

# 2. Restfull API

- Gate Token
- API
- PAP(pick-act-page) API



# Gate Token

“

## JWT & OAuth 2.0

JWT 是 JSON 格式 令牌 的缩写。 它用于在分布式系统中进行身份验证。 在 JWT 中，令牌 包含用户信息 和过期时间。 它通过 API 接口 返回给客户端。 客户端在请求其他资源时，将令牌 作为请求头 的一部分发送。 服务器收到请求后，会验证令牌 的有效性。 如果令牌 有效，服务器会返回请求的资源。 如果令牌 无效，服务器会返回错误信息。 这就是 JWT 的基本原理。

“

## Gate Token

dabory sso(OAuth2 接口) 返回 GateToken 接口 API 接口 DB 接口 接口。 GateToken 接口 接口 接口 Frontend 接口 接口 session 接口 gatetoken 接口 check 接口。 接口 gatetoken 接口 接口 main\_api server 接口 api server 接口 接口 接口 接口 接口 接口 接口。 api 接口 GateToken 接口 session 接口。 接口 session 接口 接口 接口 接口 header 接口。



# API 调用

## HTTP Method 说明

- ERP API 调用 POST method 说明。
- API 调用 说明。

## DaborySSO 说明

- DaborySSO API 调用 说明 Key Pair API 调用 SSO 调用 说明

## GateToken:

- API 调用 GateToken 说明。
- GateToken DB 调用 PKI BeforeBase64 Key 说明。
- Frontend Server ClientID 说明。 DaborySSO App 说明。
- Client(Frontend) GateToken API 调用 DB 说明。

API 调用 说明。



# PAP(pick-act-page) API

“

## PAP API

PAP(pick-act-page) API는 Table 데이터를 API로 제공하는 API입니다.

이 API는 데이터를 Pick, Action, row (Page)로 api를 제공합니다.

### pick-api

WHERE Id를 포함한 데이터를 api를 제공합니다.

php

```
$itemPick = $this->callApiService->callApi([
    'url' => 'item-pick',
    'data' => [
        'Page' => [
            ['UpdatedMd5' => $updatedMd,]
        ],
    ],
    'headers' => [
        'GateToken' => $this->gateToken['main']
    ]
]);
```

JavaScript

```
const response = await axios.post('/ajax/get-data', {
    url: 'item-pick',
    data: {
        Page: [
            { Id: parseInt(window.User['SgroupId']) }
        ]
    }
});
```



```
]
}
});
```

## act-api

Insert, Update, Delete는 각각 insert, update, delete 메서드를 호출한다. Id는 insert(0), update(id), delete(id)로 받는다.

insert는 Id를 0으로 받는다. update는 Id를 받아서 update한다.

Delete는 Id를 받아서 delete한다.

### php 코드

```
// insert
$itemAct = $this->callApiService->callApi([
    'url' => 'item-act',
    'data' => [
        'Page' => [
            [
                'Id' => 0, // 0 : insert, id : update, id : delete
                'IgroupId' => 526,
                'ItemCode' => Str::limit($linkproMd5, 21, ''),
                'ItemSlug' => $linkproMd5,
                'ItemName' => $scrap['ItemName'],
                'SalesPrc' => (string)$scrap['SalesPrice'],
            ]
        ],
    ],
    'headers' => [
        'GateToken' => $this->gateToken['main']
    ]
]);

// update
$itemAct = $this->callApiService->callApi([
    'url' => 'item-act',
    'data' => [
```



```

'Page' => [
    [
        'Id' => 4, // 0 : insert, []: update, [] : delete
        'IgroupId' => 526,
        'ItemCode' => Str::limit($linkproMd5, 21, ''),
        'ItemSlug' => $linkproMd5,
        'ItemName' => $scrap['ItemName'],
        'SalesPrc' => (string)$scrap['SalesPrice'],
    ]
],
],
'headers' => [
    'GateToken' => $this->gateToken['main']
]

```

// delete

```

$itemAct = $this->callApiService->callApi([
    'url' => 'item-act',
    'data' => [
        'Page' => [
            [ 'Id' => -4 ] // 0 : insert, []: update, [] : delete
        ],
    ],
    'headers' => [
        'GateToken' => $this->gateToken['main']
    ]
]

```

## JavaScript □□□□

```

const response = await axios.post('/ajax/get-data', {
    url: 'item-act',
    data: {
        Id : 0,
        ItemCode: $(item_form).find('#item-code-txt').val(),
        IgroupId: Number($(item_form).find('#igroup-id-txt').data('id')),
        ItemName: $(item_form).find('#item-name-txt').val(),
        SubName: $(item_form).find('#sub-name-txt').val(),
        ItemSlug: $(item_form).find('#item-slug-txt').val(),
    }
})

```



```

    }
  });

const response = await axios.post('/ajax/get-data', {
  url: 'item-act',
  data: {
    Id : 3,
    ItemCode: $(item_form).find('#item-code-txt').val(),
    IgroupId: Number($(item_form).find('#igroup-id-txt').data('id')),
    ItemName: $(item_form).find('#item-name-txt').val(),
    SubName: $(item_form).find('#sub-name-txt').val(),
    ItemSlug: $(item_form).find('#item-slug-txt').val(),
  }
});

const response = await axios.post('/ajax/get-data', {
  url: 'item-act',
  data: {Id : -3}
});

```

## page-api

[[[ SELECT [[[ [[[[ [[[[ 2[[ [[[ [[[ [[[[[.

Query, Asc, Desc, Limit, Offset [[ [[[[ [[ request[[ data[[ [[[[ [[[[[.

php [[[[

```

$this->callApiService->callApi([
  'url' => 'app-guest-page',
  'data' => [
    'PageVars' => [
      'Query' => "app_name = '$appName' and is_on_use = 1",
      'Limit' => 1,
    ]
  ],
  'headers' => [
    'GateToken' => $this->gateToken['main']
  ]

```



## JavaScript

```
let response = await get_api_data('setting-search-page', {
  QueryVars: {
    QueryName: 'igroup',
    FilterName: 'dbr_igroup.id',
  },
  PageVars: {
    Limit: 9999,
    Offset: 0,
  }
})

// get_api_data(url, data)
```