

# [JavaScript] Protocal22-API

API

Dabory Protocol22-API for JavaScript Main App Guest App JavaScript API

API

- script
- app

## JavaScript API

- JavaScript OwnerKey MAIN APP URL
- JavaScript ownerApp() API
- API

“

JavaScript ( )



## EYERECORD 통합 서비스 회원가입

성명을 입력해주세요.

이메일을 입력해 주세요.

인증번호 발송

이메일 인증번호를 입력해 주세요.

인증확인

'-'없이 입력해 주세요.

인증번호 발송

휴대전화 인증번호를 입력해 주세요.

인증확인

비밀번호를 입력하세요.

비밀번호 한번 더 입력하세요.

비밀번호는 특수문자 포함 8자 이상 가능합니다.

☐ 약관에 동의합니다. [약관보기](#)

확인



EYERECORD 통합 싱글사인온 계정으로 등록되며 향후 EYERECORD 에서 제공하는 모든 서비스에서 동일한 아이디로 서비스가 가능합니다.

**script** `<script>` `</script>`

javascript `<script>` `</script>` `src` `url` `ServerUrl` `OwnerKey` `</script>`.

```
<script src="/js/dabory.owner-key-
```

```
linker.js?serverUrl=https://www.daborysso.com&ownerKey=zEUhgZEpPbFpz/clghtYCsoFINi8JX3Lh45ank1s+w2P
```

GuK4phOxmemeGKuulJ/OQZlrpdSzIK03MpBLBS01o537brbDPKaq9ydK8QK8JgEAYGKSNzD8W6dJhSijPfxLJu+cma0w  
cAMDmSV7DMhwwgiwVb2ADWu+QOVKpU+VmIR1TBIfT00EPpuW1Ju+jop8WcBEJRpEAGLK5jcRnAGSXaDTumqEV  
sLRSq87hVqRJGImIE7Bf68lx4b2PqgVhTUY1Bpalw==></script>

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

```
serverUrl : MAIN APP [ ] [ ] [ ] [ ] [ ] [ ].
```

```
ownerKey : MAIN_APP [ ] [ ] OwnerKey [ ] [ ] [ ] [ ].
```

## API 11 (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](http://www.daborysso.com/dabory-app/auth/signup))

```
request : [ ] [ ] [ ] (JSON[ ])
```

callback : 関数 呼び出し

async : 非同期/同期 有 (true true -> 同期 有)

beforeSend: 関数 呼び出し setRequestHeader 関数 OwnerKey 関数 header 関数 呼び出し.

## API 関数 呼び出し(OwnerApi)

JavaScript 関数 API 関数 呼び出し ownerApi() 関数 呼び出し 関数 呼び出し. ajax 関数 呼び出し API 関数 呼び出し 関数 呼び出し.

```
$.fn.ownerKeyLinker.ownerApi = function (url, request, callback, async = true) {  
  getParameter(); // $.fn.ownerKeyLinker.serverUrl 関数 呼び出し $.fn.ownerKeyLinker.ownerKey 関数 呼び出し 関数 呼び出し  
  
  $.ajax({  
    url: `${$.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 関数 呼び出し ( ex. www.daborysso.com/dabory-app/call-owner-api)

request : 関数 呼び出し 関数 呼び出し (JSON 関数 呼び出し)

callback : 関数 呼び出し 関数 呼び出し

async : 非同期/同期 有 (true true -> 同期 有)

beforeSend: () => { setRequestHeader('X-OwnerKey', header['X-OwnerKey']);

OwnerApi () => { OwnerApp () { api () { api url() header() {

## OwnerApi vs OwnerApp

API는 클라이언트와 서버 간의 통신을 담당하며, 앱은 클라이언트를 담당합니다.

API는 클라이언트와 서버 간의 통신을 담당하며, 앱은 클라이언트를 담당합니다.

	OwnerApp()	OwnerApi()
API	API는 클라이언트와 서버 간의 통신을 담당하며, 앱은 클라이언트를 담당합니다.	API는 클라이언트와 서버 간의 통신을 담당하며, 앱은 클라이언트를 담당합니다.
API URL	guest (frontend)-> main (backend) -> guest(frontend)	guest (frontend)-> main (backend) -> api () (backend) -> guest(frontend)

## ownerApi ()

- API는 클라이언트와 서버 간의 통신을 담당하며, 앱은 클라이언트를 담당합니다.

```
$.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');  
  }  
);
```

API URL : api () url

API body : api () body request

📄 📄 : **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: '❌ Email or Password is incorrect. Please check and try again.'
      });
    } 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');
  }
});

```

**API URL : main app API url**

**API URL : main app API url**

**API URL : response callback**

**response['error']** : api response status 200 response['error']