

MIDAS CIVIL NX Open API 이해와 실무 활용

건설사업팀

© MIDAS IT Co., Ltd

중급 기술자를 위한 PSC 교량의 이해 및 실습에 이어 ' MIDAS CIVIL NX Open API 이해와 실무 활용 ' 교육을 진행하도록 하겠습니다.

이 시간에는 단순 반복적인 모델링 업무에서 벗어나, 어떻게 하면 해석과 설계를 자동화하고 더 높은 부가가치를 창출할 수 있는지,

그 핵심 열쇠인 API에 대해 심도 있게 알아보겠습니다. 이 시간이 여러분의 기술 경쟁력을 한 단계 끌어올리는 중요한 전환점이 되기를 바랍니다.

CONTENTS

01 Part 1: API의 필요성

- 기존 (MCT) 방식의 특징과 자동화 영역

02 Part 2: API란 무엇인가? - 개념 정복하기

- API의 개념: 소통을 위한 약속된 규칙
- MIDAS Open API: 무한한 확장성

03 Part 3: CIVIL NX API 심층 분석 - 작동 원리

- API 통신의 3가지 핵심 요소: 주소(URL), 방식(Method), 내용(JSON)
- 핵심 1. URI (Endpoint): 명령을 전달할 고유 주소
- 핵심 2. HTTP Method: 4가지 요청 방식 (POST, GET, PUT, DELETE)
- 핵심 3. JSON: 데이터를 담는 표준 그릇
- JSON 심화 (1): Table Format (DB 형식 데이터)
- JSON 심화 (2): Operation Function Format (기능 실행 데이터)
- 데이터 흐름 종합: CIVIL NX와 외부 프로그램의 통신 과정

04 Part 4-7: 실전! Excel VBA 연동 자동화

- Part 4. 실습 목표: VBA 기본 설정
- Part 5. 실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링
- Part 6. 실습 목표: FCM 교량의 사공단계해석
- Part 7. 실습 목표: 지하철 정거장의 해석 및 설계

© MIDAS IT Co., Ltd

오늘 교육은 총 4개의 파트로 구성되어 있습니다.

Part 1에서는 우리가 왜 API를 배워야 하는지, 기존의 MCT 방식과 비교하여 API의 필요성을 알아보겠습니다.

Part 2에서는 API가 정확히 무엇인지, 그 기본 개념을 확실하게 정립하겠습니다.

Part 3에서는 CIVIL NX API가 실제로 어떻게 작동하는지, 그 핵심 원리를 심층적으로 분석해 보겠습니다.

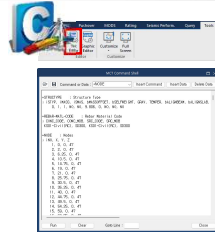
Part 4에서는 앞서 배운 이론을 바탕으로, Excel VBA와 CIVIL NX를 연동하여 RC 라멘교 자동 모델링 및 해석, FCM 모델링, 지하철 정거장 변경 및 재해석 하는 실습을 단계별로 진행하겠습니다.

그럼 Part 1부터 시작하겠습니다.

Part 1

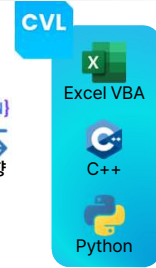
기존 (MCT) 방식의
특징과 자동화 영역

VBA가 텍스트 명령어(MCT) 파일을 생성	핵심원리	VBA가 CIVIL NX의 기능을 직접 제어
Shell 프로그램 / 외부에서 실행 파일을 호출	연결 방식	CIVIL NX와 VBA가 직접 연결
단방향 [VBA → MCT 파일 → Midas] Midas로부터 VBA로의 실시간 피드백 불가	데이터 통신	양방향 [VBA ↔ Midas] VBA의 요청과 Midas의 응답이 실시간으로 이뤄짐
열려있는 모델 정보를 읽어올 수 없음	모델 접근성	Get 함수를 통해 현재 모델의 모든 데이터를 실시간으로 조회 가능
MCT 파일을 통해 새로운 모델을 생성하는 것만 가능, 기존 모델의 '수정' 또는 '부분 변경' 불가	모델 제어	Put, Delete 함수로 절점, 부재, 하중 등을 실시간으로 생성, 수정, 삭제 가능
[텍스트 파싱] 해석 후 생성된 MGT 엑셀 리포트를 파일을 직접 가져와야 함	결과 데이터 접근	[API 호출] API 함수로 프로그램의 결과 DB에 직접 접근하여 반환 받음
Midas 버전 변경으로 인하여 리포트 양식이 조금만 바뀌어도 텍스트 파싱 로직 전체가 오류 발생	데이터 호환	Midas 버전 변경이 되어도 API 함수의 규격(이름, 인자)은 하위 호 환성을 유지하므로 코드가 깨지지 않음
CIVIL 화면 제어 불가, 결과 이미지 자동 캡처 불가	시각화	특정 뷰, 특정 결과 (BMD, SFD, Shape 등)를 이미지 파일로 저장 가능
동일한 형식의 교량 모델 일괄 생성, 단순 배치 해석	활용 예시	설계 최적화(결과값 → 단면변경 → 재해석 루프), 자동화된 구조 계산서 (결과값 + 이미지 자동 삽입)



MCT Shell

→ 단방향

[JSON]
↔ 양방향

자, 그럼 비교표를 보시겠습니다. 이 표는 우리가 기존에 사용하던 **MCT/Shell 방식**과 **CIVIL NX**에서 새롭게 활용하는 **API 방식**이 실무에서 구체적으로 어떻게 다른지 **9가지 항목**으로 상세하게 비교하고 있습니다.

MCT는 텍스트를 일방적으로 전달하는 '**단방향**'방식인 반면, API는 실시간으로 데이터를 주고받는 '**양방향**' 방식입니다.

이 '단방향'과 '양방향'이라는 차이가 구체적으로 어떤 실무적 차이를 만드는지, 9가지 세부 항목을 통해 하나씩 짚어보겠습니다.

1. 가장 근본적인 차이는 '연결 방식'과 '데이터 통신'입니다.

MCT는 VBA가 텍스트 명령어(MCT) 파일을 '생성'하면, 'Shell 프로그램'이 이 파일을 CIVIL NX로 전달하는 방식입니다.

연결이 일회성이고 일방적이라 '**단방향**' 통신입니다. 즉, VBA는 명령을 내릴 뿐, Midas가 그 작업을 성공했는지, 현재 상태가 어떤지 전혀 피드백을 받을 수 없습니다.

반면 **API**는 VBA와 CIVIL NX 프로그램이 '직접 연결'됩니다. 그래서 VBA의 요청과 Midas의 응답이 실시간으로 이뤄지는 '**양방향**' 통신이 가능합니다.

2. 이 '양방향' 통신이 가능하기 때문에, '모델 제어'의 범위가 완전히 달라집니다.

MCT는 단방향이라, 이미 열려있는 모델의 정보를 '읽어올 수 없습니다'.

또한, 기존 모델의 '수정'이나 '부분 변경'이 불가능하고, 오직 '새로운 모델을 생성'하는 것만 가능합니다.

하지만 **API**는 양방향이므로, **Get 함수**를 사용해 현재 모델의 모든 데이터를 실시간으로 '조회'할 수 있습니다.

또한 **Post(생성)**, **Put(수정)**, **Delete(삭제)** 함수를 이용해 절점, 부재, 하중 등을 실시간으로 생성,

수정, 삭제하는 것이 모두 가능합니다.

3. '결과'를 다루는 방식에서도 큰 차이가 납니다.

MCT는 해석 결과를 얻기 위해 MGT 같은 거대한 텍스트 리포트 파일을 VBA로 다시 가져와야 합니다.

그리고 그 텍스트 안에서 내가 원하는 값을 찾기 위해 복잡한 '**텍스트 파싱**'(문자열 분석) 작업을 해야 합니다.

API는 API 함수를 호출하여 프로그램의 결과 DB에 '직접 접근'합니다.

예를 들자면 " 10번 부재의 최대 모멘트 값을 줘 " 라고 직접 물어보고 값을 바로 반환받습니다.

이 차이가 '데이터 호환성', 즉 안정성 문제를 만듭니다. MCT의 '파싱' 방식은 Midas 버전이 업그레이드되어 리포트 양식이 조금만 바뀌어도, 해석이 진행되지 않는 치명적인 오류가 발생합니다.

API는 Midas 버전이 바뀌어도 API 함수의 규격은 '하위 호환성'을 유지하도록 관리되기 때문에 코드가 깨지지 않고 훨씬 안정적입니다.

마지막으로 '시각화'와 '최종 활용 예시'입니다.

MCT는 화면 제어가 불가능하지만, API는 특정 뷰를 설정하거나, BMD, SFD 같은 특정 결과 다이어그램을 '이미지 파일로 저장'하는 것까지 자동화할 수 있습니다.

따라서 **MCT**는 동일한 형식의 모델을 일괄 생성하는 '단순 배치 작업'에 유용합니다.

반면 **API**는 해석 결과를 실시간으로 받아 단면을 변경하고 재해석하는 '**설계 최적화 루프**'나, 결과값과 이미지를 자동으로 조합하는 '**구조 계산서 자동화**' 같은 훨씬 고차원적인 작업이 가능해집니다.

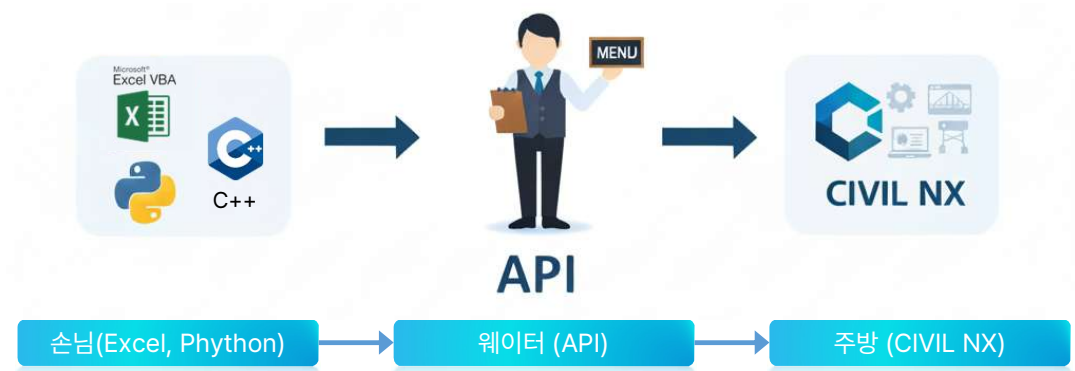
다음 장에서는 이 API가 정확히 무엇인지, 그 개념을 짚어보겠습니다.

Part 2

API의 개념: 소통을
위한 약속된 규칙

API (Application Programming Interface) 정의

- 응용 프로그램(Application)이 서로 상호작용(Interface)하기 위해 사용하는 프로그래밍(Programming) 규칙의 집합
- 핵심 비유: 식당의 '메뉴판'과 '웨이터'
- 손님(외부 프로그램)은 주방(CIVIL NX)의 작동 방식을 몰라도, 메뉴판(API 메뉴얼)의 규칙에 따라 웨이터(API 서버)에게 주문(요청)을 전달하고 원하는 결과(요리)를 얻음



두 번째 파트, '**API란 무엇인가?**' 그 기본 개념부터 확실히 정립해 보겠습니다.

