

V2. GraphQL

Middleware GraphQL 적용

Server Team. 김세영

1. Request Parameter
2. Error Handling
3. Upload File
4. Multi Query
5. API Test Tool & Call URL

V2

V1 Exception 처리

```
@PostMapping("/list")
public Result getProposal(@RequestBody String json) throws Exception {

    Result result = Result.successInstance();
    JSONObject param = new JSONObject(json);
    String did = param.getString("did");
    AuthCheck.checkAuthorities(Command.LIST_PROPOSAL, did);

    FabricAdaptor fabric = new FabricAdaptor();
    JSONArray requestor = JsonBuilder.getIdentifier(did);
    JSONObject qResult = fabric.query(Command.LIST_PROPOSAL, FabricConfig.CONTRACT_TICKET,
        requestor, param.getJSONObject("payload"));

    result.setData(JsonBuilder.getResultData(Command.LIST_PROPOSAL, qResult).toString());
    return result;
}
```

V1 Proposal - get list

```
{
  "command": "listProposal",
  "did": "G5rw9qAMbozGxySHkMaztD",
  "payload": {
    "LowerBound": "2019-08-19T14:40:23+09:00",
    "UpperBound": "2020-05-27T03:39:52.251Z",
    "Index": "UpdatedAt"
  }
}
```

```
{
  "status": 500,
  "message": "Internal Server Error",
  "data": {
    "Code": "8004",
    "Payload": null,
    "Description": "no data found in blockchain"
  }
}
```

- ✓ Status : HTTP Status Code
- ✓ Message : HTTP Status Message
- ✓ Data : Exception Content

V2 Exception 처리

```
public ProposallistResponse getProposals(ProposallistRequest param) throws Exception {  
  
    String did = param.getDid();  
    AuthCheck.checkAuthorities(Command.LIST_PROPOSAL, did);  
  
    Gson gson = new GsonBuilder().excludeFieldsWithoutExposeAnnotation().create();  
    String json = gson.toJson(param);  
  
    FabricAdaptor fabric = new FabricAdaptor();  
    JSONArray requestor = JsonBuilder.getIdentifier(did);  
  
    JSONObject qResult = null;  
    try {  
        qResult = fabric.query(Command.LIST_PROPOSAL, FabricConfig.CONTRACT_TICKET, requestor,  
                               new JSONObject(json));  
    } catch (ContractException e) {  
        throw new FabricException(e.getMessage());  
    }  
}
```

V2 Proposal - get list

```
query {  
  getProposals(param: {  
    command: "listProposal"  
    index: "UpdatedAt"  
    upperBound: "2019-12-08T00:36:12.148Z"  
    lowerBound: "2019-08-19T14:40:23+09:00"  
    did: "G5rw9qAMbozGxySHkMaztD"  
  }) {  
    proposals {  
      category  
    }  
  }  
}
```

```
{  
  "errors": [  
    {  
      "message": "{\\"Code\\":8004,\\"Payload\\":null,\\"Description\\":\\"no data found in blockchain\\"}",  
      "locations": [  
        {  
          "line": 2,  
          "column": 3  
        }  
      ],  
      "path": [  
        "getProposals"  
      ],  
      "extensions": {  
        "type": "FabricException"  
      }  
    }  
  ],  
  "data": {  
    "getProposals": null  
  }  
}
```

Blockchain ←

Middleware ←

```
public enum ErrorType {  
  FabricException,  
  VendorException,  
  DatabaseException,  
  DuplicatedException,  
  FileNotFoundException,  
  WrongTypeException,  
  ContentException,  
  AuthorizeException  
}
```

Prm - resolver

```
public PrivacySetResponse setRisk(BaseRequest param, DataFetchingEnvironment env) {  
    String did = param.getDid();  
    AuthCheck.checkAuthorities(Command.PRM_LEVLES, did);  
  
    DefaultGraphQLServletContext context = env.getContext();  
    List<Part> parts = context.getFileParts();  
    if (parts == null || parts.isEmpty()) {  
        throw new FileSaveException("Failed to save empty file.");  
    }  
  
    String result = privacyService.saveRiskLevels(parts);  
    return PrivacySetResponse.from(result, Command.PRM_LEVLES);  
}
```

