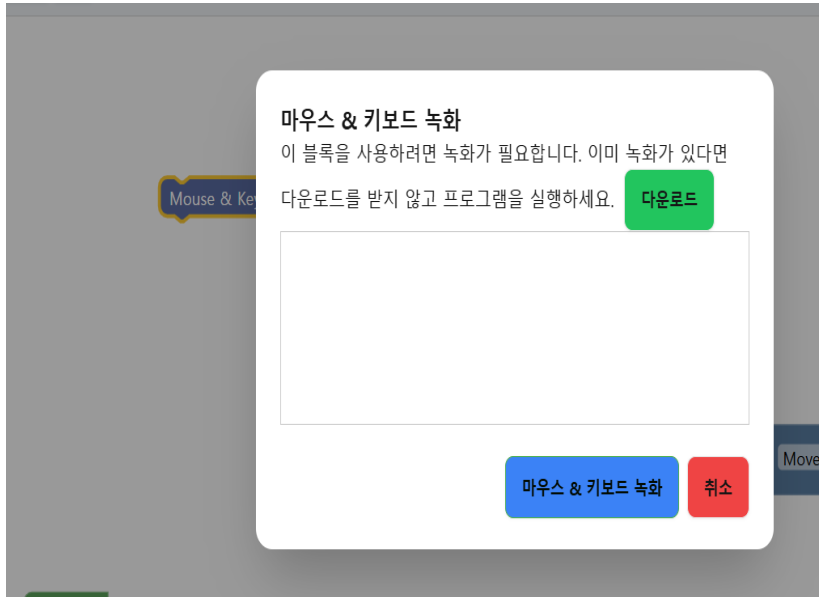
File system Block

- 특정 조건에 맞게 파일, 폴더를 어떻게 처리할 건지를 정의할 수 있음.
- 현재는 기본적인 기능만 제공

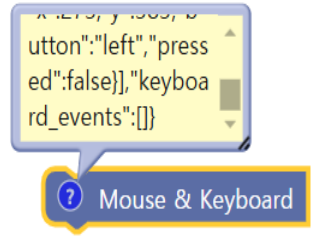
사용자 인터랙션 시나리오



드롭다운



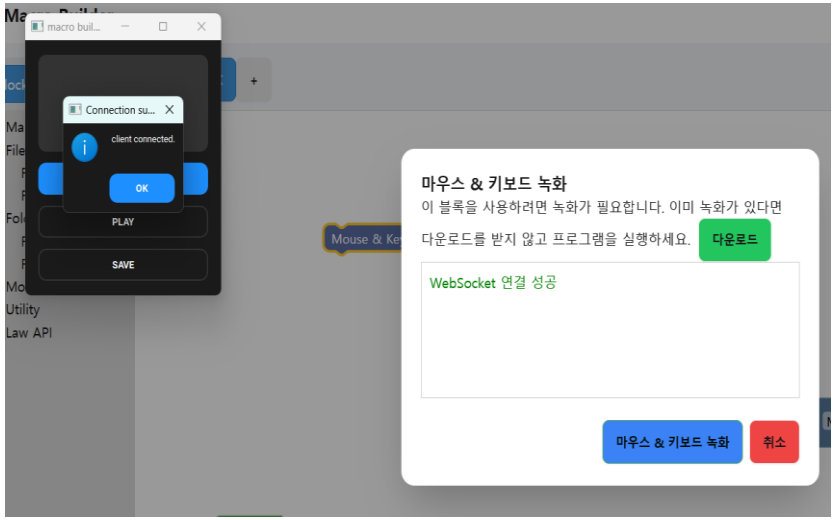
녹화
완료시



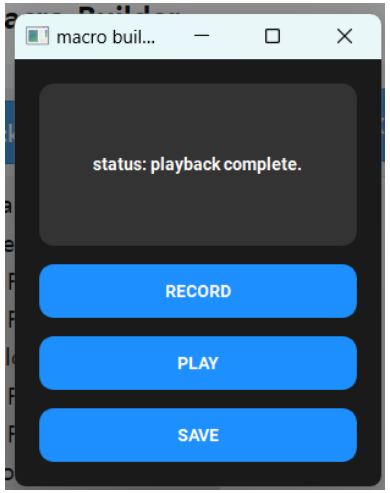
Mouse & Keyboard Block

- Worksheet로 드롭다운 시, GUI를 다운받거나 실행시킬 수 있음
- Gui에서 녹화를 완료하면 블록으로 만들어짐

사용자 인터랙션 시나리오



다운 받은 후 client와 연결



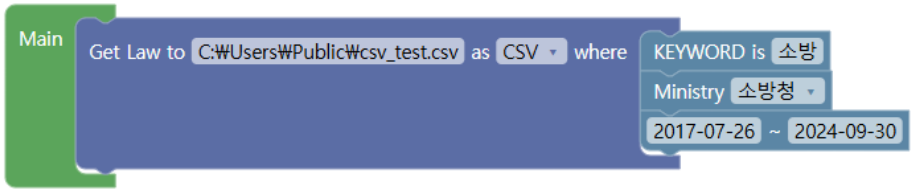
녹화를 끝낸 이후

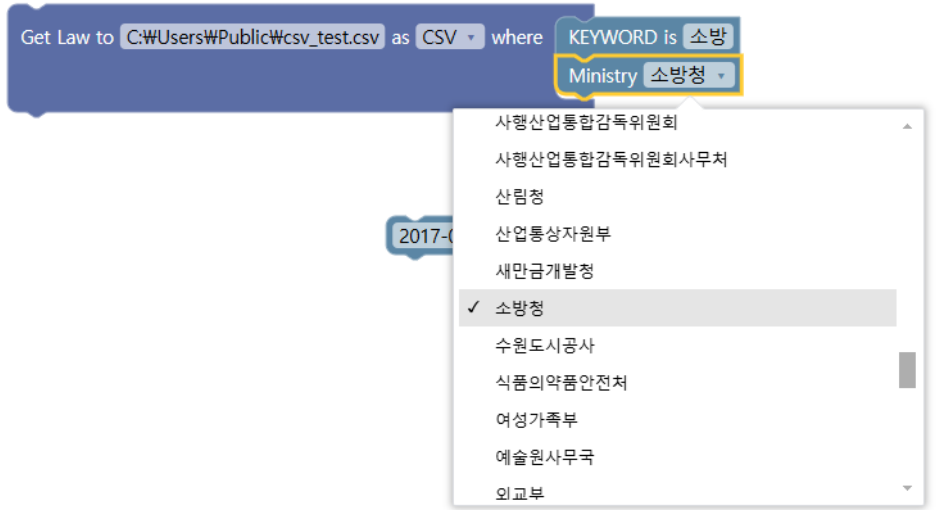
Save 버튼 클릭시 client 에게 넘김

Recorder GUI

- **Client(웹)**와 연결하지 않으면 단독 동작 안됨
- 녹화한걸 재생할 수 있음
- 녹화한걸 저장할 시 client 로 넘겨준 후 종료