API는 'Application Programming Interface'의 약자입니다. '응용 프로그램이 서로 상호작용하기 위해 사용하는 프로그래밍 규칙의 집합'이라는 정의인데, 조금 복잡하게 들릴 수 있습니다.

여기 보이는 그림처럼, API는 '식당의 메뉴판과 웨이터'에 비유할 수 있습니다. 이게 가장 이해하기 쉽습니다.

우리가 '**손님**' 즉 **Excel VBA, Python, C++** 같은 외부 프로그램이고,

'주방'이 '**CIVIL NX**'라고 생각해 보겠습니다.

손님은 주방의 복잡한 작동 방식 즉, CIVIL NX의 내부 로직을 몰라도, API 매뉴얼 '메뉴판'에 적힌 규칙대로 '웨이터'에게 주문을 전달합니다.

그러면 웨이터(**API**)가 주방(**CIVIL NX**)에 주문을 전달하고, 완성된 요리(결과)를 다시 손님에게 가져다줍니다.

즉, API는 두 프로그램이 소통하기 위한 '**약속된 규칙**' 또는 '**소통 창구**'라고 생각하시면 됩니다.

Part 2

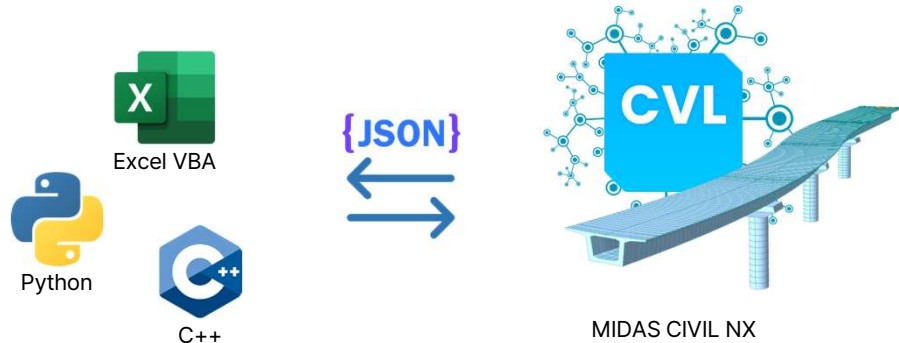
MIDAS Open API:
무한한 확장성

Midas Open API의 특징

- 특정 언어에 종속되지 않고, HTTP 통신을 지원하는 모든 프로그래밍 환경과 연동 가능함
- 사용 가능 언어 예시: Excel VBA, Python, C++, C# 등

데이터 교환 표준: JSON

- 모든 명령어와 데이터는 전 세계적으로 통용되는 웹 표준 데이터 형식인 'JSON'을 통해 교환됨



MIDAS Open API의 가장 큰 특징은 바로 '무한한 확장성'입니다.

앞서 API를 '소통 규칙'이라고 말씀드렸는데, Midas API는 이 규칙을 특정 언어에 종속시키지 않았습니다.

즉, HTTP 통신이라는 표준 방식만 지원한다면, 오늘 우리가 배울 **Excel VBA**는 물론이고 데이터 분석에 강한 **Python**, 또는 속도가 빠른 C++나 C# 등, 여러분이 선호하는 **어떤 프로그래밍 환경** **과도 연동**이 가능합니다.

그리고 이 모든 프로그램들과 Midas가 소통할 때, 데이터 교환은 '**JSON**'이라는 표준 형식을 사용합니다. 모든 명령어와 데이터는 이 JSON이라는 전 세계 웹 표준 데이터 그릇에 담겨 교환됩니다.

다음 장에서는 이 API 통신의 핵심 작동 원리를 좀 더 깊게 파헤쳐 보겠습니다.

Part 3

API 통신의 3가지
핵심 요소

API 통신의 3대 구성 원칙

주소 (Where)

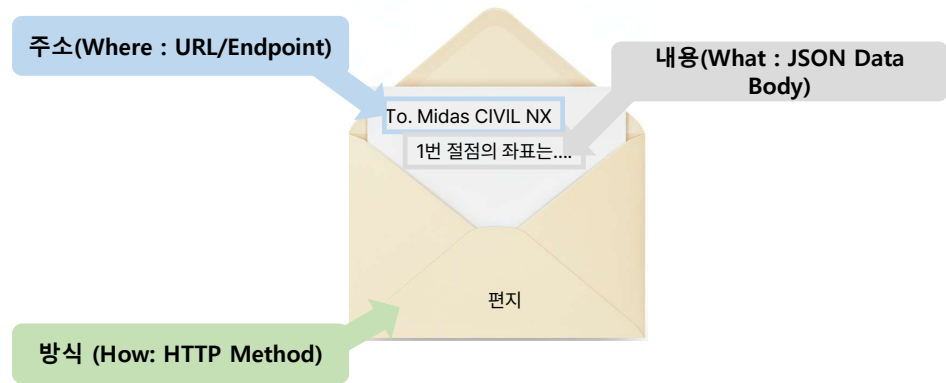
- 어디로 요청을 보낼 것인가? (URL/Endpoint)

방식 (How)

- 어떤 종류의 요청을 할 것인가? (HTTP Method)

내용 (What)

- 내용 (What): 무엇을 요청할 것인가? (JSON Data Body)



세 번째 파트, **API의 작동 원리**입니다.

API 통신은 복잡해 보이지만, 사실 이 **3가지 핵심 요소**로 모든 것이 이루어집니다. 바로 '**주소 (URI)**', '**방식(Method)**', '**내용(JSON)**'입니다.

우리가 **편지**를 보내는 과정을 생각하시면 아주 쉽습니다.

첫번째로는

주소 (Where): ' 어디로 ' 요청을 보낼 것인가? (이것이 'URI/Endpoint' 입니다.)

두번째로는

방식 (How): ' 어떤 종류 ' 의 요청을 할 것인가? (이것이 ' HTTP Method ' 입니다.) (예: ' 보내기 ', ' 가져오기 ' 등) POST, PUT, GET, DELETE

마지막으로

내용 (What): '무엇을' 요청할 것인가? (이것이 'JSON Data Body'입니다.) (예: "1번 절점의 좌표는...")

이 세 가지를 정확하게 조합해서 Midas CIVIL NX로 요청을 보내는 것입니다. 다음 장부터 이 세 가지 요소를 하나씩 자세히 살펴보겠습니다

Part 3

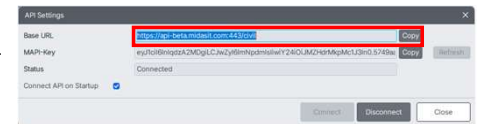
핵심 1. URI (Endpoint): 명령을 전달할 고유 주소

URI (Uniform Resource Identifier)의 구조

- Base URL + Endpoint 로 구성됨

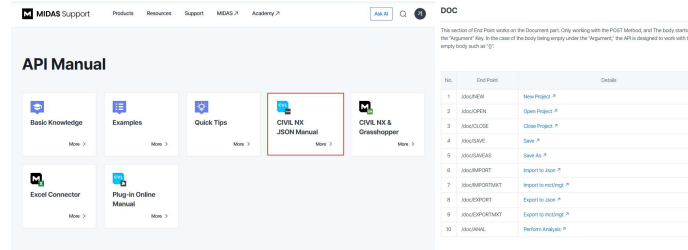
Base URL

- API 서버의 기본 주소. CIVIL NX 사용자는 모두 동일한 주소를 사용함
- 예: <https://moa-engineers.midasit.com:443/civil>



Endpoint

- 절점, 요소, 재료 등 각 기능(리소스)에 접근하기 위한 고유한 상세 주소
- 예: /db/NODE, /db/MATL, /ope/LINEBMLD
- 사용 가능한 모든 Endpoint는 API 매뉴얼에 명시되어 있음
- [MIDAS API Online Manual - MIDAS Support](#)



첫 번째 요소는 **URI**, 즉 '명령을 전달할 고유 주소'입니다.

URI는 '**Base URL**'과 '**Endpoint**' 이 두 가지가 더해져서 구성됩니다.

먼저, 'Base URL'은 API 서버의 기본 주소입니다. 여기 보이는 <https://moa-engineers.midasit.com:443/civil> 이와 같은 주소인데, 이건 CIVIL NX 사용자는 모두 동일한 값을 사용합니다.

우리가 실제로 다루게 될 것은 바로 'Endpoint'입니다.

Endpoint는 **절점, 요소, 재료** 등 각 기능에 접근하기 위한 '고유한 상세 주소'입니다. 예를 들어, 절점 데이터베이스는 /db/NODE, 재료는 /db/MATL 과 같은 주소를 가집니다.

사용 가능한 모든 Endpoint의 목록은 여기 보시는 'MIDAS API Online Manual'에 모두 명시되어 있습니다.

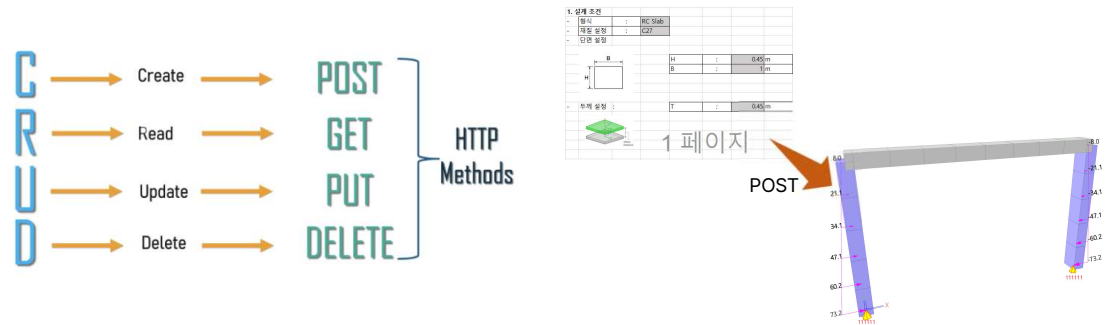
그래서 우리가 어떤 작업을 하고 싶을 땐, 매뉴얼을 찾아보고 "아, 절점을 만들려면 /db/NODE 라는 주소로 요청을 보내야 하는구나" 하고 확인하시면 됩니다.

Part 3

핵심 2. HTTP Method: 4가지 요청 방식

요청의 종류를 정의하는 HTTP 메서드(Methods)

Method	기능	상세설명	CIVIL NX API 활용 예시
POST	생성 (CREATE)	새로운 데이터를 서버에 추가해달라고 요청	새로운 절점, 요소, 재료, 하중 등 생성
GET	조회(READ)	서버에 있는 데이터를 보내달라고 요청	생성된 모델의 정보, 해석 결과 조회
PUT	수정 (UPDATE)	기존 데이터를 새로운 데이터로 변경해달라고 요청	특정 절점의 좌표, 재료의 물성치 변경
DELETE	삭제(DELETE)	특정 데이터를 삭제해달라고 요청	불필요한 하중 케이스, 경계 조건 삭제



두 번째 요소는 **HTTP Method**, 즉 '4가지 요청 방식'입니다.

