

CAM-IE



Integration Document

Version 1.0

May 2025

Issued by **Canadia Technology Group**

Table of contents

Table of contents.....	2
Changelog.....	3
Introduction.....	4
Basic Integration.....	4
Vue.js Integration.....	4
Vue 3.....	4
Vue 2.....	4
Usage.....	5
Invoke Methods.....	5
getFullProfile.....	5
shareContent.....	6
closeApp.....	7
Events Listener.....	8
onNavigateBack.....	8

Changelog

Date	Title	Description
06 May 2025	Release v1.0	Initial Release

Introduction

Instructions for integrating a Cam-IE's mini app web platform using Javascript or Vue.js.

Basic Integration

```
const bridge = window.miniappbridge.init();
```

Vue.js Integration

The SDK can be easily integrated with both Vue 2 and Vue 3 applications.

Vue 3

```
// main.js
import { createApp } from 'vue';
import App from './App.vue';
import { VueMiniAppBridge } from 'canadia_bank_bridge.min.js';

const app = createApp(App);
app.use(VueMiniAppBridge, {
  debugMode: false
});

app.mount('#app');
```

Vue 2

```
// main.js
import Vue from 'vue';
import App from './App.vue';
import { VueMiniAppBridge } from 'canadia_bank_bridge.min.js';

Vue.use(VueMiniAppBridge, {
  debugMode: false
});

new Vue({
  render: h => h(App),
}).$mount('#app');
```

Usage

Invoke Methods

getFullProfile

- **Description:** Retrieves the user's detailed profile information.
- **Returns:**
 - A Promise that resolves with an object containing the following fields:
 - **id:** (string) User's ID.
 - **firstName:** (string) User's first name.
 - **lastName:** (string) User's lastName.
 - **email:** (optional) User's email
 - **phone:** (string) User's phone number
 - **sex:** (string) User's gender e.g (F, M).
 - **nid:** (string) User's National Identification Number.
 - **nidType:** (string) Identify Type. //1 => NID, 2 => PASSPORT
- **Example:**

```
this.$bridge.callHandler("getFullProfile")
.then(profile => {
  console.log('First Name:', profile.firstName);
  console.log('Last Name:', profile.lastName);
  console.log('Nid:', profile.nid);
})
.catch(error => {
  console.error('Error getting profile:', error);
});
```

shareContent

- **Description:** Share file via the native platform's share sheet
- **Parameters:**
 - A Promise that resolves with an object containing the following fields:
 - **content:** (string) Base64 encoded content.
 - **name:** (optional string) Random value will be used if not available
 - **type:** (string) Mime types // **image/png**, **text/plain**, **application/pdf**
- **Returns:**
 - A Promise that resolves with an object containing the following fields:
 - **status:** (bool) A **true/false** value to indicate that file has been shared successfully
- **Example:**

```
this.$bridge.callHandler("shareContent", {
  content: '{base64}',
  fileName: '{name}.{ext}',
  type: 'application/pdf'
})
.then(result => {
  console.log('Status:', result.status);
})
.catch(error => {
  console.error('Error Sharing Content:', error);
});
```

closeApp

- **Description:** Close the current open mini app.
- **Parameters:** N/A
- **Example:**

```
this.$bridge.callHandler("closeApp")
```

Events Listener

onNavigateBack

- **Description:** Registers a back button click listener for navigation. The MiniApp manages its back navigation state and must call [closeApp](#) when no further back navigation is possible.
- **Parameters:**
 - **callback:** (function) The callback function that will be executed when clicking the back button in the navigation appbar.
- **Example:**

```
this.$bridge.registerHandler('onNavigateBack', () => {  
  console.log('Navigate Back event');  
});
```