

Service API Integration Plug-in Guide

개요

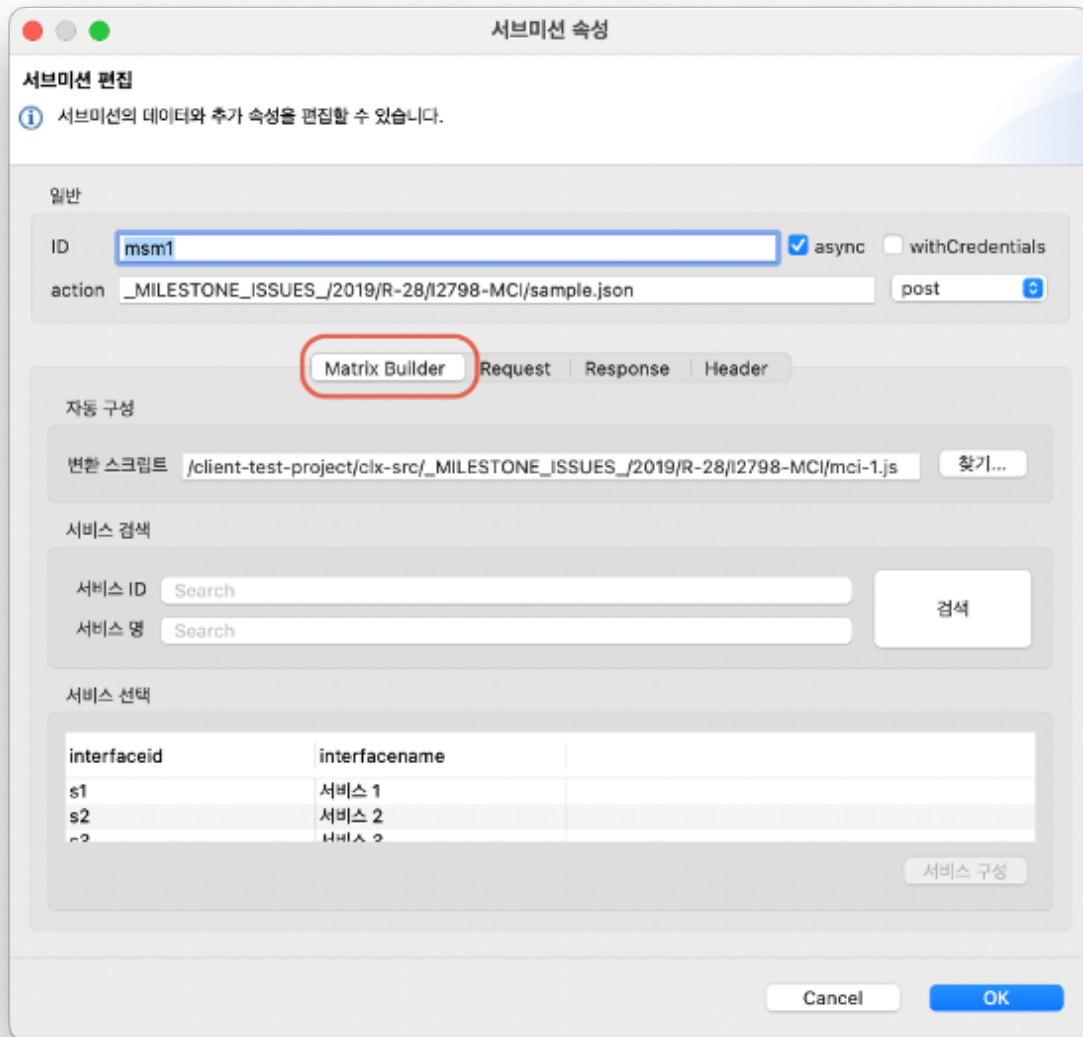
서비스API 통합(Service API Integration) 플러그인은 통상 **MCI 플러그인**이라는 코드명으로 언급되던 이클립스 플러그인입니다.