'주소(URI)'가 우리가 가야 할 '목적지'였다면, 이 'Method'는 그 목적지에 가서 '무엇을 할 것인지'를 정하는 **4가지 행동 방식**입니다.

이 4가지 방식은 데이터베이스 관리 용어인 **CRUD** (Create, Read, Update, Delete)와 정확히 일치합니다.

POST는 생성 (CREATE) 입니다. "이 데이터 줄 테니, 새 절점, 새 요소, 새 하중을 **만들어줘**"라고 요청할 때 씁니다.

GET은 조회 (READ) 입니다. "CIVIL NX에 있는 데이터 좀 **보내줘**"라고 요청하는 겁니다. 생성된 모델 정보나 해석 결과를 가져올 때 사용합니다.

PUT은 수정 (UPDATE) 입니다. "기존 데이터를 이 새 데이터로 **바꿔줘**"라고 요청합니다. 특정 절점의 좌표를 바꾸거나 재료의 물성치를 변경할 때 씁니다.

DELETE는 그대로 삭제 (DELETE) 입니다. "이 데이터 **지워줘**"라고 요청하는 거죠. 불필요한 하중 케이스나 경계 조건을 삭제할 때 사용합니다.

우리가 할 모든 작업은 이 4가지 요청 방식 중 하나로 반드시 구분됩니다.

Part 3

핵심 3. JSON: 데이터를 담는 표준 그릇

JSON (JavaScript Object Notation)의 특징

- 사람과 기계 모두가 쉽게 이해할 수 있는 텍스트 기반의 데이터 교환 형식
- Key-Value 쌍으로 구성: "Key" : "Value"

JSON 기본 구조

- {} (중괄호): JSON 객체(Object)를 의미
- "" (큰따옴표): Key 이름이나 문자열 Value를 감쌈
- : (콜론): Key와 Value를 구분
- , (쉼표): 여러 Key-Value 쌍을 구분

JSON 기본 구조 예시 (절점 정보)

JSON NODE EXAMPLE

```
{
  "Assign": {
    "1": { "X": -1, "Y": -1, "Z": -1 },
    "2": { "X": -2, "Y": -2, "Z": -2 },
    "3": { "X": -3, "Y": -3, "Z": -3 }
  }
}
```

- Assign: 수행할 작업의 최상위 Key
- "1", "2", "3": 각 절점의 번호(ID)
- X, Y, Z: 각 절점의 좌표 값

세 번째 요소는 **JSON**, 즉 '데이터를 담는 표준 그릇'입니다.

우리가 '주소(URI)'로 찾아가서, '방식(Method)'을 정했다면, 실제로 전달할 '내용(What)'이 바로 이 JSON입니다.

JSON은 'JavaScript Object Notation'의 약자인데요, 이 세세한 내용은 정확히 모르셔도 됩니다.

중요한 건, 이것이 "Key-Value 쌍"으로 구성된, 사람과 기계 모두가 쉽게 이해할 수 있는 **텍스트 기반의 데이터 교환 형식**이라는 점입니다.

아래의 예시를 보겠습니다. 이걸 절점 정보를 나타내는 JSON입니다.

"Assign" 이라는 것이 'Key'이고, 그 값(Value)으로 {...} 중괄호 객체가 있습니다.

그 안을 보면, "1", "2", "3" 이라는 각 절점 번호가 다시 'Key'가 됩니다.

그리고 각 절점 번호의 'Value'로, { "X": -1, "Y": -1, "Z": -1 } 처럼 X, Y, Z 좌표 값을 담는 구조입니다.

이처럼 JSON은 중괄호{}(객체), 큰따옴표""(Key 또는 문자열), 콜론:(Key-Value 구분), 쉼표,(여러 쌍 구분) 이 기본 구조를 가집니다.

우리가 CIVIL NX로 보내는 모든 데이터는 이 형식으로 만들어서 보내게 됩니다.

Part 3

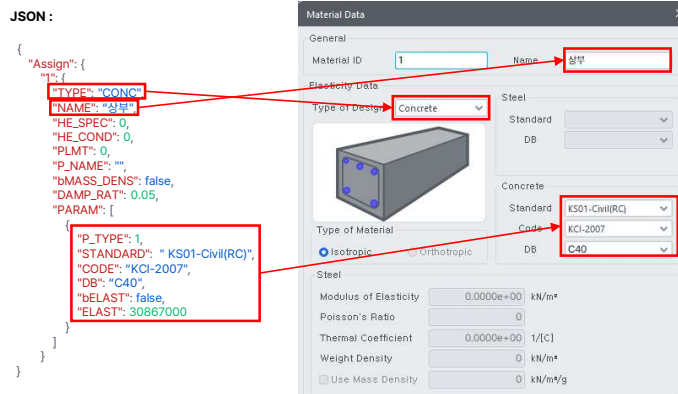
JSON 심화 (1):
Table Format
(DB 형식 데이터)

Table Format의 특징

- CIVIL NX의 테이블(표) 형식 데이터와 직접적으로 대응되는 구조
- 주로 모델의 기본 정보(재료, 단면, 절점, 요소 등)를 다룰 때 사용됨
- 메서드: 데이터의 생성(POST), 조회(GET), 수정(PUT), 삭제(DELETE) 모두 활용 가능

데이터 흐름 예시: 재료(Material) 데이터

- CIVIL NX의 재료 대화상자 또는 테이블에 입력하는 모든 항목이 JSON의 Key-Value 값과 1:1로 매핑됨



9

CIVIL NX에서 사용되는 JSON에는 크게 두 가지 형식이 있습니다. 그 첫 번째가 'Table Format'입니다.

이름 그대로, **CIVIL NX의 테이블(표) 형식 데이터와 직접적으로 대응되는 구조**입니다.

우리가 CIVIL NX에서 작업할 때 만나는 **재료, 단면, 절점, 요소**처럼 모델의 '기본 정보'를 다룰 때 주로 사용됩니다.

이 Table Format은 데이터베이스의 데이터를 직접 다루는 방식이기 때문에, 앞서 배운 4가지 HTTP Method, 즉 **POST, GET, PUT, DELETE**를 모두 활용할 수 있습니다.

예시를 보겠습니다. 왼쪽은 재료데이터의 JSON 예시이고, 오른쪽은 실제 CIVIL NX의 재료 입력 대화상자입니다.

자세히 보시면, CIVIL NX 대화상자에 입력하는 'Material ID', 'Name', 'Type of Design' (TYPE), 'Standard', 'Code', 'DB' 같은 **모든 항목이 JSON의 'Key-Value' 값과 정확히 1:1로 매핑되는 것**을 볼 수 있습니다.

이것이 바로 Table Format의 특징입니다.

Part 3

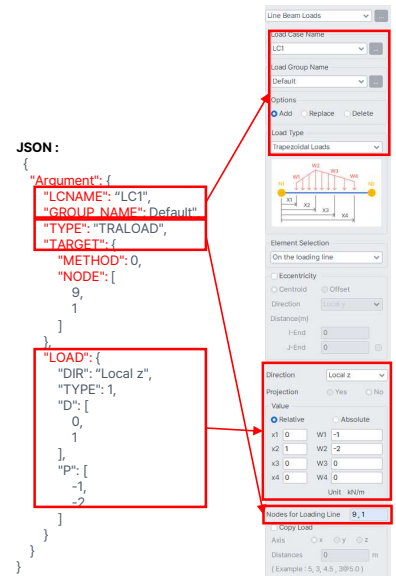
JSON 심화 (2): Operation Function Format (기능 실행 데이터)

Operation Function Format의 특징

- CIVIL NX의 특정 '기능(Function)'이나 '명령(Operation)'을 실행할 때 사용됨
- 데이터를 DB에 직접 저장하기보다, 입력된 인자(Argument)를 바탕으로 프로그램이 내부 계산을 수행하고 그 결과를 테이블에 저장함
- 메서드: 특정 기능을 '실행'하는 요청이므로 POST만 사용 가능

데이터 흐름 예시: 보 요소 하중 (Line Beam Load)

- 하중의 위치, 방향, 크기 등 기능 실행에 필요한 조건들을 JSON으로 전달하면, CIVIL NX가 이를 계산하여 최종적으로 각 요소에 적용된 하중 값을 Beam Load Table에 저장함



JSON의 두 번째 형식은 'Operation Function Format'입니다.

이 형식은 CIVIL NX의 특정 '기능'이나 '명령'을 실행할 때 사용됩니다.

앞서 본 Table Format이 DB의 데이터를 직접 다루는 방식이라면, 이 Operation Format은 조금 다릅니다.

이것은 우리가 입력한 인자(Argument)를 바탕으로, 프로그램이 **내부 계산을 수행**하고 그 **결과를 테이블에 저장**하는 방식입니다.

예를 들어 오른쪽의 '**보 요소 하중(Line Beam Load)**' 기능을 보겠습니다.

우리는 하중 케이스 이름, 하중의 종류, 하중을 적용할 노드, 그리고 하중의 값 등을 "Argument"라는 Key 값으로 묶어서 JSON을 전달합니다.

그러면 CIVIL NX는 이 정보들을 받아 내부 계산을 수행한 뒤, 최종적으로 계산된 하중 값을 Beam Load Table에 저장해 줍니다.

이처럼 이 형식은 특정 기능을 '실행'하라고 요청하는 것이기 때문에, 4가지 Method 중에서 **POST만 사용** 가능합니다.

Part 3

데이터 흐름 종합:
CIVIL NX와 외부
프로그램의 통신 과정

API 통신 프로세스 요약

1. 요청(Request) 생성 (by 외부 프로그램 - 예: Excel VBA)

- API 매뉴얼을 참고하여 목적에 맞는 URI(주소), Method(방식), JSON(내용) 을 조합

2. 요청 전송

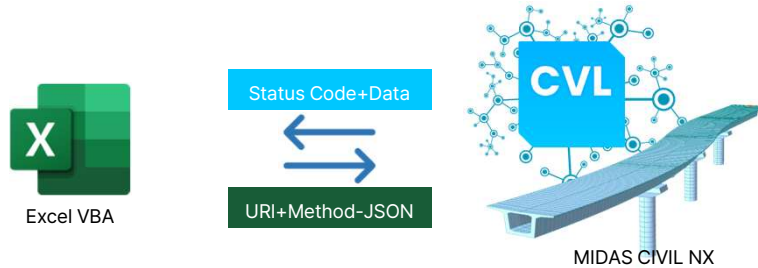
- HTTP 프로토콜을 통해 CIVIL NX API 서버로 요청 메시지를 전송

3. 요청 처리 (by CIVIL NX API 서버)

- 수신된 요청의 유효성(MAPI-Key, JSON 형식 등)을 검증검증 완료 시, CIVIL NX 프로그램에 해당 명령을 수행하도록 지시

4. 응답(Response) 반환

- 처리 결과를 나타내는 상태 코드와 데이터(요청 성공/실패 여부, 조회된 데이터 등)를 다시 외부 프로그램으로 전송



© MIDAS IT Co., Ltd.

