Get Started

Web

With the BSA Web, BSA Authentication can be used on the web browser. This document will describe how to make best use of the BSA Web Library.

Introduction

BSA Web Library consists of BSA-JS which enables custom configuration and the Login-Component that can configure default authentication page without further configuration.

Registration Process

Overview

In order to fully utilize the BSA Web, the client must be registered to generate the client key. Please contact the FNSVALUE development team regarding this matter.

Request for Issuance

Only the minimum information will be collected in the process of client key generation as follows.

- Company name

Client Key Generation

If the request is successfully completed, the client key will be generated like below.

```
{
    "clientKey" :
"ea3aca8g59354cff908tu7fae6849d06"
}
```

Prerequisites for Authentication

If the actual user wishes to activate the BSA authentication, pre-registration process is required as described below.

1. Download the BSA mobile app and sign in
    i. Search for the 'BSA' or 'fnsvalue' at the Play Store/App Store and download the app
    ii. In case of new user, sign in is required
2. Link to the client website
    i. Go through Menu > My information > Trusted Website
    ii. Search for the client websites' name
    iii. Click the [Link] button to connect with the client site
3. Now the user can authenticate with the BSA information

BSA-JS

Introducing the BSA, a JavaScript library built for an easy authentication on the web.

Load BSA-JS

Latest version of `BSA-JS` is now available.

```
<script type="text/javascript"
src="https://resource.fnsbsa.com/resources/itu-sandbox/bsa-web-
sdk.js"></script>
```

Specify the version

If necessary, specify the version on the `BSA-JS` URL as below.

```
<script type="text/javascript"
src="https://resource.fnsbsa.com/resources/itu-sandbox/bsa-
web-sdk.js "></ script>
```

Methods provided

Below are the methods provided by the BSA-JS

Name	Description
constructor	Constructor of the BSA-JS
requestAuth	Handles the BSA authentication request, and redirect to the setup URL after authentication
requestAuthCallback	Handles the BSA authentication request, and returns the result to the setup callback function after authentication
onCancel	Cancels the BSA authentication request
setAuthTimer	Provides remaining time for authentication
setAuthMessage	Provides authentication process message
requestQr	Handles the BSA QR authentication request, and redirect to the setup URL after authentication

requestQrCallback	Handles the BSA QR authentication request, and returns the result to the setup callback function after authentication
onQrCancel	Cancels the BSA QR authentication request
setQrTimer	Provides remaining time for QR authentication
setQrMessage	Provides QR authentication process message
requestOtp	Handles the BSA OTP authentication request, and redirect to the setup URL after authentication
requestOtpCallback	Handles the BSA OTP authentication request, and returns the result to the setup callback function after authentication
onOtpCancel	Cancels the BSA OTP authentication request

setOtpTimer	Provides remaining time for OTP authentication
setOtpMessage	Provides OTP authentication process message
requestTotpCallback	Handles the BSA TOTP authentication request, and returns the result to the setup callback function after authentication

Function Description and Example

Constructor

A constructor is an **`essential element`** for the BSA-JS. The client key which is necessary to utilize the BSA-JS, can be confirmed by inquiring the person in charge or directly at the BSA Portal.

```
constructor(clientKey)
```

Parameter

Name	Type	Description
clientKey	String	Client key generated to utilize BSA

Example

```
const bsa = new BSA("{Client Key}");
```

BSA Authentication

This document describes how to utilize BSA authentication in the BSA-JS

Authentication Request

When the authentication is requested, the API call will be made with `requestAuthCallback()` . Push notification will be sent to the app, and the result will be returned through `onSuccess` if successfully authenticated.

```
requestAuthCallback(userKey, successCallback, errCallback)
```

Parameter

Name	Type	Description
userKey	String	BSA user account

Example

```
const bsa = new BSA("{Client Key}"); bsa.requestAuthCallback(userKey,
 (data) => {
   console.log('onSuccess');
   console.log('data : ', data);
}, (errorCode, errorMsg) => {
   console.log('onError');
   console.log('errorCode : ',
errorCode);
   console.log('errorCode : ',
errorMsg);
});
```

onSuccess

Name	Type	Description
data	String	Token

The token will be returned if authentication succeeds, and it can be utilized for the BSA authentication.

onError

Name	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If authentication fails, the error code and error message will be returned.
Possible error codes are as follows.