이 플러그인은 개발자들이 MCI 등과 같은 API 서버를 통해 서비스 API를 검색하고, 해당 서비스를 사용하는 **서브미션** 및 **매트릭스 서브미션** 그리고 그와 연관된 데이터 컨트롤들을 자동으로 구성하거나 생성할 수 있는 도구들을 제공합니다. **eXBuilder6**가 설치된 이클립스 환경에 추가로 설치 하여 사용 할 수 있습니다.

이 플러그인은 **매트릭스 빌더**와 **서브미션 빌더**로 구성 됩니다. API 서버의 정의에 따라 각각 매트릭스 서브미션 또는 일반 서브미션을 자동 구성 합니다. 고객사가 체계화된 서비스 API 명세를 검색할 수 있는 API를 갖춘 시스템을 가지고 있고, 이미 구축된 서비스를 eXBuilder6와 연동하고자 하는 경우 사용할 수 있습니다.

매트릭스 빌더

매트릭스 빌더는 매트릭스 서브미션 편집 다이얼로그의 첫 번째 탭에서 찾을 수 있습니다.



매트릭스 빌더는 최종 사용자가 UI를 통해 서비스를 검색할 수 있도록 하며, 선택된 서비스 명세를 바탕으로 매트릭스 서브미션을 자동으로 구성할 수 있도록 돕는 도구입니다.

변환 스크립트

변환 스크립트는 API 서버와 eXBuilder6 사이의 중재자 역할을 수행하는 스크립트입니다. 매트릭스 빌더가 작동하기 위해서는 먼저 변환 스크립트가 작성되고 지정되어야 합니다.



변환 스크립트는 반드시 다음 함수들을 최상위 범주(Scope)에 정의해야 하며, 주어진 입출력 조건을 준수하여야 합니다.

표1 변환 스크립트 함수 목록

함수	입력	출력
getSearchFields()	없음	서비스 검색 필드 정의
getLookupInfo()	검색 쿼리 객체	검색 Rest API 호출 방법
convertSearchResult()	API 서버의 서비스 검색 응답	서비스 목록 및 목록 표시 방법
getServiceURL()	사용자가 선택한 서비스	해당 서비스의 Rest API 호출 방법
convert()	API 서버가 반환한 서비스 정의 응답	매트릭스 서브미션 정의

붉은색은 eXBuilder6에서 정의한 규격이며, 초록색은 API서버가 정한 규격을 나타냅니다. API서버의 규격은 고객사 마다 다를 수 있으므로 그에 맞추어 변환 스크립트가 작성되어야 합니다.

함수 호출 흐름

매트릭스 빌더는 다음과 같이 **변환 스크립트**와 연계하여 **매트릭스 서브미션**을 구성하게 됩니다:

1. 최종 사용자가 **매트릭스 서브미션** 다이얼로그를 열게 되면 최초 getSearchFields() 함수가 호출 됩니다. 이 함수의 반환 값에 따라 검색 유저 인터페이스가 만들어지게 됩니다.
2. 사용자가 검색을 수행하면 getLookupInfo()가 호출되고 이 함수는 사용자가 입력한 검색 조건을 바탕으로 API서버를 통해 검색하는 방법(예: URL 및 메소드, 파라미터 등)을 반환 합니다.
3. **매트릭스 빌더**는 그 결과를 바탕으로 API 서버에 질의 후 그 검색 결과를 convertSearchResult()에 전달합니다. 해당 함수는 API 응답 결과를 바탕으로 검색 결과를 표시하기 위한 정보를 반환해야 합니다.
4. 사용자가 서비스를 선택하면 getServiceURL() 함수로 전달 됩니다. 이 함수는 사용자가 선택한 서비스에 대한 세부사항을 얻기 위해 API 서버에게 질의 하는 방법을 반환합니다.
5. **매트릭스 빌더**는 해당 정보를 바탕으로 API 서버에 서비스의 세부사항을 질의하고 그 응답 결과를 convert() 함수에 전달 합니다. 이 함수는 **매트릭스 서브미션**을 구축하는 방법을 반환합니다. 이 결

과를 바탕으로 매트릭스 서브미션이 구성 됩니다.

getSearchField()

이 함수는 최종 사용자가 서비스를 검색하기 위한 조건을 입력할 수 있는 검색 필드들을 정의하는 JSON 을 반환 해야 합니다. 사용자 인터페이스에 표시할 레이블 텍스트를 키로 하고 사용자 입력 값을 저장할 멤버 변수명을 그 값으로 하는 JSON 형식입니다.

