[JavaScript] Protocal22-API 사

시작하기 전에

이 문서는 Dabory Protocol22-API for JavaScript를 이용해 **Main App**과 **Guest App**을 개발할 **JavaScript**를 활용한 클라이언트 **API** 호출 방식

예제실행전에 아래 준비사항들을 반드시 준비해주시기 바랍니다.

- ☐ script 다운로드
- ☐ app 등록하기

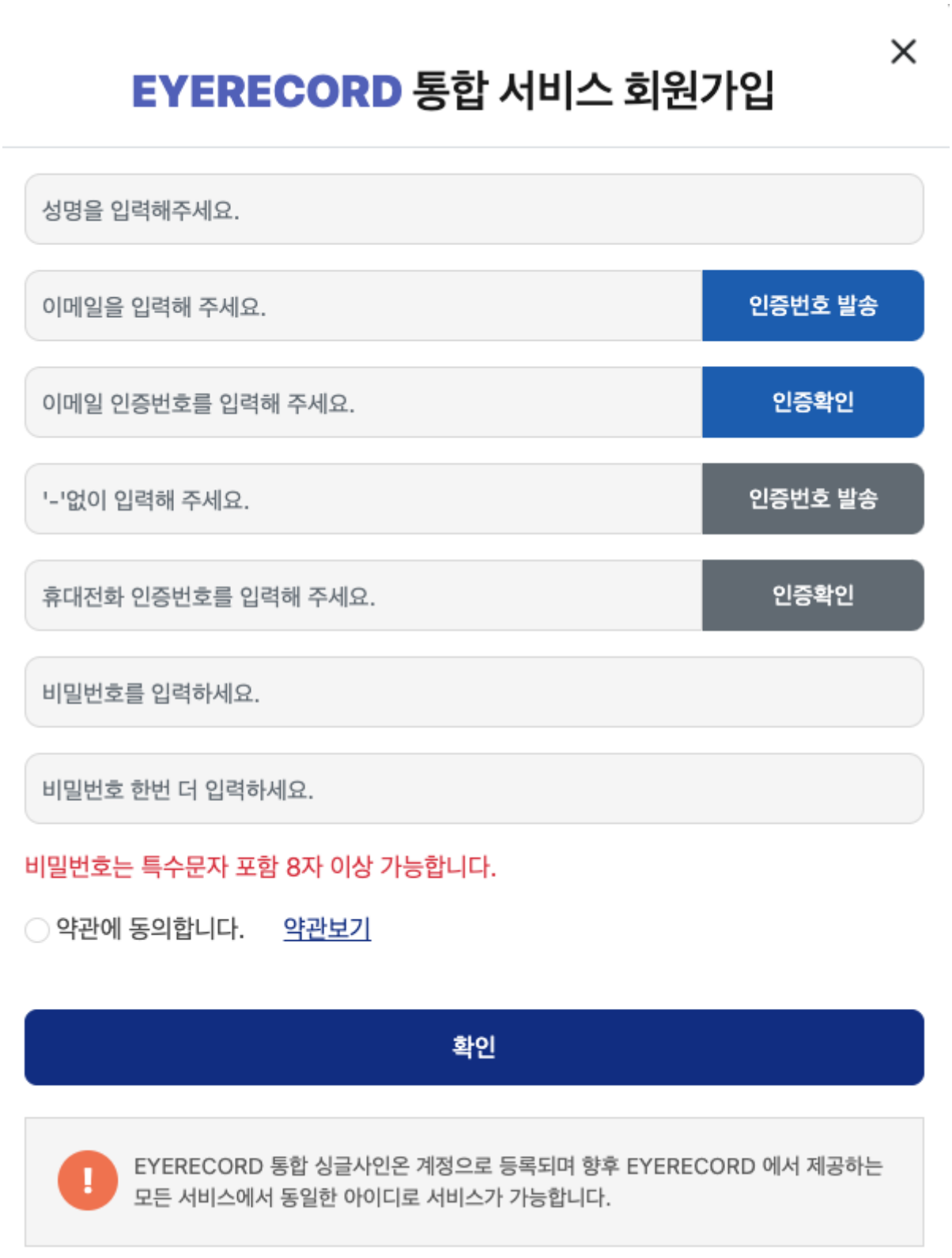
JavaScript 방식의 **API** 호출 흐름

1. 클라이언트 측 JavaScript **OwnerKey** 및 **MAIN APP**의 **URL** 정보를 설정

2. JavaScript에서 ownerApp() 함수를 사용하여 API 요청을 보냅니다.

3. API 응답을 받아 클라이언트 측에서 데이터를 처리합니다.

> “
>
> # JavaScript 사용예제 (회원가입 구현)