ErrorCode	Description	Solution
2000	Invalid client key	Check the client key
2008	Unregistered user	Check BSA sign in status
3201	Not properly linked client	After signing up for BSA, go through Menu => My BSA => Trusted Website => Site Link and then connect with the client website
3301	Unspecified client login type	Error in specifying the client, contact the person in charge to solve this matter
5001	Authentication timeout	Make request for authentication once again because previous authentication is no longer valid

5005	Unauthorized user	Contact the person in charge to solve this matter
5006	Temporarily suspended user	Contact the person in charge to solve this matter
5007	Permanently suspended user	Contact the person in charge to solve this matter
5008	Withdrawn user	User accounts can be reactivated within a certain period of time
2010	User authentication in-progress	Depending on the circumstances, cancel previous authentication and request for new one
5011	User authentication canceled	Make request for re-authentication
5015	Failed to create channel	It can occur when the parameters are not enough If it happens constantly, please inquire the person in charge
5017	Failed to send push notification	Problems have occurred with the FCM (Firebase Cloud Messaging), etc. If it happens constantly, please inquire the person in charge
5022	Verification failure	Node verification failed If it happens constantly, please inquire the person in charge

Cancel Authentication

Authentication in progress will be canceled if requested. The API call will be made with `onCancel()` and users can request for authentication again any time.
If the cancel request is successful, `5011` errorCode will be returned. More in detail can be found in the onError.

```
onCancel(userKey, errCallback)
```

Parameter

Name	Type	Description
userKey	String	BSA user account

Example

```
const bsa = new BSA("{Client Key}"); bsa.onCancel(userKey,
(errorCode, errorMsg) =>
{
  console.log('onError');
console.log('errorCode : ', errorCode);
  console.log('errorCode : ', errorMsg);
});
```

onError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If cancel request fails, the error code and error message will be returned.
Possible error codes are like below.

ErrorCode	Description	Solution
3100	Unregistered user	Check the user key requested
5019	No authentication in progress	Authentication has been already canceled, or not in progress now

Set Authentication Timer

Add callback function to check valid BSA authentication time. The remaining time for authentication will be displayed and if expired, the user should request for authentication once again.

```
setAuthTimer(onCallBack)
```

Parameter

- none

Example

```
const bsa = new BSA("{Client Key}");
bsa.setAuthTimer((time) => {
  console.log('onTime');
  console.log('time : ' + time); });
```

onTime

Key	Type	Description
time	Int	Valid authentication time

Valid authentication time will be returned as theresult of a callback function.

Set Authentication Status

Add callback function to check BSA authentication status.
It is possible to see the authentication status during the whole process from authentication request to the final authentication.

```
setAuthMessage(onCallBack)
```

Parameter

- none

Example

```
const bsa = new BSA("{Client Key}");
bsa.setAuthMessage((message) => { console.log('onMessage');
  console.log('AuthStatus : ' + message);
});
```

onMessage

Key	Type	Description
message	String	Authentication status

Authentication status will be returned as the result of a callback function.

QRAuthentication

This document describes how to utilize QR authentication in the BSA-JS

Function Description

QR Authentication can be used via mobile devices without an ID by following the steps below.
First, create a QR code implemented by `bsa.js` and then click `QR Authentication` on the main screen of `BSA` app to activate the QR Scanner.