11

자, 지금까지 말씀드린 세 가지 요소(URI, Method, JSON)를 모두 종합하여, **CIVIL NX와 외부 프로그램의 전체 통신 과정**을 요약해 보겠습니다.

이 과정은 총 4단계로 이루어집니다:

요청 생성

먼저, Excel VBA와 같은 외부 프로그램에서 API 매뉴얼을 참고합니다.

"절점을 생성"하려면 "어떤 URI(주소)로", "어떤 Method(방식)를 써서", "어떤 JSON(내용)을" 보내야 하는지 확인하고, 이 세 가지를 조합해 '요청 메시지'를 만듭니다.

요청 전송

생성된 요청 메시지를 HTTP 프로토콜을 통해 CIVIL NX API 서버로 '전송'합니다.

요청 처리

서버는 요청을 받으면, "이 요청이 유효한가?" (MAPI-Key가 맞는지, JSON 형식이 올바른지 등)를 '검증'합니다.

검증이 완료되면, 서버는 CIVIL NX 프로그램 본체에 "이 명령을 수행해"라고 지시합니다.

응답 반환

마지막으로, CIVIL NX는 명령을 수행하고 난 '처리 결과'를 다시 외부 프로그램(Excel VBA)으로 '전송'해 줍니다.

이 1-2-3-4 단계가 실시간 '양방향'으로 이루어지는 것이 바로 API 통신의 핵심 프로세스입니다.

Part 4

실습 목표:
VBA 기본 설정

실전! Excel VBA 연동 자동화

실습 개요

- 목표: Excel 시트에 정의된 설계 변수를 기반으로 RC 라멘교, FCM 교량, 지하철 정거장 모델의 결과 검토까지
- 자동화프로세스: Excel 시트 입력 → VBA 매크로 실행 → CIVIL NX 모델 자동 생성

Step 0. API 연동 사전 설정

API 통신을 위한 기본 정보 설정

- Base-URL: API 서버 주소 (고정값)
- MAPI-Key: 사용자 인증 키. CIVIL NX
[메인메뉴 > Apps > API Settings] 에서 복사하여 사용
- 단위계: API 통신 시 사용할 기본 단위(힘, 길이)를 명시

0. 설정

-	Base-URL	:	https://moa-engineers.midasit.com:443/civil
-	MAPI-Key	:	liwiY24iOiI4VXRqcTc1dFFnln0.62898b04f114
-	단위계	:	힘 단위 kN 길이단위 m

Base Url : 모두 같은 주소 사용 수정 필요 없음
Mapi-key : 인증 정보

자, 이제 이론은 충분히 다뤘으니, Part 4에서, 실제 **Excel VBA를 연동한 자동화 실습**을 시작하겠습니다.

실습 목표는 **Excel 시트에 정의된 설계 변수를 기반으로 RC 라멘교, FCM 교량, 지하철 정거장 모델의 결과 검토까지**하는 것입니다.

프로세스는 간단합니다: Excel 시트에 값을 입력하고 VBA 매크로를 실행하면, CIVIL NX에 데이터가 전송되어, 모델링이되고 해석이 실행되고, 결과를 가져오게 되는 것입니다.

CIVIL NX의 API를 사용하기 위해서 가장 먼저 해야 할 일은 **API 연동 사전 설정**입니다. API 통신을 위해 Excel 시트에 이 두가지 기본 정보를 설정해야 합니다.

첫번째로는 Base-URL입니다. 가장처음으로 API 서버 주소를 CIVIL NX에서 COPY 하시고 엑셀에 입력하면 됩니다. 다만 이 값은 고정값이므로 입력이 되어 있으시다면, 수정할 필요가 없습니다.

두번째로는 MAPI-Key입니다. MAPY KEY는 사용자 인증 키입니다. 이 키는 CIVIL NX 프로그램의 [메인메뉴 > Apps > API Settings] 에 들어가셔서 'Copy' 버튼을 눌러 복사한 뒤, 엑셀 시트에 붙여넣기 해주시면 됩니다.

Part 5

실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링

Step 1. 설계 조건 입력 (재료/단면 정의)

Excel 시트 입력

- [1. 설계 조건] 섹션에 재료 강도(C27), 단면 제원(폭, 높이) 등 입력

VBA 로직

- 1. 데이터 읽기: Excel 시트의 특정 셀(Cell)에서 'C27', '0.45' 등의 값을 변수로 읽어옴
- 2. JSON 생성: 읽어온 변수들을 API 매뉴얼에 정의된 재료(/db/MATL), 단면(/db/SECT) JSON 형식에 맞춰 텍스트로 조합
- 3. API 요청: /db/MATL, /db/SECT 엔드포인트로 POST 메서드를 사용하여 생성된 JSON 데이터를 전송

1. 설계 조건			
- 형식	:	RC Slab	
- 재질 설정	:	C27	→ 압축강도 설정 > "C27" 과 같이 입력하여 압축 강도 설정
- 단면 설정			
		H	: 0.45 m
		B	: 1 m
		→ 사각단면의 폭과 높이 설정	
		T	: 0.45 m
		→ 판요소의 면 내/외 두께 설정	



1 페이지

자, 실습 첫 번째 단계, **설계 조건 입력**입니다. 여기 보시는 엑셀 시트의 [1. 설계 조건] 섹션에 자동 생성을 위한 기본 정보들을 입력합니다.

재료 강도, 여기서는 'C27'을 입력하고, 사각 단면의 폭(B)과 높이(H), 그리고 판요소의 면 내/외 두께(T) 값을 각각 셀에 입력합니다.

그러면 VBA 매크로는 다음과 같이 작동합니다. 엑셀 시트의 특정 셀(Cell)에서 'C27', '0.45' 같은 값들을 변수로 읽어옵니다.

읽어온 변수들을 API 매뉴얼에 정의된 재료(/db/MATL)와 단면(/db/SECT) JSON 형식에 맞춰 텍스트를 조합합니다.

마지막으로, /db/MATL과 /db/SECT 엔드포인트로 **POST** 메서드를 사용하여 생성된 JSON 데이터를 CIVIL NX로 전송합니다.

이렇게 하면 CIVIL NX 내부에 재료와 단면이 정의됩니다.

Part 5

실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링

Step 2. 구조물 형상 및 경계조건 생성

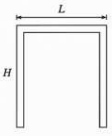
Excel 시트 입력

- [2. 모델링] 섹션에 구조물 제원(벽체 높이, 지간)

VBA 로직

- 1. 절점 좌표 계산: 입력된 벽체 높이(H)와 지간(L)을 기준으로 모든 절점(Node)의 X, Y, Z 좌표를 For문 등을 통해 순차적으로 계산
- 2. 절점/요소 JSON 생성: 계산된 좌표로 절점 생성용 JSON을 만들고, 생성된 절점들을 연결하여 요소(Element) 생성용 JSON을 만들
- 3. API 요청: 각 엔드포인트(/db/NODE, /db/ELEM)로 POST 요청 전송

2. 모델링

-	벽체 높이	H	:	4 m
-	지간 구성	L	:	6 m
				
-	경계조건	*스프링 지점으로 입력		
	변위스프링	:		180000 kN/m
	회전스프링	:		90000 kN*m/rad

Node JSON

```
{
  "Assign": {
    "1": {
      "X": 0,
      "Y": 0,
      "Z": 0
    }
  }
}
{
  "Assign": {
    "2": {
      "X": 0,
      "Y": 0,
      "Z": 1
    }
  }
}
{
  "Assign": {
    "5": {
      "X": 0,
      "Y": 0,
      "Z": 4
    }
  }
}
{
  "Assign": {
    "6": {
      "X": 1,
      "Y": 0,
      "Z": 4
    }
  }
}
```

다음은 구조물 형상 및 경계조건 생성입니다.

Excel 시트 입력: [2. 모델링] 섹션에 구조물의 기본 제원인 **벽체 높이(H)**와 **지간(L)** 값을 입력합니다. 그리고 하단에 지점 조건인 **경계 조건**, 즉 변위 스프링과 회전 스프링의 강성 값을 입력합니다.

VBA 매크로의 내부 로직은 다음과 같이 작동합니다.

VBA가 엑셀에서 입력된 벽체 높이(H)와 지간(L) 값을 읽어와, For문 등을 통해 모델링에 필요한 모든 절점(Node)의 X, Y, Z 좌표를 순차적으로 계산합니다.

계산된 좌표를 바탕으로 절점 생성용 JSON을 만들고, 이 절점들을 연결하여 요소생성용 JSON을 만듭니다.

마지막으로, 준비된 각 JSON 데이터(절점, 요소)를 해당 엔드포인트(/db/NODE, /db/ELEM)로 **POST** 요청을 전송하여 기본적인 모델의 형상을 완성합니다.

Part 5

실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링

Step 2. 구조물 형상 및 경계조건 생성

Excel 시트 입력

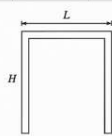
- [2. 모델링] 경계 조건(스프링 강성) 입력

VBA 로직

- 1. 경계조건 JSON 생성: 지점부 절점 번호와 입력된 스프링 강성 값으로 경계조건(/db/ NSPR) JSON 생성
- 2. API 요청: 각 엔드포인트(/db/NSPR)로 POST 요청 전송

2. 모델링

- 벽체 높이	H	:	4 m
- 지간 구성	L	:	6 m



- 경계조건			
*스프링 지점으로 입력			
변위스프링	:	180000	kN/m
회전스프링	:	90000	kN*m/rad

Point Spring Supports

Boundary Group Name: Default

Options: ☐ Add ☒ Replace ☐ Delete

Point Spring (Local Direction): Type: Linear

SDx(SPy) SDy(SRy) SDz(SRz)

SDx 180000 kN/m

SDy 180000 kN/m

SDz 180000 kN/m

SRx 90000 kN-m/[rad]

SRy 90000 kN-m/[rad]

SRz 90000 kN-m/[rad]

XYZ 방향 동일하게 입력

스프링 경계 JSON

```
{
  "Assign": {
    "T": {
      "ITEMS": [
        {
          "ID": 1,
          "TYPE": "LINEAR",
          "F.S": [
            false,
            false,
            false,
            false,
            false,
            false
          ],
          "SD": [
            180000,
            180000,
            180000,
            90000,
            90000,
            90000
          ],
          "DAMPING": true,
          "Cr": [
            1,
            2,
            3,
            4,
            5,
            6
          ]
        }
      ],
      "GROUP_NAME": "Service"
    }
  }
}
```

그 다음은 스프링 경계(Point Spring) JSON입니다.