사용자 인터랙션 시나리오



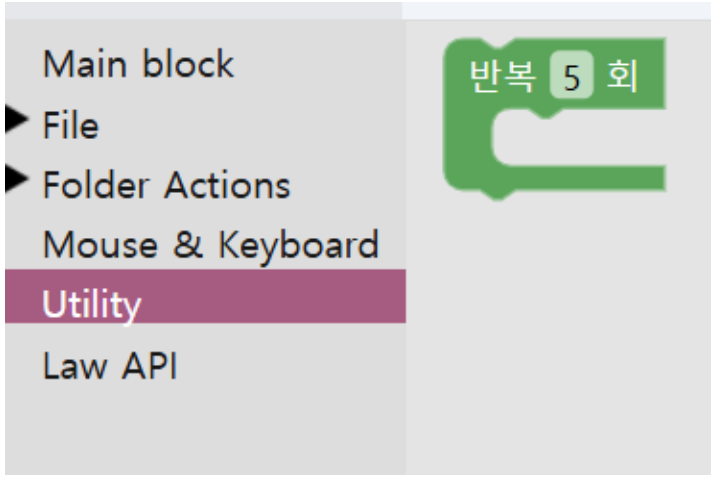


Korean Law Api Block

- **플러그인 블록의 일종** (특수한 목적의 블록)
- Conditions 에 맞는 법률 본문을 원하는 위치에 csv 파일(Excel)로 저장 할 수 있음

사용자 인터랙션 시나리오

Reference Block

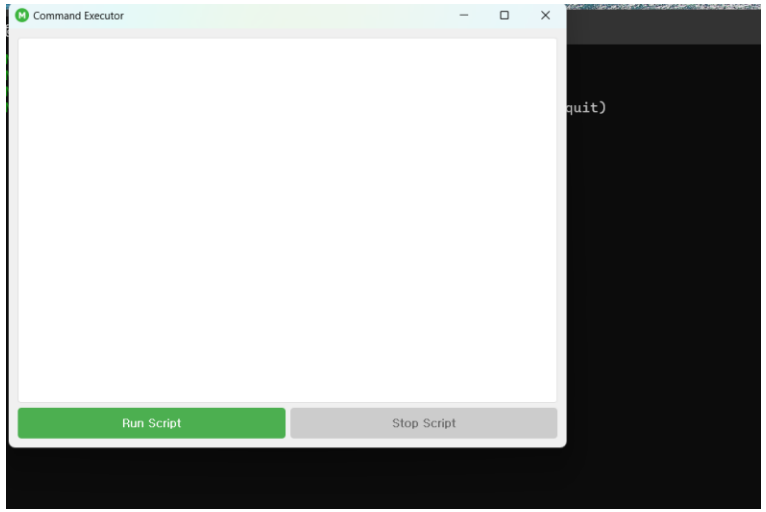


Main Loop Block

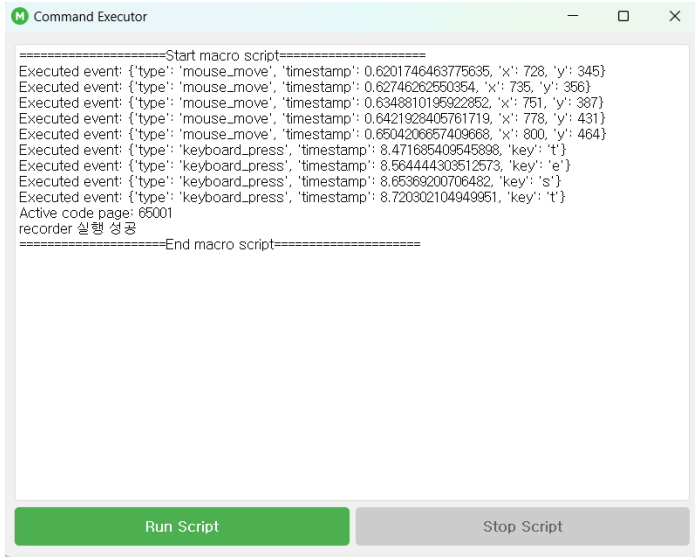
기타 Base block

- **Reference Block** 은 만든 함수를 블록처럼 사용할 수 있음. **함수내의 함수도 가능**.
- **Base Loop Block**은 코딩의 while 문 루프와 동일