예시

```
{
  "카테고리": "category",
  "서비스 명": "serviceName"
}
```

위의 예시는 다음과 같은 UI를 구성하게 됩니다.

최종 사용자가 검색 필드 창을 채우고 검색 버튼을 누르면, getLookupInfo() 메소드가 호출됩니다.

getLookupInfo(query)

getLookupInfo(query) 함수는 최종 사용자가 입력한 검색어 조건들을 바탕으로 서비스 명세 서버에게 질의하는 방법을 나타내는 **API 호출 명세**를 반환하도록 구현해야 합니다.

사용자가 검색 버튼을 누르면 getLookupInfo(query) 함수가 호출됩니다. query 호출 인자는 getSerarchFields() 의 반환 값에 정의한 멤버 변수명들을 속성으로 가지고 있는 JSON 객체 입니다.

상술한 예시에서 사용자가 카테고리 필드에 tomato를 입력하고 서비스 명 필드에 system을 입력했다면, getLookupInfo(query)에 전달되는 query는 다음 값을 갖게 됩니다.

```
{
  "category": "tomato",
  "serviceName": "system"
}
```

getLookupInfo(query) 함수는 다음 예시와 같이 검색 서비스를 호출 하는 방법을 설명하는 **API 호출 명세** 객체를 반환해야 합니다:

```
{
  // 검색 API URL
  "url": "http://test.com/api/list-service",

  // 요청 메소드
  "method": "get",

  // 리퀘스트 바디 또는 파라미터로 전달할 데이터
  "request": {"cn" : "tomato", "sn" : "system"}
}
```

- url - 서비스를 검색할 요청을 보낼 API 서버 주소
- method - 요청을 보낼 때 사용할 메소드. "get" | "post" | "delete" | "put" 을 사용할 수 있습니다.
- request - 서버로 전달할 키 / 값을 가진 JSON을 정의합니다. method가 "post"또는 "put"인 경우에 JSON의 속성 값들은 다시 JSON객체 이거나 배열일 수 있지만 그 외의 경우 모두 문자열 이어야 합니다.

전송된 예시의 경우 **매트릭스 빌더**는 다음과 같은 URL을 이용하여 서비스 검색 호출 API를 사용합니다:

<http://test.com/api/list-service?cn=tomato&sn=system>

선택된 메소드가 "post" 등인 경우 요청 바디에 JSON이 그대로 적재되어 제출됩니다.

convertSearchResult()

convertSearchResult() 함수는 getLookupInfo() 결과로 얻은 API 호출 규격에 따라 API를 호출 하고 그 결과 응답 JSON을 바탕으로 최종사용자에게 검색 결과를 테이블 형태로 표시할 방법을 결정합니다. 이 함수는 다음과 같은 형태의 반환 값을 갖도록 작성되어야 합니다:

```
{
  "interfaces": [서비스1, 서비스2, 서비스3...],
  "columns": ["컬럼으로 표시할 키1", "키2"...],
}
```

만일 API 서버가 다음과 같은 검색 결과를 반환 했다면

```
{
  /* 중략 */
  "services":[
    {"name": "사원 조회", "id": "search_emp", /* 중략 */},
    {"name": "증명서 발급", "id": "pub_cert", /* 중략 */},
    /* 후략 */
  ]
}
```

다음과 같이 convertSearchResult() 함수를 구현할 수 있습니다.

```
function convertSearchResult(response) {
  return {
    // 서버 응답 부분 중 서비스 정보 배열 부분
    interfaces: response.services,

    // 검색 결과 테이블에 컬럼 형태로 노출 시킬 서비스 항목의 키
    columns: ["name", "id"]
  };
}
```

이 함수의 반환 결과에 따라 다음과 같은 검색 결과 화면이 구성됩니다.