엑셀 시트에서 입력받은 '변위 스프링' 값(180000)과 '회전 스프링' 값(90000)이

이 JSON 내부의 "SDR"이라는 Key의 값, 즉 배열(Array) 값으로 [180000, 180000, 180000, 90000, 90000, 90000] 이런 식으로 차곡차곡 들어가서 Midas로 전달되는 것을 확인할 수 있습니다.

Part 5

실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링

Step 3. 하중 조건 정의

Excel 시트 입력

- [3. 하중 조건] 섹션에 활하중, 토압, 온도하중 등 각 하중의 종류와 크기를 입력

VBA 로직

1. 하중 케이스 생성: '활하중', '토압' 등 정적 하중 케이스(/db/STLD)를 먼저 생성
2. 하중 데이터 할당: 입력된 하중 값을 이용해 보 요소 하중(/db/BMLD), 온도 하중(/db/ETMP), 온도구배하중(/db/GTMP) 등 각 하중 종류에 맞는 JSON 데이터를 생성
3. API 요청: 생성된 하중 케이스 이름(LCNAME)과 함께 해당 하중 데이터를 POST 방식으로 전송

3. 하중조건			
- 고정하중			
자중	:	CIVIL 자동반영	
- 활하중	*본 예제는 자로하중과 같은 하중들을 분포하중으로 가정!		
w	:	1 kN/m	
- 토압	*양토압		
q_max	:	2 kN/m^2	
q_min	:	1 kN/m^2	
- 온도하중	콘크리트교의 온도변화 범위		
TU	:	10 °C	
슬래브 상하연 온도차			
TG	:	5 °C	

```

{
  "BMLD": {
    "T": {
      "ITEMS": [
        {
          "ID": 1,
          "LCNAME": "LC3",
          "GROUP_NAME": "",
          "CMD": "LINE",
          "TYPE": "UNILOAD",
          "DIRECTION": "GX",
          "USE_PROJECTION": false,
          "USE_ECCEN": false,
          "D": {
            "0": 0,
            "1": 1,
            "0": 0,
            "0": 0
          },
          "P": {
            "2": 2,
            "1.75": 1.75,
            "0": 0,
            "0": 0
          },
          "USE_ADDITIONAL": false,
          "ADDITIONAL_LEND": 0,
          "ADDITIONAL_JEND": 0,
          "USE_ADDITIONAL_JEND": false
        }
      ]
    }
  }
}

```

```

{
  "ETMP": {
    "S": {
      "ITEMS": [
        {
          "ID": 5,
          "LCNAME": "LC4",
          "GROUP_NAME": "",
          "TEMP": 10
        }
      ]
    }
  },
  "GTMP": {
    "S": {
      "ITEMS": [
        {
          "ID": 5,
          "LCNAME": "LC5",
          "GROUP_NAME": "",
          "TYPE": 1,
          "TZ": 5,
          "USE_HZ": true,
          "HZ": 0,
          "TY": 0,
          "USE_HY": true,
          "HY": 0
        }
      ]
    }
  }
}

```

온도 구배 하중 JSON

자, 이제 모델의 뼈대와 경계조건이 완성되었으니, **하중 조건 정의**로 넘어가겠습니다.

[3. 하중 조건] 섹션에 우리가 적용할 하중의 종류와 크기를 입력합니다. 자중은 'CIVIL 자동 반영'으로 설정하고, **활하중**, **토압**, 그리고 **온도하중**, **온도차** 값을 엑셀 시트에 직접 입력합니다.

VBA 매크로는 이 값들을 받아 다음과 같이 작동합니다.

먼저, '활하중', '토압' 등 엑셀에 정의된 이름으로 정적 하중 케이스(/db/STLD)를 생성합니다.

그 다음, 엑셀에서 입력받은 하중 값(예: 1 kN/m)을 이용해 '보 요소 하중'이나 '온도 하중', 온도구배하중, '분포하중' 처럼 각 하중 종류에 맞는 JSON 데이터를 생성합니다.

마지막으로, 생성된 하중 케이스 이름과 하중 데이터를 묶어서 **POST** 방식으로 CIVIL NX에 전송합니다. 오른쪽 하단은 이렇게 생성된 하중 조건 JSON의 예시입니다.

Part 5

실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링

Step 4. 하중 조합 생성

Excel 시트 입력

- [4. 하중 조합] 섹션에 조합 이름, 타입(Add/Envelope), 조합할 하중 케이스와 계수를 입력

VBA 로직

- 1. 데이터 구조화: 테이블 형태의 하중 조합 정보를 읽어와서 하중 조합 생성용(/db/LCOM-GEN) JSON 구조에 맞게 데이터를 재구성
- 2. API 요청: /db/LCOM-GEN 엔드포인트로 POST 메서드를 사용하여 하중 조합 데이터 전송

자동화의 장점

- 수십, 수백 개의 복잡한 하중 조합을 실수 없이 신속하게 생성 가능

4. 하중조합							
하중조합	Type	Case 1	Factor 1	Case 2	Factor 2	Case 3	Factor 3
LCB1	Add	LC1	1.3	LC3	1.7		
LCB2	Add	LC1	1.3	LC4	1.3		
LCB3	ENV	LCB1	1	LCB2	1		

하중조합명

하중조합 타입
Add : Add
Envelope : ENV

단일 하중조건 (Envelope의 경우 하중조합) 과 조합 계수입력

```
{
  "Assign": {
    "1": {
      "NAME": "LC1",
      "ACTIVE": "ACTIVE",
      "bCB": false,
      "iTYPE": 0,
      "DESC": "",
      "vCOMB": [
        {
          "ANAL": "ST",
          "LCNAME": "LC1",
          "FACTOR": 1
        }
      ]
    }
  }
}
```

RC 라멘교 실습의 마지막 단계, **Step 4. 하중 조합 생성**입니다.

[4. 하중 조합] 섹션에 있는 테이블을 보겠습니다. 여기에 하중 조합 이름(예: LCB1), 조합 타입(Type)을 'Add' 또는 'Envelope'으로 입력합니다. 그리고 조합에 사용할 각 하중 케이스(Case 1, Case 2...)와 해당 계수(Factor 1, Factor 2...)를 차례대로 입력합니다.

VBA 매크로는 이 테이블 정보를 읽어와서 다음과 같이 작동합니다.

엑셀 테이블 형태의 데이터를 하중 조합 생성용 API 엔드포인트(/db/LCOM-GEN)가 요구하는 JSON 구조에 맞게 재구성합니다.

재구성된 JSON 데이터를 /db/LCOM-GEN 엔드포인트로 **POST** 메서드를 사용해 전송합니다.

이 기능의 가장 큰 장점은, 수십, 수백 개가 넘는 복잡한 하중 조합도 엑셀에서 관리하며, 실수 없이 신속하게 생성할 수 있다는 것입니다.

Part 5

실습 목표: Excel 입력을 통한 RC라멘교 자동 모델링

Step 5. 해석 및 결과 검토

해석 수행 및 결과 출력

- 'RAHMEN' 버튼 클릭 – Setting Sheet

결과 검토

- 재하 하중 검토 – 하중계산 Sheet
- 절점, 엘리먼트, 경계조건 – 모델링 Sheet
- 자중, 분포하중, 수평토압, 요소온도하중, 온도구배하중 – 하중재하도 Sheet
- 단면력도 -단면력도(LCB1/ LCB2/ENV1) Sheet



자, Step 4에서 하중 조합까지 모두 생성이 완료되었습니다.

RC 라멘교 모델의 **해석 수행 및 결과 검토** 해보겠습니다.

모든 모델링과 하중 입력이 완료되었으므로, 'Setting Sheet'에서 '**RAHMEN**' 버튼을 클릭합니다.

이 버튼은 Midas API로 해석 수행 명령을 전송하고, 해석이 완료되면 결과까지 가져오도록 구성되어 있습니다.

해석이 완료되면, VBA 매크로가 API를 통해 CIVIL NX로부터 결과 데이터를 다시 가져와 엑셀 시트의 각 결과 시트를 업데이트합니다. 우리는 이 엑셀 시트들만 확인하면 됩니다.

먼저 '**하중계산 Sheet**'에서 재하 하중이 올바르게 검토되었는지 확인하고, '**모델링 Sheet**'에서 절점, 요소, 경계조건이 의도한 대로 생성되었는지 최종 확인합니다.

'하중재하도 Sheet'에서 지중하중, 토압, 온도하중 등이 올바른 위치에 다이어그램으로 표시되는지 검토합니다.

마지막으로 '단면력도(LCB1/LCB2/ENV1) Sheet'에서, 우리가 생성한 하중 조합에 대한 휨모멘트도, 전단력도 등 결과 다이어그램을 확인하여 모델링이 성공적으로 완료되었음을 검토합니다.

여기까지가 Excel VBA를 이용한 RC 라멘교 자동 모델링 실습이었습니다. 다음 파트에서는 좀 더 복잡한 FCM 교량 실습으로 넘어가겠습니다.

Part 6

실습 목표: FCM 교량의 시공단계해석

CIVIL NX 시공 단계 기능을 활용한 FCM 교량 모델링

모델링

- 단위계, 구조물의 형식, 단면, 재질 정의

경계조건

- FCM 시공 단계를 고려한 경계 조건 설정

시공 단계 설정을 위한 그룹

- Structure Group, Boundary Group, Load Group 정의

하중조건

- 자중, 가설차, 젖은 콘크리트 등 FCM 시공 단계별 하중 설정

Tendon 설정

- Tendon의 재질, 프로파일, 프리스트레스 설정

시간의존적 재료 특성 정의 및 연결

- Creep, Shrinkage, 재료 연결 설정

구조해석

- 시공단계 및 시간 시간의존적 재료 특성 고려

결과 분석

- FCM 교량의 시공 단계를 고려한 결과 분석



이어서 두 번째 실습으로 넘어가겠습니다.

이번 실습 목표는 '**FCM 교량의 시공단계해석**' 모델링 입니다.

FCM, 즉 Free Cantilever Method 교량은 보시는 바와 같이 세그먼트를 하나씩 붙여나가는 방식이라 시공 단계가 매우 복잡합니다.

Wizard를 사용하지 않고, FCM 공법을 사용한 교량 모델링은 다음과 같습니다

모델링, 경계조건 설정, 시공단계 설정을 위한 그룹 생성, 하중조건 설정, Tendon 설정, 시간의존적 재료 특성 정의 및 연결, 구조해석

복잡하고 반복적인 구조해석에서 휴먼에러가 자주 발생합니다.

이러한 휴먼 에러를 최소화 하기 위하여 VBA 기반 API 를 활용하게 되면, 실수를 줄이고, 빠른 시간내에 모델링이 가능해집니다.

Part 6

실습 목표: FCM 교량의 시공단계해석

Step 1. 단위계(UNIT) 설정

Excel 시트 입력

- [Unit] Sheet에서 API 설정 값 입력(BaseUrl, MapiKey)
- Unit Setting Table에 Force, Length, Heat, Temperature Unit 입력
- 데이터 유효성 검사를 통하여 Drop Down 필드 활용
- [Midas_ApplyUnitSystem] Macro Button Click!