script □□□ □□□□

javascript □□□□□ script src □□□ □□□□ url□ ServerUrl□ OwnerKey□ □□ □□□□□.

```
<script src="/js/dabory.owner-key-
linker.js?serverUrl=https://www.daborysso.com&ownerKey=zEUhgZEpPbFpz/cIghtYCsoFlNi8JX3Lh45ank1s+w2P
```

```
GuK4phOxmemgKuuIJ/OQZlrpdSzlK03MpBLBSo1o537brbDPKaq9ydK8QK8JgEAYGKSNzD8W6dJhSIjPfxLJu+cma0w
cAMDmSV7DMhwwgiwVb2ADWu+QOVKpU+VmIR1TBIfT00EPpuW1Ju+jop8WcBEJRrpEAGLK5jcRnAGSXaDTumqEV
sLRSq87hVqRJGImIE7Bf68Ix4b2PqgVhTUY1Bpalw==""></script>
```

<script src="/js/dabory.owner-key-linker.js?serverUrl=&ownerKey=""></script>

serverUrl : MAIN APP□ □□□□□□ □□□□□.

ownerKey : MAIN_APP □□□ □□□□ OwnerKey□ □□□□□.

API □□ □□(OwnerApp)

JavaScript□□ API□ □□□ □□ ownerApp() □□□ □□□ □ □□□□. ajax□ □□□□ API □□□ □□ □ □□□□□.

```
$.fn.ownerKeyLinker.ownerApp = function (url, request, callback, async = true) {
    getParameter();

    $.ajax({
        // ${$.fn.ownerKeyLinker.serverUrl} : script □□□□ url□ □□□ serverUrl
        url: `${$.fn.ownerKeyLinker.serverUrl}/dabory-app/${url}`, //ex. https://www.daborysso.com/dabory-
app/auth/signup
        data: JSON.stringify(request),
        beforeSend: function (xhr) {
            xhr.setRequestHeader('OwnerKey', $.fn.ownerKeyLinker.ownerKey);
        },
        method: 'POST',
        dataType: 'json',
        async: async,
    })
    .done(function (json) {
        callback(json);
    })
    .fail(function (json, textStatus, errorThrown) {
        callback(json);
        iziToast.error({ title: 'Error', message: 'Dabory Owner App Error' });
    });
};
```

url : API □□□ □□□□□(ex. www.daborysso.com/dabory-app/auth/signup)

request : □□□ □□□ □□□□□ (JSON□□)

callback : □□□ □□□ □□□□

async : □□□ □□/□□□ □□ (□□□ true -> □□□ □□)

beforeSend: □□ □□□ setRequestHeader□ □□ OwnerKey□ header□ □□□□.

API □□ □□(OwnerApi)

JavaScript□□ API□ □□□ □□ ownerApi() □□□ □□□ □ □□□□. ajax□ □□□□ API □□□ □□ □ □□□□□.

```
$.fn.ownerKeyLinker.ownerApi = function (url, request, callback, async = true) {
    getParameter(); // $.fn.ownerKeyLinker.serverUrl□ $.fn.ownerKeyLinker.ownerKey□ □□□□ □□

    $.ajax({
        url: `${$.fn.ownerKeyLinker.serverUrl}/dabory-app/call-owner-api`,
        data: JSON.stringify(request),
        beforeSend: function (xhr) {
            xhr.setRequestHeader('OwnerKey', $.fn.ownerKeyLinker.ownerKey);
            xhr.setRequestHeader('Url', url);
        },
        method: 'POST',
        dataType: 'json',
        async: async,
    })
    .done(function (json) {
        callback(json);
    })
    .fail(function (json, textStatus, errorThrown) {
        callback(json);
        iziToast.error({ title: 'Error', message: 'Dabory Owner API Error' });
    });
};
```

url : API □□□ □□□□□(ex. www.daborysso.com/dabory-app/call-owner-api)

request : □□□ □□□ □□□□□ (JSON□□)

callback : □□□ □□□ □□□□

async : □□□ □□/□□□ □□ (□□□ true -> □□□ □□)

beforeSend: 요청 이전에 setRequestHeader를 통해 OwnerKey를 header에 넣습니다.

