**Nexacro platform JSON Adaptor**

**JSON Data Layout 정의서**

1. **JSON Adaptor 정의**

* Nexacro platform의 기본통신 전문 (XML, SSV, Binary)을 사용하지 않고, JSON 전문으로 통신할 수 있도록 구현한 통신 Adaptor

주) 기본통신 전문을 JSON으로 변환하는 과정이 필요하며, 그로 인하여 기본통신 전문에 비하여 성능이 다소 하락 할 수 있습니다.

1. **Nexacro platform data type**

* STRING: 문자열. 최대값은 system에서 한번에 할당 가능한 크기인 2GB로 제한됨

Column의 size와 관계없이 최대치까지 사용될 수 있다.

* INT: 정수형 (-231 ~ 231-1)
* ~~BOOL: Boolean 타입. INT(1) 타입으로 0이면 false, 나머지는 true를 의미한다.~~
* FLOAT, DECIMAL: 실수형 (±2.2X10-308 ~ ±1.7X10308)
* BIGDECIMAL: 실수형(문자열로 저장) (±10-1056 ~ ±10+1056)
* DATE: 날짜형 YYYYMMDD (-8192/01/01 ~ 8191/12/31)
* ~~DATETIME : 날짜와시간 YYYYMMDDHHmmssuuu (msec포함가능)~~
* ~~TIME : 6자리 HHmmssuuu(msec포함가능)~~
* BLOB: 이진 데이터 – base 64 encoding

1. **JSON Layout 정의**

* Client의 수신 JSON String 전문은 JSON.parse() 를 통해 object화 할 수 있어야 합니다.
* Server에 보내는 송신 JSON 전문은 object 를 JSON.stringify() 를 통해 변환한 JSON String 전문 입니다.
  1. 최상위 노드의 구조와 순서는 다음과 같이 구성합니다.

|  |
| --- |
| Header |
| Parameters |
| Datasets |

* 1. Header : 송수신 Header 를 정의 합니다.

Header 노드의 기본 구성은 다음과 같습니다.

|  |  |
| --- | --- |
| “datatype” | “JSON” |
| “codepage” | “utf-8” |

* 1. Parameters : 송수신 parameter를 정의 합니다.

Parameters 노드의 기본 구성은 다음과 같습니다.

|  |
| --- |
| "parameters":{"g\_val2":"globalval","Variable0":"","test":"123","aaa":"bbb"} |

* + - parameters는 key:value 형태로 구성합니다.
    - parameters type : Object
  1. Datasets : 송수신 Dataset을 정의 합니다.

Datasets는 여러개의 Dataset으로 구성되기 때문에 Object Array 형태로 구성합니다.

Datasets 노드의 기본 구성은 다음과 같습니다.

|  |
| --- |
| "datasets":  [  {  "ds\_id":"indata",  "ds\_colinfo":  {  "constcolumn":{"ConstColumn0":"10","ConstColumn1":"20"},  "column":  [  {"id":"\_RowType\_","type":"STRING","size":"1"},  {"id":"Column0","type":"STRING","size":"256"},  {"id":"Column1","type":"STRING","size":"256"},  {"id":"Column2","type":"STRING","size":"256"}  ]  },  "ds\_rows":  {  "row":  [  {"\_RowType\_":"U","Column0":"","Column1":"zzzzzzzzzzzzz","Column2":""},  {"\_RowType\_":"N",","Column0":"k",","Column1":"r",","Column2":"g"}  ],  "orgrow":  [  {"\_RowType\_":"O"," Column0":””, " Column1":"ppppp","Column2":""},  {}  ]  }  },  {  "ds\_id":"indata2",  "ds\_colinfo":  {  "constcolumn":{},  "column":  [  {"id":"\_RowType\_","type":"STRING","size":"1"},  {"id":"col1","type":"STRING","size":"256"},  {"id":"col2","type":"STRING","size":"256"}  ]  },  "ds\_rows":  {  "row":  [  {"\_RowType\_":"U","col1":"가나","col2":"다라마"}  ],  "orgrow":  [  {"\_RowType\_":"O","col1":"가나","col2":"다라"}  ]  }  }  ] |