VBA 로직

- 데이터 읽기: Excel 시트의 Unit Setting Table에서 'Value' 값을 변수로 읽어옴
- JSON 생성: 읽어온 변수들을 API 매뉴얼에 정의된 Unit(/db/UNIT) JSON 형식에 맞춰 텍스트로 조합
- API 요청: /db/UNIT 엔드포인트로 POST 메서드를 사용하여 생성된 JSON 데이터를 전송

API 설정				
BaseUrl	https://moa-engineers.midasit.com:443/civil	API 서버 주소		
MapiKey	eyJ1cil6lktSMjUwN09DUVoilCJwZyl6lmNpd	API 인증 키	Midas_ApplyUnitSystem	Click!
Key	Value	MEMO		
FORCE	N	Force(mass)		
DIST	M	Length		
HEAT	J	Heat		
TEMPER	C	Temperature Unit		
	C			
	F			

Drop Down Field

FORCE	DIST	HEAT	TEMPER
N	M	CAL	C
KN	CM	KCAL	F
KGF	MM	J	
TONF	FT	KJ	
LBF	IN	BTU	
KIPS			

유효성 조건

자, 그럼 FCM 교량 실습의 **단위계(UNIT) 설정**부터 시작하겠습니다.

먼저, 엑셀의 [Unit] 시트를 엽니다.

가장 먼저 상단의 'API 설정' 부분에 앞서 RC 라멘교 실습 때와 동일하게 본인의 **BaseUrl**과 **MapiKey**를 입력합니다.

그다음, 아래 'Unit Setting Table'에 이 모델에서 사용할 단위계를 입력합니다. Force(힘), Dist(길이), Heat(열), Temper(온도) 단위를 각각 입력합니다.

이 엑셀 시트는 편의를 위해 데이터 유효성 검사가 적용되어 있어, 'Value' 열의 셀을 클릭하면 우측 표에 있는 값들을 **드롭다운**으로 쉽게 선택할 수 있습니다.

입력이 완료되면, 우측 상단의 **[Midas_ApplyUnitSystem]** 매크로 버튼을 클릭합니다.

이 버튼을 누르면 VBA 로직이 실행되어, 엑셀 시트의 'Value' 값들 (N, M, J, C 등)을 읽어옵니다.

이 값들을 API 매뉴얼에 정의된 단위계(/db/UNIT) JSON 형식에 맞춰 텍스트로 조합합니다.

마지막으로, 이 JSON 데이터를 /db/UNIT 엔드포인트로 **POST** 메서드를 사용해 CIVIL NX로 전송하여 단위계 설정을 완료합니다.

Part 6

실습 목표: FCM 교량의 시공단계해석

Step 2. 재료(Material) 설정

Excel 시트 입력

- [Material] Sheet에서 API 설정 값 입력(BaseUrl, MapiKey)
- 상부구조(C40), 하부구조(C27), Tendon의 각 자료에 맞는 Property 입력
- [Midas_ApplyMaterials] Macro Button Click!

VBA 로직

- 데이터 읽기: Excel 시트의 Material Property Table에서 각각의 'Value' 값을 변수로 읽어옴
- JSON 생성: 읽어온 변수들을 API 매뉴얼에 정의된 Material(/db/MATL) JSON 형식에 맞춰 텍스트로 조합
- API 요청: /db/MATL 엔드포인트로 POST 메서드를 사용하여 생성된 JSON 데이터를 전송

API 설정

BaseUrl

https://moa-engineers.midasit.com:443/civil

API 서버 주소

MapiKey

eyJ1cii6iktSMjUwN09DUVoLCJwZyI6ImNpdmlsiwiY2

API 인증 키

Midas_ApplyMaterials

Click!

입력 값

ID	NAME	TYPE	P. TYPE	STANDARD	DB	CODE	ELAST	POISN	THERMAL	DEN	MASS	DAMP_RAT	HE_SPEC	HE_COND	bMASS_DENS
1	C40	CONC		1 KS01-Civil(RC)	C40										FALSE
2	C27	CONC		1 KS01-Civil(RC)	C27										FALSE
3	Tendon	USER	2				2.00E+08	0.3		78.5					TRUE

Click!

입력 값

단위계 설정이 완료되었으면, **Step 2. 재료(Material) 설정**으로 넘어갑니다.

[Material] 시트를 엽니다. 이 시트에도 상단에 'API 설정'이 있습니다. MapiKey가 올바르게 입력되었는지 확인합니다.

아래 'Material Property Table'에 우리가 사용할 재료 속성을 입력합니다. 이 예제에서는 3가지 재료를 정의합니다.

ID 1번: 상부구조용 'C40' 콘크리트 , ID 2번: 하부구조용 'C27' 콘크리트 , ID 3번: 'Tendon' (Type: USER)

각 재료의 Type, Standard, Code, DB 등 테이블에 있는 속성 값들을 입력합니다.

입력이 완료되면, **[Midas_ApplyMaterials]** 매크로 버튼을 클릭합니다.

VBA는 이 테이블에 입력된 'Value' 값들을 읽어오고, 이 값들을 API 매뉴얼에 정의된 재료 (/db/MATL) JSON 형식에 맞춰 텍스트로 조합합니다.

마지막으로, 이 JSON 데이터를 /db/MATL 엔드포인트로 **POST** 메서드를 사용해 CIVIL NX로 전송합니다.

Part 6

실습 목표: FCM 교량의 시공단계해석

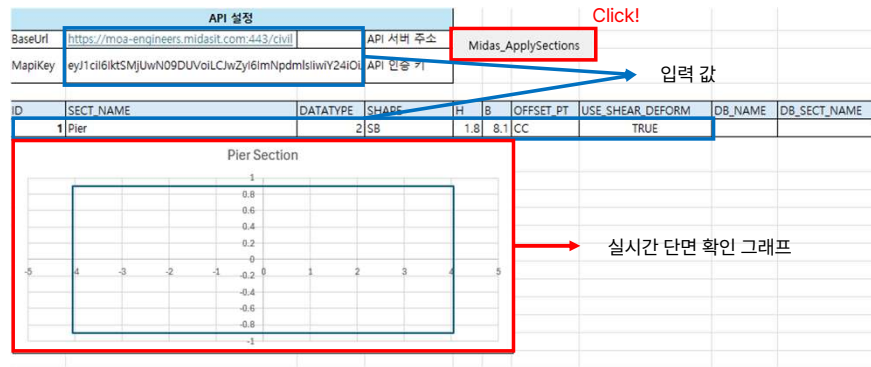
Step 3. 단면(Section) 설정

Excel 시트 입력

- [Section] Sheet에서 API 설정 값 입력(BaseUrl, MapiKey)
- 교각(Pier) Solid Rectangle 단면 정보 입력
- [Midas_ApplySections] Macro Button Click!

VBA 로직

- 데이터 읽기: Excel 시트의 Section Property Table에서 각각의 'Value' 값을 변수로 읽어옴
- JSON 생성: 읽어온 변수들을 API 매뉴얼에 정의된 Section(/db/SECT) JSON 형식에 맞춰 텍스트로 조합
- API 요청: /db/ SECT 엔드포인트로 POST 메서드를 사용하여 생성된 JSON 데이터를 전송



재료 입력이 끝났으면, **Step 3. 단면(Section) 설정**입니다.

[Section] 시트를 엽니다. 상단의 API 설정(MapiKey)을 다시 한번 확인합니다.

이 실습에서는 교각(Pier)의 단면을 Solid Rectangle(SB) 타입으로 정의합니다.

테이블에 단면 이름('Pier'), 단면 형상('SB'), 단면 제원(H: 1.8, B: 8.1), 기준점('CC') 등 속성 값을 입력합니다.

이 엑셀 시트에는 입력된 제원을 바탕으로 단면 형상을 실시간으로 보여주는 그래프가 포함되어 있어, 입력값을 시각적으로 바로 검토할 수 있습니다.

입력이 완료되면, **[Midas_ApplySections]** 매크로 버튼을 클릭합니다.

VBA는 이 Section Property Table에서 입력된 'Value' 값들을 읽어옵니다.

이 값들을 API 매뉴얼에 정의된 단면(/db/SECT) JSON 형식에 맞춰 텍스트로 조합합니다.

마지막으로, 이 JSON 데이터를 /db/SECT 엔드포인트로 **POST** 메서드를 사용해 CIVIL NX로 전송합니다.

Part 6

실습 목표: FCM 교량의 시공단계해석

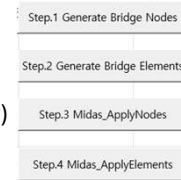
Step 5. 절점(Node), 요소(Element) 설정

Excel 시트 입력

- [FCM_Input] Sheet에서 API 설정 값 입력(BaseUrl, MapiKey)
- FCM 교량 설계의 설계 값 입력 (FCM, FSM 구간의 상부구조, 하부구조, 세그먼트 수 등)

VBA 로직

- 데이터 읽기: Excel 시트의 Design Table에서 각각의 'Value' 값을 변수로 읽어옴
- 1차적으로 Excel에서 노드의 위치를 결정하기 위한 변수 값 계산 및 Excel[Node_Output], [Element_Output] Sheet에 Node, Element 생성
- JSON 생성: 읽어온 변수들을 API 매뉴얼에 정의된 Node(/db/NODE), Element(/db/ELEM) JSON 형식에 맞춰 텍스트로 조합
- API 요청: Node(/db/NODE), Element(/db/ELEM) 엔드포인트로 POST 메서드를 사용하여 생성된 JSON 데이터를 전송

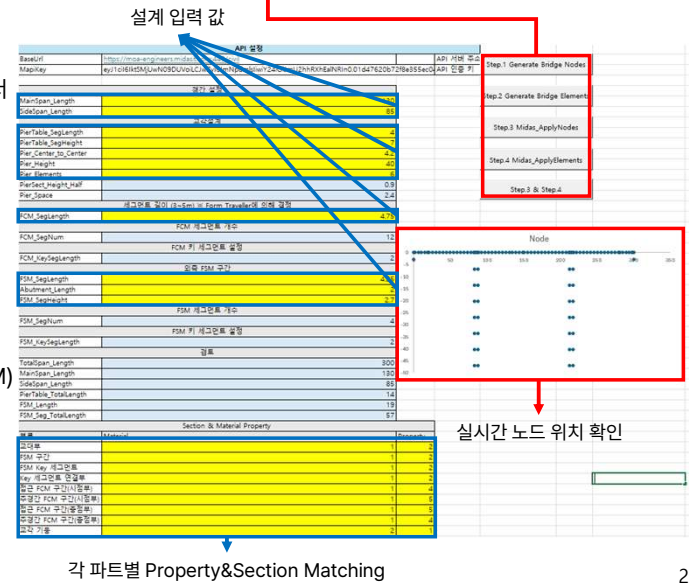


설계 값 기반 Node위치 설정 및 Node_Output 시트에 Node 생성

설계 값 및 Node 기반 Element설정 및 Element_Output 시트에 Element 생성

API를 통한 CIVIL NX로 Node 데이터 전송 Endpoint (/db/NODE)

API를 통한 CIVIL NX로 Element 데이터 전송 Endpoint (/db/ELEM)



각 파트별 Property&Section Matching

23

자, 이제 **Step 5. 절점(Node)과 요소(Element) 설정**입니다. 모델의 뼈대를 만드는 가장 핵심적인 부분입니다.