사용자 인터랙션 시나리오



Script 실행전



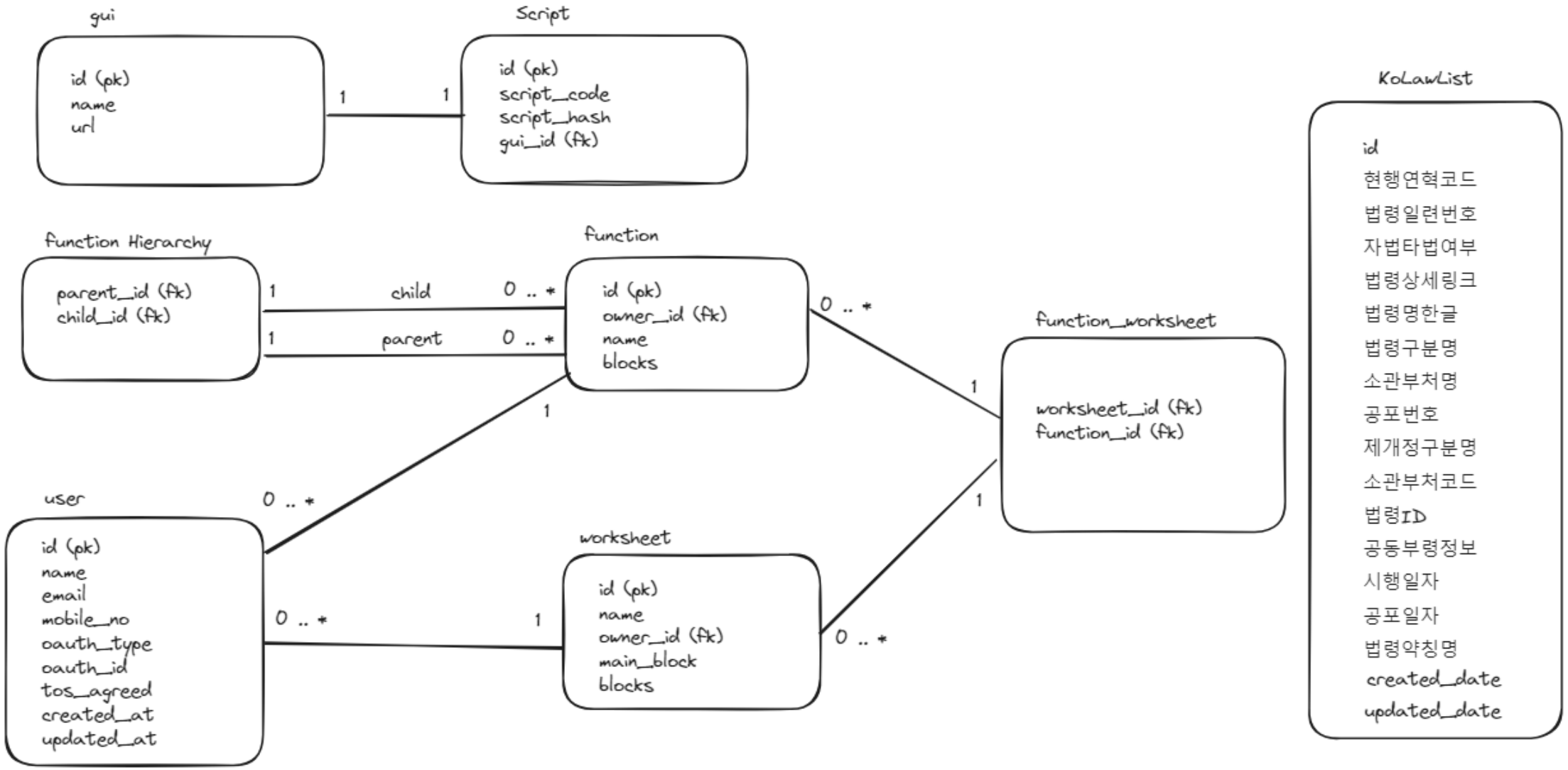
Script 실행완료

Command Gui

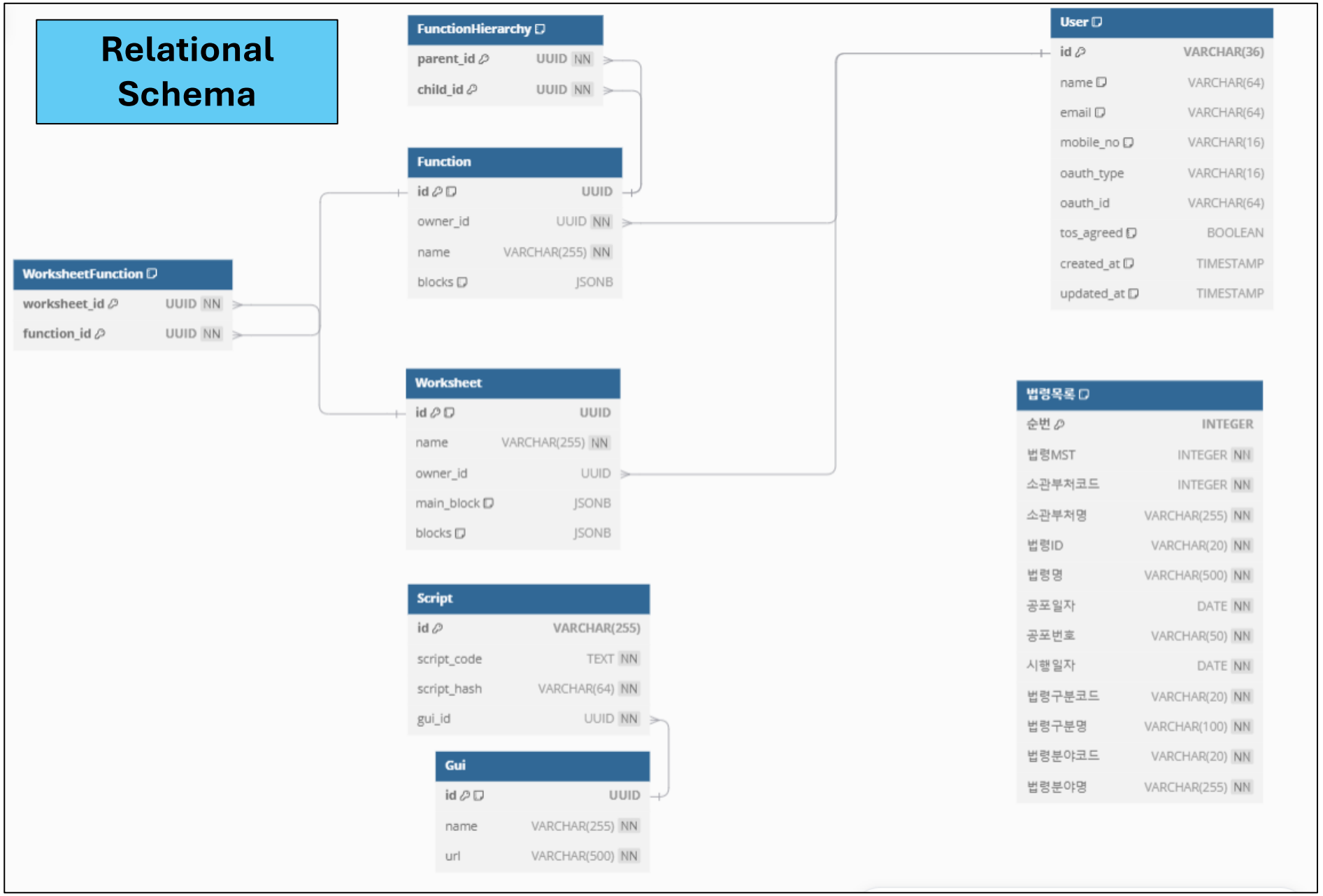
- 만든 worksheet의 매크로를 실행시킬 수 있음.