현재 실행중인 쿼리에 대한 정보를 제공하는 객체
주로, File 요청 시에 사용함

Prm - set risk

* File 1개 인 경우

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** stag /v2/graphql
- Response Status:** 200 OK
- Response Time:** 256 ms
- Response Size:** 103 B
- Request Type:** Multipart
- File:** privacy_risk_levels.json
- Query:** {"query": "mutation(\$input: BaseRequest){setf
- Response Body (JSON):**

```
1 {
2   "data": {
3     "setRisk": {
4       "message": "privacy_risk_levels.json file saved successfully.",
5       "command": "levels"
6     }
7   }
8 }
```

B2B - upload html

* File n개 인 경우

POST local /v2/graphql Send

Multipart 4 Auth Query 1 Header 2 Docs

file test.html

file test2.html

file test3.html

operations {"query": "mutation(\$did: String) {uploadHtml(did: \$did) {originalFilename fullUrlPath}}"} New name New value

200 OK 25.8 ms 441 B

Preview Header 6 Cookie Timeline

```
1 {
2   "data": {
3     "uploadHtml": {
4       "command": "uploadHtml",
5       "result": [
6         {
7           "originalFilename": "test.html",
8           "fullUrlPath": "http://localhost:8590/cec283c40561141e910b094a5469ed24/html/8V6Upz/test.html"
9         },
10        {
11          "originalFilename": "test2.html",
12          "fullUrlPath": "http://localhost:8590/cec283c40561141e910b094a5469ed24/html/8V6Upz/test2.html"
13        },
14        {
15          "originalFilename": "test3.html",
16          "fullUrlPath": "http://localhost:8590/cec283c40561141e910b094a5469ed24/html/8V6Upz/test3.html"
17        }
18      ]
19    }
20  }
21 }
```

Operations

✓ Prm - set risk

```
{
  "query": "mutation($input: BaseRequest)
  {setRisk(param:$input)
  {message command}}",
  "variables":{"input":
  {"did":"NfoPEFaG1adkCpijR85L66","command":"levels"}}
}
```

```
setRisk(
  param: BaseRequest
): PrivacySetResponse

privacy mutation
```

did
command

✓ B2B - upload html

```
{
  "query": "mutation($did: String)
  {uploadHtml(did: $did)
  {command result{originalFilename fullUrlPath}}}",
  "variables":{"did":"NfoPEFaG1adkCpijR85L66"}
}
```

```
uploadHtml(
  did: String
): UploadLocationResponse

b2b Mutation
```

query

- 실제 요청문

\$variable name

- variables에 정의할 parameter

query name(parameter : \$variable name)

- parameter 정의

variables

- query name에 맞게 데이터 전달

Auth - token (Single)

The screenshot displays a GraphQL client interface with a dark theme. The top bar shows a POST request to `/v2/graphql` with a status of 200 OK, a response time of 32 ms, and a body size of 209 B. The interface is divided into two main panels: the left panel for the query and the right panel for the response.

Left Panel (Query):

- Method: POST
- Endpoint: `/v2/graphql`
- Query:

```
1 query {
2   #user
3   signin (blob: {
4     signature :
5       "7558af49ed63059cc78c5f7cb7fcd2c62764f4f542209bc32100e3ddf600e6e6259fe7f683e1b21ade94163e8b7e82b4deb7926c21652552e3001976d221d04"
6     verkey : "9Dq5ZBptrf7Kgxe3K2Xw3aFwHsdoStiUrs98QPRgbrbt"
7     did : "G5rw9qAMbozGxySHkMaztd"
8   }) {
9     accessToken
10  }
```

Right Panel (Response):