QR Authentication Request

When authentication is requested, API calls are made with `requestQrCallback()` .
Upon the request of QR authentication, a QR code will be created on the `<canvas/> tag` that has been set up with`BSA-JS` . The result will be returned with `onSuccess` if successfully authenticated.

```
requestQrCallback(qrCanvas, successCallback, errCallback)
```

Parameter

Name	Type	Description
qrCanvas	Element	`<canvas/>` element to create BSA QR code

Example

```
const bsa = new BSA("{Client Key}"); bsa.requestQrCallback(qrCanvas,
(data) => {
  console.log('onSuccess');    console.log('data :
', data); }, (errorCode, errorMsg) => {
     console.log('onError');
     console.log('errorCode : ',
   errorCode);
     console.log('errorCode : ',
   errorMsg);
   });
```

onSuccess

Key	Type	Description
data	String	Token

The token will be returned if authentication succeeds, and it can be utilized for the BSA authentication.

onError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If authentication fails, the error code and error message will be returned.

Possible error codes are as follows.

Error Code	Description	Solution
2000	Invalid client key	Check the client key
2008	Unregistered user	Check BSA sign in status
3201	Not properly linked client	After signing up for BSA, go through Menu => My BSA => Trusted Website => Site Link and connect with the client website
5001	Authentication timeout	Make request for authentication once again, because previous authentication is no longer valid
5005	Unauthorized user	Contact the person in charge to solve this matter
5006	Temporarily suspended user	Contact the person in charge to solve this matter
5007	Permanently suspended user	Contact the person in charge to solve this matter
5008	Withdrawn user	User accounts can be reactivated within certain period of time by reactivation
2010	User authentication in-progress	Depending on the circumstances, cancel previous authentication and request for new one
5011	User authentication canceled	Make request for re-authentication
5015	Failed to create channel	It can occur when the parameters are not enough If it happens constantly, please inquire the person in charge
5017	Failed to send push notification	Problems have occurred with the FCM(Firebase Cloud Messaging), etc. If it happens constantly, please inquire the person in charge
5022	Verification failure	Node verification failed If it happens constantly, please inquire the person in charge
5023	Invalid QR ID	It can occur when the authentication has expired. In this case, re-authentication should be requested
5024	Invalid QR URL CLIENT	It can occur when the QR code was scanned through

		another app other than BSA. It must be scanned by BSA app only

Cancel QR Authentication

Authentication in progress will be canceled if requested. Users can request for authentication again any time.
If the cancel request is successful, `5011` errorCode will be returned. More in detail can be found in the onError

```
onQrCancel(qrCanvas, errCallback)
```

Parameter

Name	Type	Description
qrCanvas	Element	`<canvas/>` element to create BSA QR code

Example

```
<div>
  <canvas />
</div>
```

```
const bsa = new BSA("{Client Key}");
bsa.onQrCancel(qrCanvas,
(errorCode, errorMsg) => {
           console.log('onError');
    console.log('errorCode : ', errorCode);
    console.log('errorCode : ', errorMsg);
});
```

onError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If cancel request fails, the error code and error message will be returned. Possible error codes are like below.

ErrorCode	Description	Solution
3100	Unregistered user	Check the user key requested
5019	No authentication in progress	Authentication has been already canceled, or not in progress now

Set QR Authentication Timer

Add callback function to check valid BSA QR Authentication time. The remaining time for authentication will be displayed and if expired, the user should request for authentication again.

```
setQrTimer(onCallBack)
```

Parameter

- none

Example

```
const bsa = new BSA("{Client Key}"); bsa.setQrTimer((time)
=> {
  console.log('onTime');
console.log('time : ' + time); });
```

onTime

Key	Value	Description
time	Int	Valid authentication time

Valid authentication time will be returned as the result of a callback function.

Set QR Authentication Status

Add callback function to check BSA authentication status.
It is possible to see the authentication status during the whole process from authentication request to the final authentication.

```
setQrMessage(onCallBack)
```

Parameter

- none

Example

```
const bsa = new BSA("{Client Key}");
bsa.setQrMessage((message) => {
  console.log('onMessage');
console.log('AuthStatus : ' + message);
});
```

onMessage

Key	Value	Description
message	String	Authentication status

Authentication status will be returned as the result of a callback function.