데이터베이스 스키마 - UML

UML



데이터베이스 스키마 - Relational Schema



데이터베이스 스키마 – table tuple

KoLawList

	id	현행연혁코드	법령일련번호	자법타법여부	법령상세링크
1	a56ca905-7788-415c-8a17-f26c2df03e91	현행	253527		https://www.law.go.kr/DF
2	2cf3cf26-75df-484c-abe9-fb6eea48c23b	현행	266351		https://www.law.go.kr/DF
3	fbe1a54a-a594-4649-bbb8-f78447bc61fa	현행	262693		https://www.law.go.kr/DF
4	c5b420f9-25d8-462c-8c6a-821d189b0035	현행	257787		https://www.law.go.kr/DF
5	8af6b692-9cc2-4b1b-bc2b-06064f01fb58	현행	263521		https://www.law.go.kr/DF
6	d73ee715-a95e-42b8-9ab3-8fa10a9aa390	현행	257789		https://www.law.go.kr/DF
7	e21ba16d-1701-4033-8691-4042c086d4ab	현행	263525		https://www.law.go.kr/DF
8	23ca888a-cdf7-455a-8093-79ba36a0b332	현행	265713		https://www.law.go.kr/DF
9	118568dd-04a9-4a6e-a1ac-2b3addf46308	현행	257743		https://www.law.go.kr/DF
10	234d3c27-8c8f-40bb-aca9-4ec1f9d0e7bd	현행	263523		https://www.law.go.kr/DF
11	81e9a831-ffa7-48e5-b4af-773870755505	현행	263571		https://www.law.go.kr/DF

Worksheet

	id	name	blocks	owner_id	main_block	raw_blocks
1	6b989ab4-5e57-4bfd-834b-d9703512dc1b	test_윤규민	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"loc": "C:\\Users\\Public\\	[{"blocks": {"block
2	607cec7d-e032-4d65-82cc-eb922be45ac0	asddwqe	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]
3	ddbec7a6-a301-49bc-aa82-9f06e9e3dcc2	dd	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]
4	f8975139-1f7b-4d11-b468-4a6b182fbbd5	test_윤규민1	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"loc": "C:\\Users\\Public\\csv_test.c	[{"blocks": {"block
5	907d93c3-1a15-4290-8c38-3f29466c8901	worksheet_for_auto	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]
6	ae8eef66-b956-48da-88bc-a457f7d0bcdd	시연1	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [], "block_type": "MAIN_BLOCK"}	[{"blocks": {"block
7	fa24d55b-f76a-4c9a-b68e-55033eae84c6	gmyun1999_ws	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]
8	e8557422-26a6-497f-aa40-74887c087137	gmyun1999_ws1	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]
9	04ed5648-9f92-4872-9543-b925fb0786fb	gmyun1999_ws2	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]
10	7a9de624-d929-4440-96b6-2cf28ff29758	my_worksheet	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	{"body": [{"body": {}, "block_type": "FILE_SYS	[]

Function

	id	name	blocks	owner_id	raw_blo
1	f80a9ef1-765d-4f0a-8d48-105af707840a	내가만든_사용자_정의함수_1	[]	7673a900-4f76-4e8b-b740-18d8146e9a3b	[{"blocks":
2	2be39e19-0b1d-42ce-ba18-cabc2e82cb29	테스트1	[{"body": [{"loc": "C:\\Users\\Public", "action": "MOVE", "rei	7673a900-4f76-4e8b-b740-18d8146e9a3b	[{"blocks":
3	073ccbb4-2a51-4173-bd85-74356a090c85	test2	[{"body": [{"loc": "C:\\Users\\Public", "action": "MOVE", "rei	7673a900-4f76-4e8b-b740-18d8146e9a3b	[{"blocks":
4	df575dde-6485-4533-9b3e-f122a2240c38	test_function3123	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]
5	db902108-ea84-4092-9b97-63eed0937bf0	myfunction	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]
6	ea2a707d-9521-4eb2-8f65-aed1f070d50b	내함수	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]
7	d96af809-df5a-4ce1-ad88-e6a89e1a190a	asdzxcq	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]
8	0a0a5615-a5f4-42db-b27a-47e0028b2261	function123	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]
9	7164b998-d4dd-4edf-a37b-313672e46075	마우스키보드자동화함수	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]
10	9a30e055-e7ad-447b-9c21-fc770e28b196	소방관련법률자동다운함수	[{"body": [{"block_type": "REFERENCE_BLOCK", "reference_	7673a900-4f76-4e8b-b740-18d8146e9a3b	[]