서비스 선택

interfaceid	interfacename
s1	서비스 1
s2	서비스 2
s3	서비스 3
s4	한화 샘플 테스트 케이스

서비스 구성

getServiceURL(spec)

최종 사용자가 검색된 서비스 목록 중 하나를 더블 클릭 하거나 서비스 구성 버튼을 클릭하면 이 함수가 호출 됩니다. 사용자가 선택한 서비스 정보 객체가 인자로 전달되며, 선택한 서비스의 세부사항을 얻는 **API 호출 명세**를 반환해야 합니다. 이 규격은 getLookupInfo() 의 반환 값 형식과 동일합니다.

```
{
  "url": "API URL"
  "method": "메소드",
  "request": "페이로드/파라미터"
}
```

convert(service)

getServiceURL() 가 반환한 **API호출 명세**에 따라 API를 호출한 이후, 그 응답이 convert(service) 함수로 전달됩니다. 이 함수는 API 서버의 응답 값을 **매트릭스 빌더**가 이해할 수 있는 형태, **매트릭스_서브미션_규격**으로 변환 하여야 반환 해야 합니다. **매트릭스 빌더**가 수용하는 반환 값의 형태는 다음과 같습니다:

```
type 매트릭스_서브미션_규격정의 = {
  submission?: 매트릭스_서브미션_정의노드,
  request?: 매트릭스_정의노드,
  response?: 매트릭스_정의노드
}
```

각각의 모든 필드들은 생략될 수 있습니다, 각 노드의 유형은 후술됩니다.

서브미션 정의 노드

매트릭스 서브미션 정의 노드를 기술하면 **매트릭스 서브미션** 자체의 속성들을 지정 할 수 있습니다. 지원되는 필드들은 다음과 같으며, 각각의 필드들은 생략될 수 있습니다.

```
type SubmissionNode = {
  // 호출 URL
  action?: "URL",
  async?: true | false,
  withCredential?: true | false,
  method?: "post" | "get"
}
```

매트릭스 정의 노드

매트릭스 정의 노드(Matrix Definition Node, 이하 **MDN**)란 **매트릭스 서브미션**의 페이로드를 구성하는 각각의 노드를 처리하는 방법을 기술하는 JSON 객체를 말합니다. 각MDN은 다음과 같은 형식 중 하나 입니다.

컴플렉스 MDN

컴플렉스 타입 MDN은 JSON 및 Array 형태의 페이로드를 처리하는 방법을 기술하기 위한 노드 입니다. 형태는 다음과 같습니다:

```
{
  format: "object" | "array",
  dataControl?: "데이터 컨트롤 ID",
  dataControlType?: "dataset" | "datamap",
  linked: true | false,
  data: {... 하위 MDN ...}
}
```

컴플렉스 MDN은 JSON 또는 JSON 배열 형태의 응답 페이로드를 해석하거나 요청 페이로드를 합성하는 방법을 정의 합니다. format 필드 값이 “object” 인 경우, 해당 위치에 상응하는 페이로드가 JSON 객체임을 나타내고, “array” 인 경우 해당 위치에 상응하는 페이로드가 JSON 배열임을 나타냅니다.

“dataControl” 필드에 데이터 컨트롤의 ID를 기재하여 **서브미션**이 구성할 때, **데이터셋** 또는 **데이터맵**과 연결하도록 지시할 수 있습니다. 이 필드를 생략하면 해당 값을 가진 가장 가까운 상위 **MDN**의 정의를 승계합니다.