1. 구성

|  |  |  |  |
| --- | --- | --- | --- |
| **datasets** | Dataset Object Array | | |
| **ds\_id** | String Value | |
| **ds\_colinfo** | **constcolumn** | |
|  | Object |
| **column** | |
|  | Object Array |
| **ds\_rows** | **row** | |
|  | Object Array |
| **orgrow** | |
|  | Object Array |

* Bold 체는 예약어임.

1. constcolumn 구성

* key:value 로 구성된 Object 입니다.

|  |
| --- |
| "constcolumn":{"ConstColumn0":"10","ConstColumn1":"20"} |

1. column 구성

* key:value 로 구성된 Object Array입니다.

|  |
| --- |
| "column":  [  **{"id":"\_RowType\_","type":"STRING","size":"1"}, // 고정 값**  {"id":"Column0","type":"STRING","size":"256"},  {"id":"Column1","type":"STRING","size":"256"},  {"id":"Column2","type":"STRING","size":"256"}  ] |

* key의 경우 정의된 값을 사용합니다.(id, type, size)

id : 컬럼의 id를 지정합니다.

Array 0번째 Object의 id는 레코드의 상태를 표시하기 때문에 “\_RowType\_” 으로

미리 정의된 고정 값입니다.

(엔진 내부에서 사용하는 값입니다. 화면에는 표시되지 않습니다.)

type : 컬럼의 데이터 형태를 지정합니다.

데이터 형태에 따라 화면에서 데이터 조작 시 영향을 줍니다.

(STRING, INT, FLOAT, BIGDECIMAL, DATE, BLOB)

size : 컬럼의 크기를 지정합니다.

바인딩된 콤포넌트의 입력 제한 시 사용됩니다.

|  |  |  |  |
| --- | --- | --- | --- |
| Key | id | type | size |
| Value | Array 0번째 : “\_RowType\_”  나머지 index object는 임의 값 사용 | 지정된 값 중 선택사용 | size |
| 예) | Column0 | STRING | 256 |

1. row 구성

* key:value 로 구성된 Object Array입니다.

첫 번째 값은 레코드의 상태를 표시하는 값입니다. (\_RowType\_ 으로 구분)

나머지 컬럼은 임의 지정 값 입니다.

* \_RowType\_에 따른 상태값

N : 변경되지 않은 record로 normal 상태 값 입니다.

**(server 에서 client 로 전송 시 상태 값은 “N”으로 고정합니다.)**

I : 신규 추가된 record로 insert 상태 값 입니다.

U : 수정된 record로 update 상태 값 입니다.

D : 삭제된 record로 delete 상태 값 입니다.

|  |
| --- |
| "row":  [  {"\_RowType\_":"U","Column0":"","Column1":"zzzzzzzzzzzzz","Column2":""},  {"\_RowType\_":"N",","Column0":"k",","Column1":"r",","Column2":"g"}  ] |

1. orgrow 구성

* key:value 로 구성된 Object Array입니다.

**server 에서 client 로 전송 시에는 전문을 구성하지 않습니다.**

**client 에서 server로 전송 시에만 사용합니다.**

첫 번째 값은 레코드의 상태를 표시하는 값입니다. (\_RowType\_ 으로 구분)

나머지 컬럼은 임의 지정 값 입니다.

* Array size는 row 의 Array size와 같습니다.

row 의 \_RowType\_ 이 U 상태인 recode의 원 데이터를 가집니다.

U 이외의 rowtype은 빈 값을 가집니다.