- Status: 200 OK
- Response Time: 32 ms
- Body Size: 209 B
- Preview:

```
1 {
2   "data": {
3     "signin": {
4       "accessToken":
5         "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyY2x1IjoiaVNVNFUiIsImRhdGEiOiJHNXJ3OXF8BTWJvekd4eVNIa01henREIiwiaXhwIjoiaWJyMDk4OTg3NDU5fQ.2Wyp3mZ43u50ypEJ32Xk86PcW-Tl4IplXvo1AM0X4KM"
6     }
7   }
```

Auth - token (Multi)

* 동일 쿼리 전제

POST

stag /v2/graphql

Send

GraphQL

Auth

Query

Header 1

Docs

```

1 query {
2   #user
3   user: signin (blob: {
4     signature :
5     "7558af49ed63059cc78c5f7cb7fcd2c62764f4f542209bc32100e3ddf600e6e6259fe7f683e1b21ade94163e8b7e82b4deb7926c2165
6     2552e3001976d221d04"
7     verkey : "90q5Z8ptrf7Kgxe3K2Xw3aFwH5doStiUrs98QPRgrbrt"
8     did : "G5rw9qAMbozGxySHkMaztD"
9   }) {
10     accessToken
11   }
12   #snplab
13   snplab: signin (blob: {
14     signature :
15     "e2cf3681d058c16a578fa414006141e5b79e6666b63c8a6a26b58611694978fad50a5dbb84727605a0e6ba7d672a32a5a10bbe94a5855
16     382da19d8c577cd430b"
17     verkey : "CozvA8v6Q8sPVmUKEcgnCKCPivtVDkponyEDvB2nXpus"
18     did : "NfoPEFaG1adkCpijR8SL66"
19   }) {
20     accessToken
21   }
22   #b2b
23   b2b: signin (blob: {
24     signature :
25     "e2cf3681d058c16a578fa414006141e5b79e6666b63c8a6a26b58611694978fad50a5dbb84727605a0e6ba7d672a32a5a10bbe94a5855
26     382da19d8c577cd430b"
27     verkey : "CozvA8v6Q8sPVmUKEcgnCKCPivtVDkponyEDvB2nXpus"
28     did : "NfoPEFaG1adkCpijR8SL66"
29   }) {
30     accessToken
31   }
32 }

```

schema

200 OK

48.8 ms

606 B

3 Minutes Ago

Preview

Header 8

Cookie

Timeline

```

1 {
2   "data": {
3     "user": {
4       "accessToken":
5       "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiaVNFUiIsImRhdGEiOiJHNXJ3OXF8TWJvekd4eVNIa01henREIiwiaXhwIjoiaXNjYmDk4ODkwNjEyYFQ.R7
6       BCx9IfJoJay0Stlq40zkH929HwvaYaU9jqm5BqmQ"
7     },
8     "snplab": {
9       "accessToken":
10      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiaVNFU05QTEFCIiwiaXZGF0YSI6Ik5mb1BFRmFwFka0NwawpSODVMNjYiLCJleHAiOjE2MjIwOTg4OTA2MjB9.
11      J0kJEt2uz5AGvSeX53-gP10zMtIAHDK3SZ6izd1N0k4"
12    },
13    "b2b": {
14      "accessToken":
15      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiaVNFU05QTEFCIiwiaXZGF0YSI6Ik5mb1BFRmFwFka0NwawpSODVMNjYiLCJleHAiOjE2MjIwOTg4OTA2MjB9.
16      RAP2oQ0-5g3BiDQWRbidKCQsysocr4IBdfQ81nBQ31E"
17    }
18  }
19 }

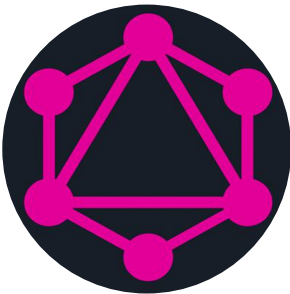
```



Swagger

API Tool : <https://openapi.myd.world/api>

API Call : <https://openapi.myd.world>



Playground

API Tool : <https://openapi.myd.world/v2/playground>

API Call : <https://openapi.myd.world/v2/graphql>

*** Production(상용)은 Playground Tool 지원 안함.**

감사합니다