OTP Authentication

This document describes how to utilize the OTP Authentication in the BSA-JS

Function Description

OTP Authentication can be used via mobile devices without an ID by following the steps below.
First, click `OTP Authentication` from the main screen of `BSA` app and then get the OTP code. Proceed on authentication by entering the OTP code.

OTP Authentication request

When the user enters the OTP code to authenticate, the API call will be made with `requestOtpCallback()` .
Push notification will be sent to the app, and the result will be returned through `onSuccess` if successfully authenticated.

```
requestOtpCallback(otpInput, successCallback, errCallback,
codeSuccessCallback,codeErrCallback)
```

Parameter

Name	Type	Description
otpInput	Element	`<input/>` element for user to enter the OTP code

Example

```
<div>
  <Input />
</div>
```

```
const bsa = new BSA("{Client Key}"); bsa.requestOtpCallback(otpInput,
(data) => {
  console.log('onSuccess');    console.log('data :
', data); }, (errorCode, errorMsg) => {
     console.log('onError');
     console.log('errorCode : ',
  errorCode);
    console.log('errorCode : ',
  errorMsg);
  }, () => {
    console.log('onCodeSuccess');
  }, (errorCode, errorMsg) => {
    console.log('onCodeError');
    console.log('errorCode : ',
  errorCode);
    console.log('errorCode : ',
  errorMsg);
  });
```

onSuccess

Key	Type	Description
data	String	Token

The token will be returned if authentication succeeds, and it can be utilized for the BSA authentication.

onError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If authentication fails, the error code and error message will be returned.

Possible error codes are as follows.

ErrorCode	Description	Solution
2000	Invalid client key	Check the client key
2008	Unregistered user	Check BSA sign in status
3201	Not properly linked client	After signing up for BSA, go through Menu => My BSA => Trusted Website => Site Link and connect with the client website
3301	Unspecified client login type	Error with specifying the client, contact the person in charge to solve this matter
5001	Authentication timeout	Make request for authentication once again, because previous authentication is no longer valid
5005	Unauthorized user	Contact the person in charge to solve this matter
5006	Temporarily suspended user	Contact the person in charge to solve this matter
5007	Permanently suspended user	Contact the person in charge to solve this matter
5008	Withdrawn user	User accounts can be reactivated within certain period of time by reactivation
2010	User authentication in-progress	Depending on the circumstances, cancel previous authentication and request for new one
5011	User authentication canceled	Make request for re-authentication
5015	Failed to	It can occur when the parameters are not enough

	create channel	If it happens constantly, please inquire the person in charge
5017	Failed to send push notification	Problems have occurred with the FCM(Firebase Cloud Messaging), etc. If it happens constantly, please inquire the person in charge
5022	Verification failure	Node verification failed If it happens constantly, please inquire the person in charge

onCodeSuccess

If the OTP code verification is successful, this function will be called.

It can have null value or can be omitted.

onCodeError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If the OTP code verification fails, the error code and error message will be returned.

It can have null value or can be omitted. Possible error codes are as follows.

ErrorCode	Description	Solution
2000	Invalid client key	Check the client key
3005	OTPcode verification failure	Make request for re-verification
3201	Not properly linked client	After signing up for BSA, go through Menu => My BSA => Trusted Website => Site Link and connect with the client website

Cancel OTP Authentication

Authentication in progress will be canceled if requested. Users can request for authentication again any time.
If the cancel request is successful, `5011` errorCode will be returned. More in detail can be found in the onError

```
onOtpCancel(otpInput, errCallback)
```

Parameter

Name	Type	Description
otpInput	Element	`<input/>` element for user to enter the OTP code

Example

```
<div>
  <Input />
</div>
```

```
const bsa = new BSA("{Client Key}");
bsa.onOtpCancel(otpInput,
(errorCode, rtMsg) => {
    console.log('onError');
    console.log('errorCode : ',
  errorCode);
       console.log('errorCode : ',
  errorMsg);
  });
```

onError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error message

If cancel request fails, the error code and error message will be returned.
Possible error codes are like below.