* \_RowType\_에 따른 상태값

O : orgrow 에 표시되며, update된 레코드의 변경 전 record로 original 상태 값 입니다.

|  |
| --- |
| "orgrow":  [  {"\_RowType\_":"O","Column0":"","Column1":"ppppp","Column2":""},  {}  ] |

1. **송수신 샘플 전문**
   1. 송신 (client -> server)

|  |
| --- |
| {  "datatype":"JSON",  "codepage":"utf-8",  "parameters":{"g\_val2":"globalval","Variable0":"","test":"123","aaa":"bbb"},  "datasets":  [  {  "ds\_id":"indata",  "ds\_colinfo":  {  "constcolumn":{"ConstColumn0":"10","ConstColumn1":"20"},  "column":  [  {"id":"\_RowType\_","type":"STRING","size":"1"},  {"id":"Column0","type":"STRING","size":"256"},  {"id":"Column1","type":"STRING","size":"256"},  {"id":"Column2","type":"STRING","size":"256"}  ]  },  "ds\_rows":  {  "row":  [  {"\_RowType\_":"U","Column0":"","Column1":"zzzzzzzzzzzzz","Column2":""},  {"\_RowType\_":"N","Column0":"A","Column1":"B","Column2":""},  {"\_RowType\_":"N","Column0":"k","Column1":"r","Column2":"g"},  {"\_RowType\_":"U","Column0":"","Column1":"","Column2":"LLLL"},  {"\_RowType\_":"D","Column0":"a","Column1":"b","Column2":"c"},  {"\_RowType\_":"I","Column0":"","Column1":"","Column2":""}  ],  "orgrow":  [  {"\_RowType\_":"O","Column0":"","Column1":"ppppp","Column2":""},  {},  {},  {"\_RowType\_":"O","Column0":"","Column1":"","Column2":""},  {},  {}  ]  }  },  {  "ds\_id":"indata2",  "ds\_colinfo":  {  "constcolumn":{},  "column":  [  {"id":"\_RowType\_","type":"STRING","size":"1"},  {"id":"col1","type":"STRING","size":"256"},  {"id":"col2","type":"STRING","size":"256"}  ]  },  "ds\_rows":  {  "row":  [  {"\_RowType\_":"U","col1":"가나","col2":"다라마"}  ],  "orgrow":  [  {"\_RowType\_":"O","col1":"가나","col2":"다라"}  ]  }  }  ]  } |

* 1. 수신 (server -> client)

|  |
| --- |
| {  "datatype":"JSON",  "codepage":"utf-8",  "parameters":{"g\_val2":"globalval","Variable0":"","test":"123","aaa":"bbb"},  "datasets":  [  {  "ds\_id":"indata",  "ds\_colinfo":  {  "constcolumn":{"ConstColumn0":"10","ConstColumn1":"20"},  "column":  [  {"id":"\_RowType\_","type":"STRING","size":"1"},  {"id":"Column0","type":"STRING","size":"256"},  {"id":"Column1","type":"STRING","size":"256"},  {"id":"Column2","type":"STRING","size":"256"}  ]  },  "ds\_rows":  {  "row":  [  {"\_RowType\_":"N","Column0":"","Column1":"zzzzzzzzzzzzz","Column2":""},  {"\_RowType\_":"N","Column0":"A","Column1":"B","Column2":""},  {"\_RowType\_":"N","Column0":"k","Column1":"r","Column2":"g"},  {"\_RowType\_":"N","Column0":"","Column1":"","Column2":"LLLL"},  {"\_RowType\_":"N","Column0":"a","Column1":"b","Column2":"c"},  {"\_RowType\_":"N","Column0":"","Column1":"","Column2":""}  ]  }  },  {  "ds\_id":"indata2",  "ds\_colinfo":  {  "constcolumn":{},  "column":  [  {"id":"\_RowType\_","type":"STRING","size":"1"},  {"id":"col1","type":"STRING","size":"256"},  {"id":"col2","type":"STRING","size":"256"}  ]  },  "ds\_rows":  {  "row":  [  {"\_RowType\_":"N","col1":"가나","col2":"다라마"}  ]  }  }  ]  } |