Script

	id	script_code	script_hash	gui_id
1	b951347a-d74b-4897-b9c4-3ff493137e3f	import subprocess¶import requests¶¶def send_recorder_c	d8d75644137d0be7e570fa51122ca56da4a34b68b561121	f6be9ca7-7aad-46a5-a4d
2	4489e6fe-41fa-4586-acc5-42d19f6e0c4e	import subprocess¶import requests¶¶def send_recorder_c	404a114d229ccd7cae641d28ee3a938dcc6eaeda7ace2ff6	e0679e6a-b891-4366-a3e
3	3af9d39c-5a62-4b72-8ef9-7acd8a2db52f	import subprocess¶import requests¶¶def send_recorder_c	3939830c26cff0aabf067cc5c3dfaa790822919803b02d783	0a89ac56-b7fb-4008-8f7
4	cbfdd5a4-33c7-4b90-85e8-134e43e9b0ca	import subprocess¶import requests¶¶def send_recorder_c	0f6a91a77085a889620088d6f96c26e53b5184d600c19412	738a80f4-16c1-4b24-b64
5	041c3109-9d03-4d6f-8ceb-03be3288dea2	import subprocess¶import requests¶¶¶subprocess.run(['p	0d2e9fd7e878feb28d0c38d5979561da2247009d53cc704!	949ddc8b-1c34-4318-aa9
6	0a1ca7c5-d2dc-43f0-990c-4208e6fb08b7	import subprocess¶import requests¶¶def send_recorder_c	1e472ce989bee4256a115e6b37c4e2e2ef2ccb56d3710d8:	86b554cb-4ca4-464b-b47
7	0211ec97-6bde-424e-889b-5bf1c5136496	import subprocess¶import requests¶¶def send_recorder_c	2854a848dd67f2eaacdf84d4dd583360662e8f19e23740d2	eaf5cb4b-8a67-4705-989
8	9f12cc9d-c16f-4c9d-8f69-df8bd3acbc38	import subprocess¶import gzip¶import io¶import reques	9312d4b637d230d5d00ed6efee6f3de58ccce995315e83f4	ebde3801-2f80-49df-a08
9	8e6f6ab4-b6b4-4be7-af00-818aff439a65	import subprocess¶import requests¶¶¶subprocess.run(['p	83be09933a95f432c33af825bb41b6087e6e7dbfa02683a5	4ecb254d-06ec-4a78-bb7
10	cb779c9c-a872-4927-83d1-6cf698aaff2b	import subprocess¶import requests¶¶¶subprocess.run(['p	564975b008db230eea1e4374f9ac03ccca563bb88b68528	1d6cd723-b766-4fee-9ab
11	2c30ea2e-3a48-4849-a556-24e00540ca15	import subprocess¶import requests¶¶def send_recorder_c	11f03f7325c10bab45bc8a6c251acb8e80c83aadbc2ad608	fd77d7b4-a951-4262-a61

KoOrg

	id	org	org_name	created_date	updated_date
1	1075000	1075000	5.18민주화운동진상규명조사위원회	2024-12-08 04:35:51.091 +0900	2024-12-08 04:35:51.091 +0900
2	1074601	1074601	가습기살균제사건과4.16세월호참사특별조사위원회	2024-12-08 04:35:51.095 +0900	2024-12-08 04:35:51.095 +0900
3	1040000	1040000	감사원	2024-12-08 04:35:51.099 +0900	2024-12-08 04:35:51.099 +0900
4	1790365	1790365	개인정보보호위원회	2024-12-08 04:35:51.103 +0900	2024-12-08 04:35:51.103 +0900
5	B552557	B552557	게임물등급위원회	2024-12-08 04:35:51.106 +0900	2024-12-08 04:35:51.106 +0900
6	1320000	1320000	경찰청	2024-12-08 04:35:51.109 +0900	2024-12-08 04:35:51.109 +0900
7	1492000	1492000	고용노동부	2024-12-08 04:35:51.114 +0900	2024-12-08 04:35:51.114 +0900
8	1790364	1790364	고위공직자범죄수사처	2024-12-08 04:35:51.118 +0900	2024-12-08 04:35:51.119 +0900
9	1160156	1160156	공적자금관리위원회사무국	2024-12-08 04:35:51.122 +0900	2024-12-08 04:35:51.122 +0900
10	1130000	1130000	공정거래위원회	2024-12-08 04:35:51.125 +0900	2024-12-08 04:35:51.125 +0900

Gui

	id	name	url
1	f6be9ca7-7aad-46a5-a4d1-e49eb7520826	9b218f8f-4420-4663-9fcf-3bacb96f4221	https://macro-command-gui.s3.amazonaws.com/packaged_files/3a85b624-6418-4e13-810
2	e0679e6a-b891-4366-a3e5-7ec05954a59c	b6e39c2f-da84-43ec-ad74-d82fa2fef6c6	https://macro-command-gui.s3.amazonaws.com/packaged_files/d74959c4-d4d1-4700-8c7
3	0a89ac56-b7fb-4008-8f73-60add91477f9	ca1a8adf-9729-4b1c-b760-b3aa33f951fd	https://macro-command-gui.s3.amazonaws.com/packaged_files/045031e9-8a2c-4082-a47
4	738a80f4-16c1-4b24-b64b-e7dcc6e4791a	eb9db591-ad94-4eec-a678-3ccb77719327	https://macro-command-gui.s3.amazonaws.com/packaged_files/70fefa06-a06e-4106-9e81
5	949ddc8b-1c34-4318-aa9a-6ad11664d098	887056ee-9844-4d1c-b8c7-f25669f43fce	https://macro-command-gui.s3.amazonaws.com/packaged_files/38588862-b385-499a-882
6	86b554cb-4ca4-464b-b472-e776fa13b32e	d886e735-4116-47ca-b2a0-9c3117a7f821	https://macro-command-gui.s3.amazonaws.com/packaged_files/e60453be-16de-4dcc-8b7
7	eaf5cb4b-8a67-4705-9895-889b0e134d15	194fcf1a-7870-411a-8ea9-939939997596	https://macro-command-gui.s3.amazonaws.com/packaged_files/40a6ff2d-1099-46df-a6fd-
8	4ecb254d-06ec-4a78-bb7f-f28c7e5d251e	b93834a5-701b-45a1-aa70-0380c1c562db	https://macro-command-gui.s3.amazonaws.com/packaged_files/e27b9c3b-30c3-45ad-880
9	1d6cd723-b766-4fee-9ab5-cd0a14effcac	af6ee012-1b1a-497c-8e38-f680a52cd3e5	https://macro-command-gui.s3.amazonaws.com/packaged_files/b4e39a09-95fa-4280-99d7
10	fd77d7b4-a951-4262-a618-ee2715f3f7d8	2f6d8c40-1cd2-42d2-9bbe-e5d05224e147	https://macro-command-gui.s3.amazonaws.com/packaged_files/65f76617-5ba3-4cdc-8fac-