[FCM_Input] 시트를 엽니다. 이 시트에 FCM 교량의 핵심 설계 변수들을 모두 입력합니다.

예를 들어, 'MainSpan_Length'(주경간장), 'SideSpan_Length'(측경간장), 'Pier_Height'(교각 높이), 'FCM_SegNum'(FCM 세그먼트 개수) 등 노란색으로 표시된 '설계 입력 값' 셀에 수치를 입력합니다.

VBA 매크로 실행 부분의 VBA 로직은 여러 단계로 나뉘어 있습니다. 우측의 버튼들을 봐주십시오.

Generate Bridge Nodes 버튼은 엑셀 '내부에서' 설계 값(경간장, 높이 등)을 기반으로 모든 노드의 좌표를 계산하고, 그 결과를 [Node_Output]이라는 별도 시트에 생성합니다.

여기서 우측의 차트를 보면, 입력된 값을 기반으로 생성될 노드의 위치를 실시간으로 시각적으로 확인할 수 있도록 구성해놨습니다.

Generate Bridge Elements 버튼은 엑셀 '내부에서', Step.1에서 생성된 노드들을 연결하여 요소(Element) 정보를 만들고, [Element_Output] 시트에 생성합니다.

Midas_ApplyNodes 버튼은 [Node_Output] 시트에 준비된 노드 데이터들을 API를 통해 CIVIL NX로 실제 '전송'합니다. (엔드포인트: /db/NODE)

Midas_ApplyElements 버튼은 [Element_Output] 시트에 준비된 요소 데이터들을 API를 통해 CIVIL NX로 '전송'합니다. (엔드포인트: /db/ELEM)

이렇게 엑셀 내부 계산과 API 전송 단계가 분리되어 있습니다. Step.3 & Step.4 버튼은 이 두 전송 작업을 한 번에 실행합니다.

매크로를 수정하시면 지금 설명해드렸던 모든 프로세스들을 버튼 하나로도 실행하실 수 있습니다.

Part 6

실습 목표: FCM 교량의 시공단계해석

Step 6. 그룹 (Struture/Boundary/Load Group) 설정

Excel 시트 입력

- [Group] Sheet에서 API 설정 값 입력(BaseUrl, MapiKey)
- 그룹(Group) Structure/Boundary/Load Group 정보 입력
- [Midas_ApplyGroups] Macro Button Click!

VBA 로직

- 데이터 읽기: Excel 시트의 Section Property Table에서 각각의 'Value' 값을 변수로 읽어옴
- JSON 생성: 읽어온 변수들을 API 매뉴얼에 정의된 Structure Group(/db/GRUP), Boundary Group(/db/BNGR), Load Group(/db/LDGR) JSON 형식에 맞춰 텍스트로 조합
- API 요청: 각각의 엔드포인트로 POST 메서드를 사용하여 생성된 JSON 데이터를 전송

입력 값

API 설정				
BaseUrl	https://moa-engineers.midas.com:443/civil		API 서버 주소	Click! Midas_ApplyGroups
MapiKey	ey/1ci6ikt5MjUwN09DUVoLCjwZyl6ImNpdmlslwiY24iOiJmU		API 인증 키	
StructureGroupName	Element	BoundaryGroupName	Autotype	LoadGroupName
Pier1	83, 87, 91, 95, 99, 103, 84, 88, 92	S-Pier	Auto	Self
Pier2	85, 89, 93, 97, 101, 105, 86, 90, 94	S-FsmLeft	Auto	PS-PierTable
PierTable1	21, 22, 23, 24, 25, 26, 27, 28	S-FsmRight	Auto	PS-P1Seg
PierTable2	62, 63, 64, 65, 66, 67, 68, 69	EL-Pier	Auto	PS-P2Seg
P1Seg1	20, 29	EL-FsmLeft	Auto	PS-KeySeg
P1Seg2	19, 30	EL-FsmRight	Auto	
P1Seg3	18, 31			
P1Seg4	17, 32			

절점과 요소 생성이 완료되었으니, 이제 시공단계해석에 필수적인 **Step 6. 그룹 (Group) 설정**입니다.

[Group] 시트를 엽니다. 상단의 API 설정(MapiKey)을 확인합니다.

이 테이블에 시공 단계를 정의하기 위한 **Structure Group, Boundary Group, Load Group** 정보를 모두 입력합니다.

예를 들어, Structure Group 'Pier1'에는 요소 '83, 87, 91...' 등이 포함된다고 정의하고, Boundary Group 'S-Pier' 등도 함께 정의합니다.

모든 그룹 정보 입력이 완료되면, **[Midas_ApplyGroups]** 매크로 버튼을 클릭합니다.

VBA는 이 테이블에 입력된 모든 'Value' 값들을 읽어옵니다.

어온 값들을 API 매뉴얼에 정의된 각 그룹별 JSON 형식으로 조합합니다.

Structure Group은 /db/GRUP 엔드포인트로,

Boundary Group은 /db/BNGR 엔드포인트로,

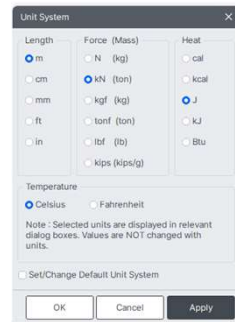
Load Group은 /db/LDGR 엔드포인트로 각각 **POST** 메서드를 사용해 생성된 JSON 데이터를 전송합니다.

Part 6

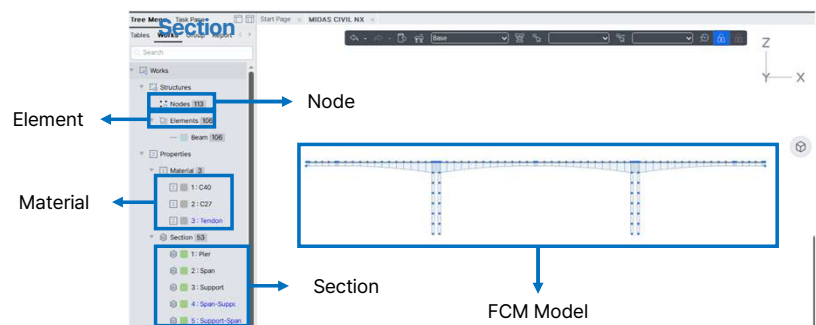
실습 목표:
FCM 교량의
시공단계해석

Step 7. 모델 검토

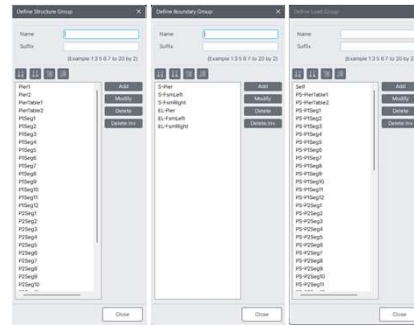
Unit System



Node, Element, Material,



Structure / Boundary / Load Group



* PSC 단면은 Postman을 통해 별도 입력됨

자, Step 6에서 그룹 생성까지 모두 API로 전송했습니다.

FCM 교량 모델링의 마지막 단계는 **Step 7. 모델 검토**입니다.

엑셀에서 모든 매크로를 실행한 뒤, 이제 CIVIL NX 화면으로 돌아와서 모델이 엑셀에서 입력한 대로 잘 생성되었는지 최종 검토합니다.

먼저 단위계가 우리가 [Unit] 시트에서 설정한 값 (예: kN, m, J, Celsius)으로 올바르게 적용되었는지 확인합니다.

다음으로, 화면 왼쪽의 Works Tree를 확인합니다.

Node / Element: [FCM_Input] 시트에서 계산된 개수만큼 절점과 요소가 생성되었는지, 그리고 모델 뷰에 형상이 올바르게 생성되었는지 확인합니다.

Material / Section: [Material]과 [Section] 시트에서 입력한 재료(C40, C27, Tendon)와 단면(Pier)이 정확히 생성되었는지 확인합니다.

Groups: 마지막으로, 시공단계해석에 가장 중요한 그룹들을 확인합니다.

Structure Group, Boundary Group, Load Group 각각의 대화상자를 열어, [Group] 시트에서 입력했던 그룹 리스트(Pier1, P1Seg1, S-Pier 등)가 빠짐없이 모두 정의되었는지 검토합니다.

여기까지 FCM 교량 실습을 마치겠습니다. 다음은 마지막 실습입니다.

Part 7

실습 목표: 지하철 정거장의 해석 및 설계

Step 1. Rahmen/Culvert Auto Generation Wizard

Rahmen/Culvert Auto Generation Wizard 실행

- Structure 탭 → RC Structure
→ Rahmen/Culvert Auto Generation 실행

모델

- 구조물 형식(암거_복층), 형상, 현치부, 재질, 두께 입력

경계조건

- 지진시 검토에 대한 고려 [✓], 구조물 폭, 지반 탄성계수 λ(지진시), 지반 특성 입력

하중

- 하중 조합 자동 생성, 상시하중, 지진시하중, 지진구역, 지반분류
성능별 감쇠율(%) 설정

해석/보고서

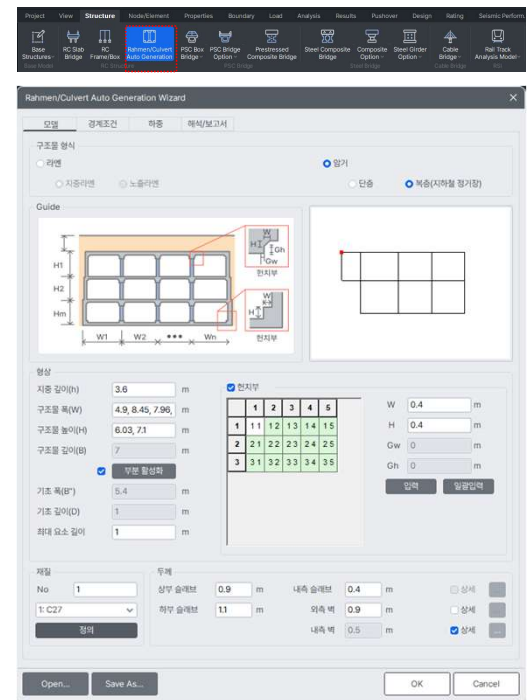
- 보고서 출력(제목, 설명, 출력 그림, 단면력도) 설정

Wizard 실행

- OK 버튼 클릭 후 해석 수행

결과 검토 (지하철 정거장.xlsx)

- 설계조건, 단면가정, 부력검토, 모델링, 하중산출근거 등등



26

자, 이제 마지막 세 번째 실습입니다. 이번 실습 목표는 '지하철 정거장의 해석 및 설계'입니다.

이 실습의 핵심은, 매우 강력한 CIVIL NX의 내장 기능인 'Wizard'와 우리가 배운 'API'를 연계하여 활용하는 것입니다.