“dataControl” 필드 값으로 존재하지 않는 데이터 컨트롤 ID를 제공한 경우, 서브미션이 구성되는 동안 데이터 컨트롤이 함께 생성됩니다. format이 “object”인 경우 **데이터맵**이 “array”인 경우 **데이터셋**이 만들어 집니다. 생성할 데이터 컨트롤 타입을 별도로 명시하기 위해 “dataControlType” 필드를 추가로 지정할 수 있습니다. 가용 값은 “dataset” | “datamap” 입니다.

컴플렉스 MDN은 페이로드의 하위 속성 값들의 처리 방법을 기술하기 위해 data 필드를 정의할 수 있습니다. data 필드는 페이로드의 속성명을 키로 하고 해당 속성 값에 대한 처리 방법을 설명하는 하위 **MDN**을 값으로 하는 JSON형태로 정의 합니다.

다음의 예시는 {name: ..., age: ...} 형태가 반복되는 응답 JSON을 처리하는 **컴플렉스 MDN**의 예시입니다.

```
{
  format: "array",
  data: {
    name: 하위MDN,
    age: 하위MDN
  }
}
```

상위 **MDN**이 **데이터셋**과 연결되어 있고, 현재 **MDN**또한 **데이터셋**과 연결되어 있다면 두 데이터셋을 부모/자식 테이블 관계로 만들기 위해 linked 필드 값을 true로 기재할 수 있습니다. 이 경우 자식 **데이터셋**의 bindParentRow() API를 통해 부모 행을 변경할 수 있습니다.

단순 MDN

단순 타입 MDN은 단순 값(예: 문자열/숫자) 페이로드를 처리하는 방법을 기술하기 위한 노드입니다. 형태는 다음과 같습니다:

```
{
  format: "simple",
  data?: "고정 상수 값",
  path?: "@서브노드이름"
}
```

MDN의 format 필드 값이 “simple” | “decimal” | “string” | “number” 인 노드를 **단순 MDN**이라고 합니다. **단순 MDN**은 **컴플렉스 MDN**과 달리 하위 **MDN**을 갖지 않으며 페이로드의 특정 속성 값과 **데이터 컬럼**을 연결하기 위해 정의합니다.

페이로드의 특정 속성 값을 **데이터 컬럼**에 연결하거나(path 속성을 통해), 고정 상수 값을 공급(data 속성을 통해)하기 위한 용도로 사용됩니다. **단순 MDN**이 path 및 data를 모두 정의한 경우 data 필드는 무시 됩니다.

format 값으로 “simple”을 지정한 경우 “string” 과 동일한 의미를 갖습니다. **단순 MDN**에 의해 **데이터 컬럼**이 자동으로 만들어질 경우 format 값에 따라 **데이터 컬럼**의 타입이 결정 됩니다.

다음의 예시는 JSON 페이로드의 foo 속성의 값을 “price” 라는 이름의 데이터 컬럼에 연결하도록 지시하는 합니다. path 값은 @컬럼이름 형태로 기술 하십시오.

```
{
  format: "object",
  // ... 중략 ...
  data: {
    foo: {
      format: "number",
      path: "@price"
    }
  }
}
```

확장 필드

모든 **MDN** 및 **서브미션 정의 노드**는 페이로드의 처리 방법과는 관련이 없으나 추가적인 응용을 위해 확장 필드들을 가질 수 있습니다.

comment

MDN노드에 comment: string 필드 값이 존재하는 경우 연결될 데이터 컨트롤 또는 데이터 컬럼에 주석이 추가 됩니다.

```
foo: {
  format: "number",
  path: "@price",
  comment: "가격"
}
```

서브미션 정의 노드에 comment: string 필드 값이 존재하는 경우, 해당 필드 값이 **서브미션**의 주석으로 추가 됩니다.

userAttr

userAttr필드 값은 모든 속성 값이 문자열인 JSON 형태로 제공해야 합니다.