SQL문 – 스키마 생성

```
-- UUID 생성을 위한 uuid-ossp 확장 활성화
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

-- 1. Function 모델
CREATE TABLE "Function" (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  owner_id UUID NOT NULL REFERENCES "user_user" (id) ON DELETE CASCADE,
  name VARCHAR(255) NOT NULL,
  blocks JSONB DEFAULT '[]'::jsonb
);

-- 2. Worksheet 모델
CREATE TABLE "Worksheet" (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  name VARCHAR(255) NOT NULL,
  owner_id UUID NULL REFERENCES "user_user" (id) ON DELETE CASCADE,
  main_block JSONB DEFAULT NULL,
  blocks JSONB DEFAULT '[]'::jsonb
);

-- 3. WorksheetFunction 모델
CREATE TABLE "WorksheetFunction" (
  worksheet_id UUID NOT NULL REFERENCES "Worksheet" (id) ON DELETE CASCADE,
  function_id UUID NOT NULL REFERENCES "Function" (id) ON DELETE CASCADE,
  PRIMARY KEY (worksheet_id, function_id)
);

-- WorksheetFunction에 대한 인덱스 생성
CREATE INDEX idx_worksheet ON "WorksheetFunction" (worksheet_id);
CREATE INDEX idx_function ON "WorksheetFunction" (function_id);

-- 4. FunctionHierarchy 모델
CREATE TABLE "FunctionHierarchy" (
  parent_id UUID NOT NULL REFERENCES "Function" (id) ON DELETE CASCADE,
  child_id UUID NOT NULL REFERENCES "Function" (id) ON DELETE CASCADE,
  PRIMARY KEY (parent_id, child_id)
);

-- FunctionHierarchy에 대한 인덱스 생성
CREATE INDEX idx_parent ON "FunctionHierarchy" (parent_id);
CREATE INDEX idx_child ON "FunctionHierarchy" (child_id);

-- 5. Gui 모델
CREATE TABLE "Gui" (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  name VARCHAR(255) NOT NULL,
  url VARCHAR(500) NOT NULL
);

-- 6. Script 모델
CREATE TABLE "Script" (
  id VARCHAR(255) PRIMARY KEY,
  script_code TEXT NOT NULL,
  script_hash VARCHAR(64) UNIQUE NOT NULL,
  gui_id UUID NOT NULL REFERENCES "Gui" (id) ON DELETE CASCADE
);

-- 8. user모델
CREATE TABLE "User" (
  id VARCHAR(36) PRIMARY KEY,
  name VARCHAR(64) DEFAULT NULL,
  email VARCHAR(64) DEFAULT NULL,
  mobile_no VARCHAR(16) DEFAULT NULL,
  oauth_type VARCHAR(16) NOT NULL,
  oauth_id VARCHAR(64) NOT NULL,
  tos_agreed BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  -- Enforce unique constraint on oauth_type and oauth_id
  UNIQUE (oauth_type, oauth_id)
);
-- user 모델에 대한 인덱스 생성
CREATE INDEX idx_user_name ON "User" (name);
CREATE INDEX idx_user_created_at ON "User" (created_at);
CREATE INDEX idx_user_updated_at ON "User" (updated_at);

-- . 법령목록모델에 대한 인덱스
CREATE INDEX idx_law_id ON "KoLawList" (법령ID);
CREATE INDEX idx_competent_ministry_code ON "KoLawList" (소관부처코드);
CREATE INDEX idx_promulgation_date ON "KoLawList" (공포일자);
CREATE INDEX idx_enforcement_date ON "KoLawList" (시행일자);
CREATE INDEX idx_law_name ON "KoLawList" (법령명한글);

-- 9. 법령목록모델
CREATE TABLE "KoLawList" (
  id VARCHAR(255) PRIMARY
KEY,
  현행연혁코드 VARCHAR(100)
NOT NULL,
  법령일련번호 VARCHAR(100)
NOT NULL,
  자법타법여부 VARCHAR(100),
  법령상세링크 VARCHAR(500),
  법령명한글 VARCHAR(255),
  법령구분명 VARCHAR(100),
  소관부처명 VARCHAR(255),
  공포번호 VARCHAR(100),
  제개정구분명 VARCHAR(255),
  소관부처코드 VARCHAR(100),
  법령ID VARCHAR(100) NOT
NULL,
  공동부령정보 JSON,
  시행일자 VARCHAR(10),
  공포일자 VARCHAR(10),
  법령약칭명 VARCHAR(255),
  created_date TIMESTAMP
DEFAULT CURRENT_TIMESTAMP,
  updated_date TIMESTAMP
DEFAULT CURRENT_TIMESTAMP
ON UPDATE
CURRENT_TIMESTAMP
);
```

SQL문 – 서비스 제공

```
-- ----------------------------
-- 레포지토리: BlockFunctionRepo (IBlockFunctionRepo)
-- ----------------------------
-- 설명: owner_id 및/또는 id로 필터링된 Function 레코드를 조회합니다.
"""
SELECT id::text, owner_id::text, name, blocks
FROM "Function"
WHERE ($1 IS NULL OR owner_id = $1)
  AND ($2 IS NULL OR id = $2);
"""

-- 설명: 새로운 Function 레코드를 생성합니다.
"""
BEGIN;

INSERT INTO "Function" (owner_id, name, blocks)
VALUES ($1, $2, $3)
RETURNING id::text, owner_id::text, name, blocks;

COMMIT;
"""

-- 설명: 기존 Function 레코드를 업데이트합니다.
"""
BEGIN;

UPDATE "Function" SET name = $1,
   blocks = $2 WHERE id = $3 RETURNING id::text, owner_id::text, name, blocks;

COMMIT;
"""

-- 설명: 특정 Function 레코드를 삭제합니다.
"""
BEGIN;

DELETE FROM "Function" WHERE id = $1;

COMMIT;
"""
```

```
-- ----------------------------
-- 레포지토리: FunctionClosureRepo (IFunctionClosureRepo)
-- ----------------------------

-- 설명: FunctionHierarchy에 새로운 관계를 저장합니다.
"""
BEGIN;

INSERT INTO "FunctionHierarchy" (parent_id, child_id) VALUES ($1, $2);

COMMIT;
"""

-- 설명: 특정 FunctionHierarchy 관계를 삭제합니다.
"""
DELETE FROM "FunctionHierarchy" WHERE parent_id = $1 AND child_id = $2;
"""

-- 설명: 필터를 기반으로 FunctionHierarchy 관계를 조회합니다.
"""
SELECT parent_id, child_id FROM "FunctionHierarchy" WHERE ($1 IS NULL OR parent_id = $1)
  AND ($2 IS NULL OR child_id = $2);
"""

-- 설명: 여러 FunctionHierarchy 관계를 한 번에 삽입합니다.
"""
BEGIN;

INSERT INTO "FunctionHierarchy" (parent_id, child_id) VALUES
   ($1, $2),
   ($3, $4)
   -- 필요한 추가 관계를 여기에 추가하세요
;

COMMIT;
"""

... 다음 슬라이드에 계속
```