먼저 Wizard를 사용해 기본 모델을 생성하겠습니다.

CIVIL NX 상단 메뉴의 **[Structure] 탭**으로 이동합니다.

[RC Structure] 그룹에서 **[Rahmen/Culvert Auto Generation]** Wizard 아이콘을 실행합니다.

Wizard 창이 뜨면, 각 탭에서 기본 설정을 입력합니다.

[모델] 탭에서 '복층(지하철 정거장)' 형식을 선택하고, 지간, 높이, 현치, 재질, 두께 등 구조물 형상 정보를 입력합니다.

[경계조건] 탭에서 '지진시 검토'를 체크하고, 구조물 폭이나 지반 특성, 지반 탄성계수등을 입력합니다.

[하중] 탭에서 '하중 조합 자동 생성'을 체크하고, 토압, 지하수, 포장 하중 등 각종 상시하중과 지진시하중 등을 설정을 입력합니다.

[해석/보고서] 탭: 보고서 제목이나 출력 그림 옵션 등 마지막 설정을 완료합니다.

모든 입력이 끝나면 [OK] 버튼을 클릭합니다.

그러면 Wizard가 이 모든 정보를 바탕으로 모델링, 하중 재하, 해석까지 자동으로 수행하고, 지하철 정거장.xlsx라는 엑셀 리포트 파일까지 생성해 줍니다.

이렇게 CIVIL NX의 Wizard를 사용하시면, 위자드가 지원되는 다양한 구조물에 대하여 보다 쉽고 빠르게 모델링과 해석 그리고 리포트 파일 까지 하나의 프로세스로 해결 할 수 있습니다.

Part 7

실습 목표: 지하철 정거장의 해석 및 설계

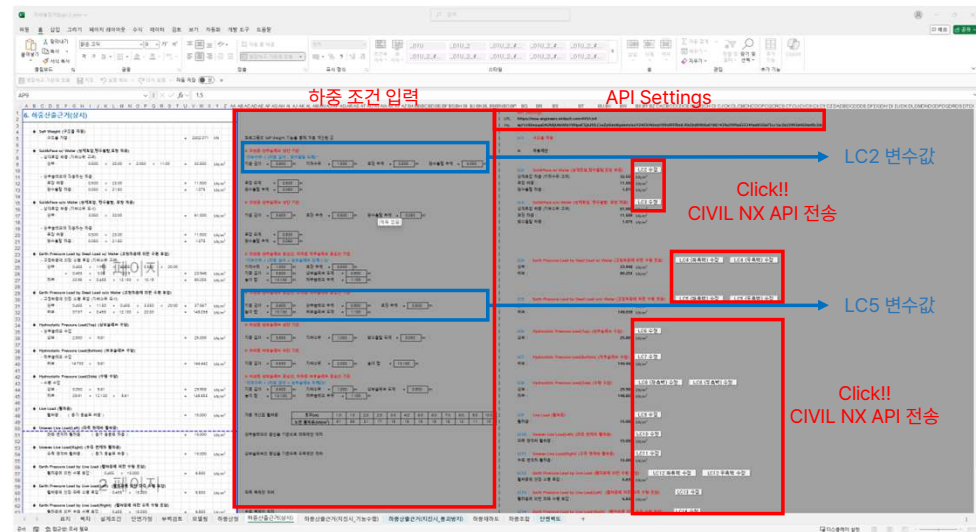
Step 2. VBA(API)를 통한 하중 조건 변경

지하철정거장api.xlsm 실행

- Wizard로 생성된 (지하철 정거장.xlsx) 동일 구조

하중 조건 변경 - 하중산출근거(상시) sheet

- 예1) LC2 ▶ 상재도압, 방수몰탈, 포장 하중 변경
- 예2) LC5 ▶ 지중깊이, 상부슬래브 두께, 포장 두께, 높이 합, 하부 슬래브 두께 변경



하지만 실무에서는 특정 조건, 예를 들어 '상재도압'이나 '포장 두께'만 살짝 바뀌가며 민감도를 검토하거나 갑작스러운 설계변경으로 다시 구조 검토를 해야할 때가 많습니다.

이때 Wizard를 매번 다시 실행하는 것은 매우 비효율적일수 있습니다.

이때 바로 API가 활약합니다.

Wizard가 생성한 파일이 아닌, **지하철정거장api.xlsm** 파일을 열겠습니다. 이 파일에는 기존 구조 계산서내에 하중 재계산 로직과 API 매크로가 심어져 있습니다.

파일을 열고 '**하중산출근거(상시)**' 시트로 이동합니다. 여기서 Wizard가 초기에 계산했던 변수 값들을 직접 수정합니다. 예를 들어,

LC2의 '상재도압', '방수몰탈', '포장' 하중 값을 변경합니다.

LC5의 '지중깊이', '상부슬래브 두께' 등 하중 산출 근거가 되는 변수 값을 변경합니다.

값이 수정되었으면, 우측의 **[CIVIL NX API 전송]** 버튼을 클릭합니다.

이 버튼을 누르면 VBA 매크로가 실행되어, 엑셀 시트에서 변경된 값들을 읽어옵니다.

이 값들을 이용해 하중 데이터를 수정하는 JSON을 생성하고, API를 통해 현재 열려있는 CIVIL NX 모델에 **PUT**으로 전송하여 하중 조건을 덮어쓰기(수정)합니다.

이렇게 하면 모델 전체를 다시 생성할 필요 없이, 원하는 하중 조건만 신속하게 변경할 수 있습니다.

Part 7

실습 목표: 지하철 정거장의 해석 및 설계

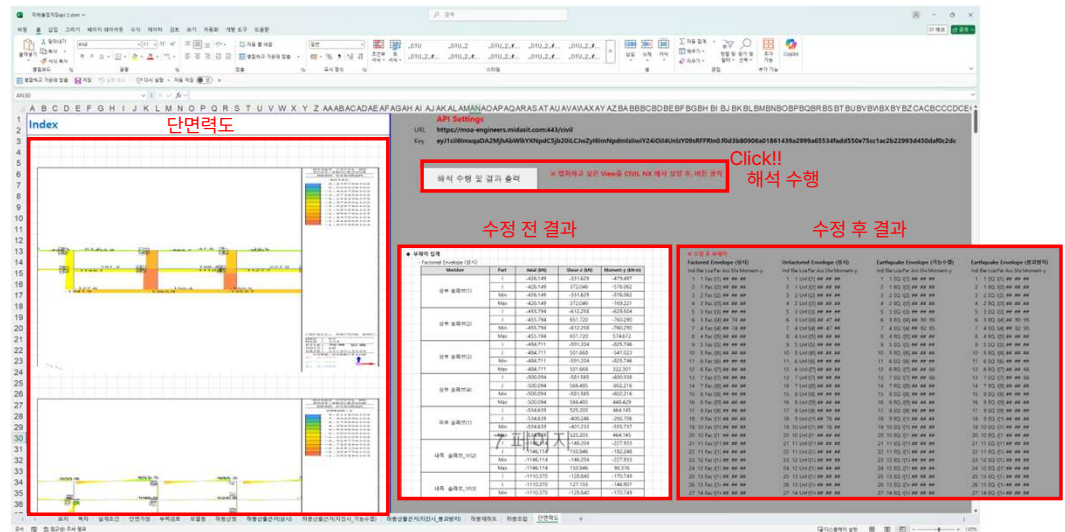
Step 3. 재해석 및 결과 검토

해석 수행 및 결과 출력

- '해석 수행 및 결과 출력' 버튼 클릭 - 단면력도 Sheet

새로운 결과 검토

- 예1) LC2 ▶ 상태도압, 방수몰탈, 포장 하중 변경
- 예2) LC5 ▶ 지중깊이, 상부슬래브 두께, 포장 두께, 높이 합, 하부 슬래브 두께 변경



하중이 바뀌었으니, 당연히 재해석을 수행하고 결과를 다시 검토해야 합니다.

해석 수행 및 결과 출력:

단면력도 시트에서 '해석 수행 및 결과 출력' 버튼을 클릭합니다.

이 버튼은 VBA 매크로를 실행시켜, CIVIL NX에 '해석을 수행하라'는 명령을 API로 전송합니다.

매크로는 해석이 완료될 때까지 기다린 다음, 완료되면 다시 API를 통해 변경된 해석 결과(단면력 등)를 CIVIL NX로부터 가져와 엑셀 시트에 업데이트합니다.

이제 엑셀 시트에서 변경된 결과를 확인합니다.

슬라이드 하단에 '수정 전 결과'와 '수정 후 결과'가 비교되어 있듯이, API를 통해 하중 조건을 변경한 후 재해석한 결과가 엑셀 시트에 새롭게 반영된 것을 확인할 수 있습니다.

이처럼 Wizard로 기본 모델을 빠르고 편리하게 생성한 뒤, 세부적인 파라미터 변경 및 반복 검토는 API를 연계하여 자동화함으로써 매우 효율적인 실무 작업이 가능해집니다.

여기까지가 마지막 실습이었습니다.

요약

Today's Key Takeaway

- API는 '약속된 규칙': 외부 프로그램이 CIVIL NX와 소통하는 표준 방식임
- 3대 핵심 요소: URI(주소), HTTP Method(방식), JSON(내용)의 조합으로 통신함
- 무한한 확장성: Excel VBA뿐만 아니라 Python 등 다양한 언어로 나만의 자동화 툴을 개발 가능함

자동화를 통한 설계 패러다임의 전환

- 단순 반복 업무 → 자동화
- 엔지니어의 시간 → 고부가가치 창출(최적화, 대안 설계, 핵심 기술 검토)

네, 이상으로 모든 실습을 마쳤습니다. 오늘 교육의 핵심 내용을 마지막으로 요약해 드리겠습니다.

Today's Key Takeaway(오늘의 핵심 요약)

오늘 이것만은 꼭 기억해 주셨으면 합니다.

API는 '약속된 규칙'입니다. 외부 프로그램(Excel, Python)이 CIVIL NX와 소통하기 위해 만들어진 표준 방식입니다.

3대 핵심 요소로 통신합니다. 모든 API 통신은 **URI(주소)**, **HTTP Method(방식)**, 그리고 **JSON(내용)** 이 세 가지의 조합으로 이루어집니다.

API는 '무한한 확장성'을 가집니다. 오늘 배운 Excel VBA뿐만 아니라 Python 등 다양한 언어로 여러분만의 자동화 툴을 개발할 수 있습니다.

자동화를 통한 설계 패러다임의 전환

저희가 API를 통해 자동화를 추구하는 이유는 명확합니다.

기존의 단순 반복 업무는 **자동화**에 맡기고,

엔지니어인 우리는 그렇게 확보한 시간을 **고부가가치 창출** (예: 설계 최적화, 다양한 대안 설계 검토, 핵심 기술 검토)에 사용하는 것,

이것이 바로 API를 통한 '**설계 패러다임의 전환**'입니다.

Thank You!

© MIDAS IT Co., Ltd

경청해 주셔서 감사합니다. (질문 있으시면 받겠습니다.)