ErrorCode	Description	Solution
3100	Unregistered user	Check the user key requested
5019	No authentication in progress	Authentication has been already canceled, or not in progress now

Set OTP Authentication Timer

Add callback function to check valid OTP authentication time.
The remaining time for authentication will be displayed and if expired, the user should request for authentication again

```
setOtpTimer(onCallBack)
```

Parameter

- none

Example

```
const bsa = new BSA("{Client Key}"); bsa.setOtpTimer((time)
=> {

    console.log('onTime');
    console.log('time : ' + time);

});
```

onTime

Key	Value	Description
time	Int	Valid authentication time

Valid authentication time will be returned as the result of a callback function.

Set OTP Authentication Status

Add callback function to check BSA authentication status.
It is possible to see the authentication status during the whole process from authentication request to the final authentication.

```
setOtpMessage(onCallBack)
```

Parameter

- none

Example

```
const bsa = new BSA("{Client Key}");
bsa.setOtpMessage((message) => {
           console.log('onMessage');
    console.log('AuthStatus : ' + message);
});
```

onMessage

Key	Value	Description
message	String	Authentication status

Authentication status will be returned as the result of a callback function.

TOTP Authentication

This document describes how to utilize the TOTP authentication in the BSA-JS. TOTP authentication can be used when the mobiledevice cannot access the internet.

Function Description

TOTP authentication can be used via mobile devices with the ID and TOTP code by following the steps below.
First, click **`TOTP Authentication`** from the main screen of `BSA` app and then get the TOTP code. Proceed on authentication by entering the ID and TOTP code.

TOTP Authentication Request

When the user enters the TOTP code to authenticate, the API call will be made with **`requestTotpCallback()`** . The result will be returned through **`onSuccess`** if successfully authenticated.

```
requestTotpCallback(userKey, totpCode, successCallback, errCallback)
```

Parameter

Name	Type	Description
userKey	String	BSA user account
totpCode	String	TOTP code that user entered

Example

```
const bsa = new BSA("{Client Key}"); bsa.requestTotpCallback(userKey,
totpCode, (data) => {
  console.log('onSuccess');    console.log('data : ',
data); }, (errorCode, errorMsg) => {
   console.log('onError');
   console.log('errorCode : ',
 errorCode);
   console.log('errorCode : ',
 errorMsg);
 });
```

onSuccess

Key	Type	Description
data	String	Token

The token will be returned if authentication succeeds, and it can be utilized for the BSA authentication.

onError

Key	Type	Description
errorCode	Int	Error code
errorMsg	String	Error Message

If authentication fails, the error code and error message will be returned.

Possible error codes are as follows.

ErrorCode	Description	Solution
2000	Invalid client key	Check the client key
2008	Unregistered user	Check BSA sign in status
3005	TOTP code verification failure	Make request for re-verification
3201	Not properly linked client	After signing up for BSA, go through Menu => My BSA => Trusted Website => Site Link and connect with the client website
3301	Unspecified client login type	Error with specifying the client, contact the person in charge to solve this matter
5001	Authentication timeout	Make request for authentication once again because previous authentication is no longer valid
5005	Unauthorized user	Contact the person in charge to solve this matter
5006	Temporarily suspended user	Contact the person in charge to solve this matter
5007	Permanently suspended user	Contact the person in charge to solve this matter
5008	Withdrawn user	User accounts can be reactivated within certain period of time by reactivation
2010	User authentication in-progress	Depending on the circumstances, cancel previous authentication and request for new one
5011	User authentication canceled	Make request for re-authentication
5015	Failed to create channel	It can occur when the parameters are not enough. If it happens constantly, please inquire the person in charge
5017	Failed to send push notification	Problems have occurred with the FCM(Firebase Cloud Messaging), etc. If it happens constantly, please inquire the person in charge
5022	Verification failure	Node verification failed If it happens constantly, please inquire the person in charge
5026	Exceeded daily limit for TOTP authentication attempt	Make request for authentication with another method