서브미션 정의 노드가 userAttr 필드를 소유한 경우 **서브미션**에 해당하는 **사용자 정의 속성**들이 추가 됩니다.

```
{
  submission: {
    userAttr: {"키1": "값1", "키2": "값2"}
  },
  request: {},
}
```

```
response: {}
}
```

노드 편집 잠금

개발자가 매트릭스 서브미션의 노드 구조를 수동으로 편집하는 것을 방지하기 위해 **매트릭스_서브미션_규격정의**에 다음과 같이 "--nodeLock" 속성을 지정할 수 있습니다.

```
type 매트릭스_서브미션_규격정의 = {
  submission?: 매트릭스_서브미션_정의노드,
  request?: 매트릭스_정의노드,
  response?: 매트릭스_정의노드,

  /** 개발자의 매트릭스 노드 편집을 방지할 것인지 여부 */
  "--nodeLock"? : boolean
}
```

노드 편집 잠금 기능을 사용하는 경우 개발자들이 해당 노드를 편집할 수 없는 이유를 파악하기 어렵기 때문에 충분한 사전 교육이 이루어진 상태에서 사용하십시오. 잠금을 해제해야 할 필요가 있을 경우 CLX 소스 탭에에서 매트릭스 서브미션 정의부 내부에서 다음과 같은 메타데이터 정의부를 찾아 제거 하십시오.

```
<std:metadata>
  <!-- 노드 편집 잠금 스튜디오 메타데이터 -->
  <std:property key="--nodeLock" value="true"/>
</std:metadata>
```

서브미션 빌더

서브미션 빌더는 매트릭스 빌더와 마찬가지로 API 서버의 응답을 기반으로 데이터 컨트롤 및 서브미션을 자동으로 구성하여 줍니다. 매트릭스 빌더와 달리 공급되는 서비스들이 **eXBuilder6** 프로토콜을 지원하는 경우에 사용 할 수 있습니다.

매트릭스 빌더와 마찬가지로 변환스크립트를 통한 동일한 단계를 거쳐 작동 합니다.

변환 스크립트

서브미션의 변환 스크립트는 매트릭스 변환 스크립트와 완전히 동일한 구성을 갖습니다.

함수	입력	출력
<code>getSearchFields()</code>	없음	서비스 검색 필드 정의
<code>getLookupInfo()</code>	검색 쿼리 객체	검색 Rest API 호출 방법
<code>convertSearchResult()</code>	API 서버의 서비스 검색 응답	서비스 목록 및 목록 표시 방법
<code>getServiceURL()</code>	사용자가 선택한 서비스	해당 서비스의 Rest API 호출 방법
<code>convert()</code>	API 서버가 반환한 서비스 정의 응답	서브미션 정의

i 단 `convert()` 반환 결과 형식에 다소 차이가 있습니다.

서브미션 정의

```
{
  "submission"? : "서브미션-정의-노드",
  "request"? : "요청-정의-노드",
  "resposne"? : "응답-정의 노드"
}
```

서브미션 정의 노드

서브미션 정의 노드는 매트릭스 서브미션에 더해 3개의 추가 속성을 더 지정할 수 있습니다.

```
{
  // 호출 URL
  action?: "URL",
  async?: true | false,
  withCredential?: true | false,
  method?: "post" | "get",
```

```
mediaType: 미디어_타입
responseType: 응답_타입
fallbackContentType: 폴백_콘텐츠_타입
}
```

요청/응답 정의 노드

요청/응답 정의노드 규격은 **컴플렉스MDN** 규격에 더해 다음의 제약사항을 갖습니다:

- request의 format은 생략되거나 항상 “requestdata” 이어야 합니다.
- response 노드의 format은 생략되거나 항상 “responsedata” 이어야 합니다.
- 직계 하위 **MDN**들은 그 format이 “object” 또는 “array”인 **컴플렉스 MDN**이어야 합니다.
- 전술한 직계 **컴플렉스MDN**들의 하위 **MDN**들은 모두 **단순 MDN**이어야 합니다.
- **매트릭스 서브미션**과 달리 **컴플렉스 MDN**의 format이 “object”인 경우 반드시 **데이터 맵**, “array”인 경우 **데이터셋**에만 연결 될 수 있습니다.