OwnerApi 함수는 OwnerApp 함수와는 다르게 api 서버에 요청할 api url을 header에 넣습니다.

OwnerApi vs OwnerApp

두 함수는 모두 다보리 API와 통신하는 역할을 하지만 요청을 처리하는 방식과 목적이 다릅니다.

두 함수에서 사[illegible] API 목록은

	OwnerApp()	OwnerApi()
차이점	게스트앱에서 메인앱에 구현된 백엔드 API를 호출할때 (메인앱의 백엔드에 로직 구현 필요)	게스트앱에서 메인앱의 백엔드 API를 거쳐 api 서버에 구현된 api를 호
호출 흐름	guest (frontend)-> main (backend) -> guest(frontend)	guest (frontend)-> main (backend) -> api 서버 (backend) -> guest(frontend)

ownerApi 호출 예제

- API를 호출하여 비밀번호 정책 (passwd-policy) 데이터를 가져오는 예제입니다.

```
$.fn.ownerKeyLinker.ownerApi(
    'remote-signup-setup-get',
    {
        'SetupCode': 'passwd-policy',
        'BrandCode': 'member'
    },
    function (response) {
        window.passwdPolicy = response;
        $('#policy-desc-em').text(window.passwdPolicy['PolicyDesc']);
        console.log(window.passwdPolicy);
        $('#join').modal('show');
    }
);
```

첫번째 인자값 : **api** 서버의 **url**

두번째 인자값 **: api** 서버 요청시 **body**에 포함될 **request** 객체

□□ □□ : response□ □□□□ □□ callback □□

ownerApp □□ □□

- API□ □□□□ □□ □□□□ (auth/send-smsConfirm)□ □□□□ □□□□□.

```
$.fn.ownerKeyLinker.ownerApp('auth/send-smsConfirm', {
    "MobileNo": $('#mobile-no-txt').val(),
}, function (response) {
    if (response['error']) {
        $($this).prop('disabled', false);
        SmsNumber = null;
        iziToast.error({ title: 'Error', message: response['message'] });
    } else {
        SmsNumber = response['data'];
        iziToast.success({ title: 'Success', message: '□□□□ □□ □□' });
    }
});
```

- API□ □□□□ □□ □□□□ (auth/send-emailConfirm)□ □□□□ □□□□□.

```
$.fn.ownerKeyLinker.ownerApp('auth/send-emailConfirm', {
    "Email": $('#email-txt').val(),
}, function (response) {
    if (response['error']) {
        $($this).prop('disabled', false);
        EmailNumber = null;
        iziToast.error({ title: 'Error', message: '□□□□ □□ □□' });
    } else {
        EmailNumber = response['data'];
        iziToast.success({ title: 'Success', message: '□□□□ □□ □□' });
    }
});
```

API□ □□□□ □□□□□ □□□□ □□□□□.

```
$.fn.ownerKeyLinker.ownerApp('auth/signup', {
    "Email": $('#email-txt').val(),
    "Password": $('#password-txt').val(),
    "MobileNo": $('#mobile-no-txt').val(),
    "FirstName": $('#first-name-txt').val(),
}, function (response) {
    if (response['error']) {
        if (response['code'] === 701) {
            iziToast.info({
                title: 'Info',
                message: '□□ □□□□ □□ □□□ □□ □□□□□ □□□□□□□. □□□ □□□ □□ □□ □□□□ □□□.'
            });
        } else if (response['validator']) {
            showErrorMessages(response['message']);
        } else {
            iziToast.info({ title: 'Info', message: response['message'] });
        }
        click_submit_btn(self, false);
    } else {
        iziToast.success({ title: 'Success', message: $('#action-completed').text() });
        $('#join').modal('hide');
    }
});
```

□□□ □□□ : main app□ □□□ api□ url□ □□□□□.

□□□ □□□ : main app□ □□□ api url□□□.

□□□ □□□ : response□ □□□□ □□□ callback □□□□□.

response['error'] : api□ response status□ 200□ □□ □□ respons['error']□ □□□□□.

Revision #32
Created 3 March 2025 09:53:06 by □□□
Updated 5 March 2025 12:44:36 by □□□