SQL문 – 서비스 제공

```
-- ----------------------------
... continue 레포지토리: FunctionClosureRepo (IFunctionClosureRepo)
-- ----------------------------
-- 설명: 여러 FunctionHierarchy 관계를 한 번에 삭제합니다.
"""
BEGIN;

DELETE FROM "FunctionHierarchy" WHERE (parent_id = $1 AND child_id = $2)
  OR (parent_id = $3 AND child_id = $4)
  -- 필요한 추가 조건을 여기에 추가하세요
;

COMMIT;
"""

-- 설명: 주어진 root_function_id에 대한 모든 상위 Function ID를 조회합니다.
"""
WITH RECURSIVE ancestors AS (
   SELECT parent_id
   FROM "FunctionHierarchy"
   WHERE child_id = $1

   UNION

   SELECT fh.parent_id
   FROM "FunctionHierarchy" fh
   INNER JOIN ancestors a ON fh.child_id = a.parent_id
)
SELECT DISTINCT parent_id FROM ancestors;
"""
```

```
-- ----------------------------
-- 레포지토리: GuiRepo (IGuiRepo)
-- ----------------------------

-- 설명: 새로운 Gui 레코드를 생성합니다.
"""
BEGIN;

INSERT INTO "Gui" (id, name, url) VALUES ($1, $2, $3) RETURNING id::text, name, url;

COMMIT;
"""

-- 설명: 필터를 기반으로 Gui 레코드를 조회합니다.
"""
SELECT id::text, name, url FROM "Gui" WHERE ($1 IS NULL OR id = $1);
"""

-- 설명: 기존 Gui 레코드를 업데이트합니다.
"""
BEGIN;

UPDATE "Gui" SET name = $1,
   url = $2 WHERE id = $3 RETURNING id::text, name, url;

COMMIT;
"""

-- 설명: 특정 Gui 레코드를 삭제합니다.
"""
BEGIN;

DELETE FROM "Gui" WHERE id = $1;

COMMIT;
"""
```

SQL문 – 서비스 제공

```
-- ----------------------------
-- 레포지토리: ScriptRepo (IScriptRepo)
-- ----------------------------

-- 설명: 필터를 기반으로 단일 Script 레코드를 조회합니다.
"""
SELECT id, script_code, script_hash, gui_id FROM "Script" WHERE ($1 IS NULL OR id = $1)
  AND ($2 IS NULL OR script_hash = $2)
  AND ($3 IS NULL OR gui_id = $3) LIMIT 1;
"""

-- 설명: 새로운 Script 레코드를 생성합니다.
"""
BEGIN;

INSERT INTO "Script" (id, script_code, script_hash, gui_id) VALUES ($1, $2, $3, $4) RETURNING id,
script_code, script_hash, gui_id;

COMMIT;
"""

-- 설명: 특정 Script 레코드를 삭제합니다.
"""
BEGIN;

DELETE FROM "Script" WHERE id = $1;

COMMIT;
"""

-- 설명: script_hash를 기준으로 Script를 조회하거나 존재하지 않으면 생성합니다.
"""
WITH ins AS (
    INSERT INTO "Script" (id, script_code, script_hash, gui_id)
    VALUES ($1, $2, $3, $4)
    ON CONFLICT (script_hash) DO NOTHING
    RETURNING id, script_code, script_hash, gui_id
)
SELECT id, script_code, script_hash, gui_id FROM ins UNION SELECT id, script_code, script_hash, gui_id
FROM "Script" WHERE script_hash = $3 LIMIT 1;
"""

-- 설명: 주어진 script_hash를 가진 Script가 존재하는지 확인합니다.
"""
SELECT EXISTS (
    SELECT 1
    FROM "Script"
    WHERE script_hash = $1
);
"""
```

SQL문 – 서비스 제공

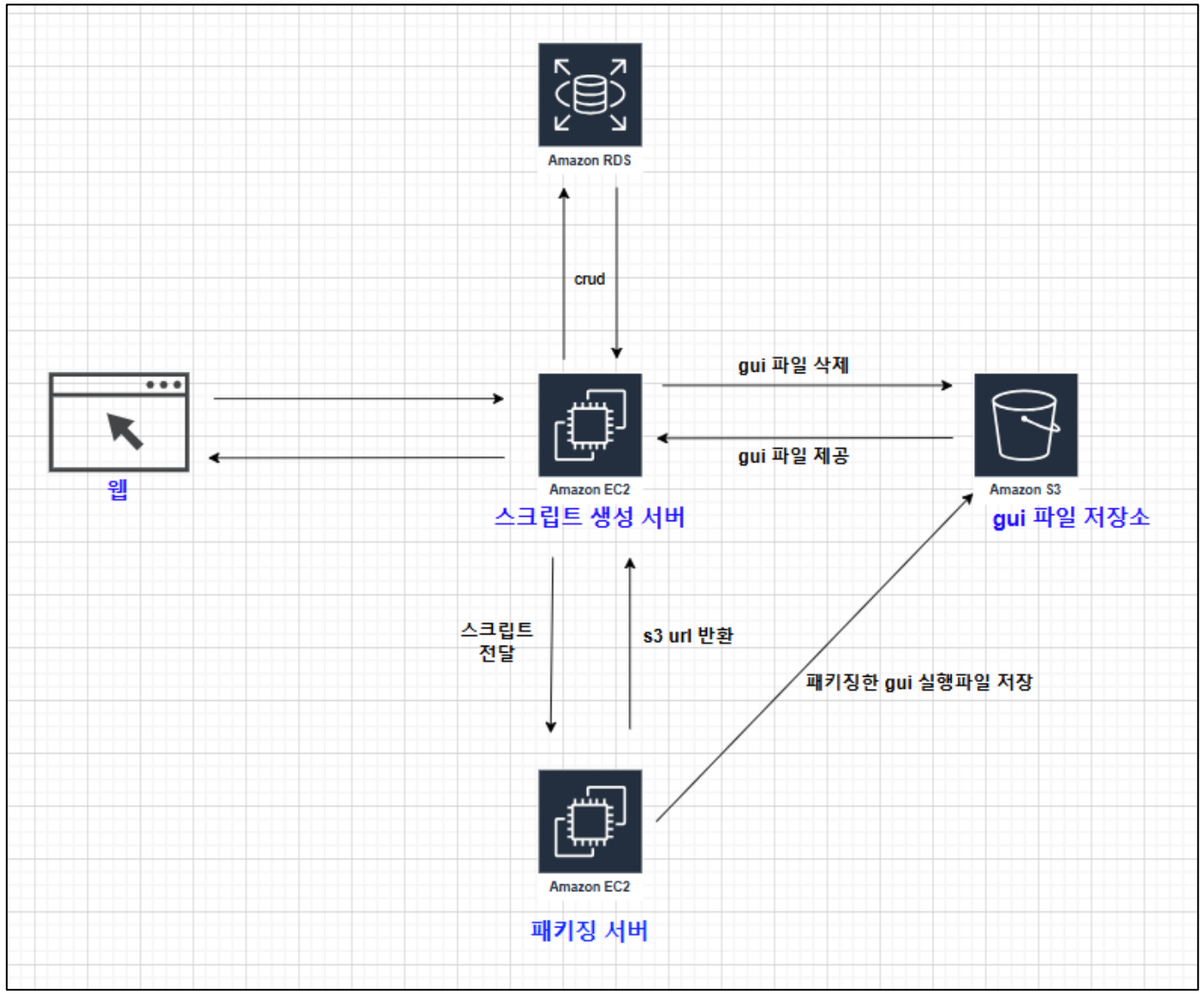
```
-- ----------------------------
-- 레포지토리: WorksheetFunctionRepo (IWorksheetFunctionRepo)
-- ----------------------------

-- 설명: 여러 WorksheetFunction 관계를 한 번에 삽입합니다.
"""
BEGIN;

INSERT INTO "WorksheetFunction" (worksheet_id, function_id) VALUES
  ($1, $2),
  ($3, $4)
  -- 필요한 추가 관계를 여기에 추가하세요
;

COMMIT;
"""

-- 설명: 특정 worksheet_id에 대한 모든 WorksheetFunction 관계를 삭제합니다.
"""
BEGIN;

DELETE FROM "WorksheetFunction" WHERE worksheet_id = $1;

COMMIT;
"""
```

```
-- ----------------------------
-- 레포지토리: WorksheetRepo (IWorksheetRepo)
-- ----------------------------

-- 설명: 필터를 기반으로 Worksheet 레코드를 조회합니다.
"""
SELECT id::text, name, owner_id, main_block, blocks FROM "Worksheet" WHERE ($1 IS NULL OR id = $1)
 AND ($2 IS NULL OR owner_id = $2);
"""

-- 설명: 새로운 Worksheet 레코드를 생성합니다.
"""
BEGIN;

INSERT INTO "Worksheet" (id, name, owner_id, main_block, blocks) VALUES ($1, $2, $3, $4, $5) RETURNING
id::text, name, owner_id, main_block, blocks;

COMMIT;
"""

-- 설명: 기존 Worksheet 레코드를 업데이트합니다.
"""
BEGIN;

UPDATE "Worksheet" SET name = $1,
  owner_id = $2,
  main_block = $3,
  blocks = $4 WHERE id = $5 RETURNING id::text, name, owner_id, main_block, blocks;

COMMIT;
"""

-- 설명: 특정 Worksheet 레코드를 삭제합니다.
"""
BEGIN;

DELETE FROM "Worksheet" WHERE id = $1;

COMMIT;
"""
```

SQL문 – 서비스 제공

```
-- ---------------------------
레포지토리: KoLawListRepo
-- ---------------------------
DO $$
DECLARE
  keyword VARCHAR(255) := '담은'; -- 검색 키워드
  department_name VARCHAR(255) := '소관부처명'; -- 검색할 소관부처명
  start_date VARCHAR(10) := '2024-01-01'; -- 시행일자 범위 시작
  end_date VARCHAR(10) := '2024-12-31'; -- 시행일자 범위 끝
BEGIN
  -- SQL 실행
  EXECUTE format(
    $$
    SELECT *
    FROM "KoLawList"
    WHERE
      "법령명한글" LIKE '%%%s%%'
      AND "소관부처명" LIKE '%%%s%%'
      AND "시행일자" BETWEEN '%s' AND '%s'
    $$,
    keyword, department_name, start_date, end_date
  );
END $$;
```

시스템 구조


Amazon RDS
crud
gui 파일 삭제
gui 파일 제공
웹
Amazon EC2
스크립트 생성 서버
Amazon S3
gui 파일 저장소
스크립트
전달
s3 url 반환
패키징한 gui 실행파일 저장
Amazon EC2
패키징 서버

개발 환경

Web Application Server (WAS)

Python

- Django Rest Framework (Main Server)
- Fast Api (Packaging Server)

DBMS

PostgreSQL

* Django에서 local sqlite도 활용.

GUI 프로그램 생성용 API 서버

Python

- FastAPI
- Pyinstaller (패키징)
- Pyqt5 (GUI 프레임워크)

Web UI

Javascript

- React
- blockly

데이터 제공 사이트

국가법령정보 공동활용 - https://open.law.go.kr/