

Confirmed Understanding

The Flow:

1. We will provide iBank with client credentials (`clientId` and `clientSecret`)
2. iBank will call our token endpoint (`POST /api/token/request`) with these credentials
3. We will validate the credentials and return a secure token + the iframe URL
4. iBank will embed our app in an iframe using the tokenized URL
5. Our app will validate the token server-side before serving the content

Your Team's Work:

- Integrate token request into iBank mobile app/Internet Banking
- **Provide account number and client number** from the logged-in user's session
- **Append these parameters** (`accountNumber`, `clientNumber`) along with the token to the iframe URL
- Display our app in an iframe with the complete URL
- Handle token refresh if needed

Our Work:

- Build the token generation endpoint
- Implement server-side token validation
- Provide API documentation and credentials

Application Overview

As referenced in our integration guide, we have two H5 applications ready for integration:

- **Airtime Top-Up App**: <https://h5-getbucks-airtime.vercel.app>
- **Bill Payments App**: <https://h5-getbucks-bill-payments.vercel.app>

Both applications support iframe embedding and are designed to work with the token-based security system we're implementing.

What We'll Build

1. Token Generation Endpoint

Endpoint: `POST /api/token/request`

Request:

```
{
  "clientId": "ibank-client-id",
  "clientSecret": "ibank-client-secret",
  "app": "bill-payments",
  "userId": "optional-user-id"
}
```

Request Parameters:

- **clientId** (string, required): Client ID provided by us
- **clientSecret** (string, required): Client secret provided by us
- **app** (string, required): Application identifier - specifies which app to access
 - **Available options:**
 - **"airtime"** - Airtime Top-Up App
 - **"bill-payments"** - Bill Payments App
- **userId** (string, optional): User identifier for session tracking

Response:

```
{
  "success": true,
  "token": "jwt-token-here",
  "expiresIn": 3600,
  "baseUrl": "https://h5-getbucks-bill-payments.vercel.app",
  "app": "bill-payments"
}
```

Response Fields:

- **token** (string): JWT token for iframe security
- **expiresIn** (number): Token expiration time in seconds (3600 = 1 hour)
- **baseUrl** (string): Base URL for the requested app
- **app** (string): Confirmation of which app the token is for

Available Apps:

- **Airtime Top-Up:** **app:** "airtime" → **baseUrl:** "https://h5-getbucks-airtime.vercel.app"
- **Bill Payments:** **app:** "bill-payments" → **baseUrl:** "https://h5-getbucks-bill-payments.vercel.app"

Important:

- iBank must specify which app they want to access using the **app** parameter
- The response returns the **baseUrl** for the specified app and **token**
- **iBank must append the required parameters** (**accountNumber**, **clientNumber**, etc.) when constructing the iframe URL

2. Token Validation

- Server-side validation before serving the app
- Token expiration handling (suggested: 1 hour lifetime)
- Secure token format

3. iBank Implementation: Constructing the Iframe URL

iBank is responsible for providing and appending the following parameters to the iframe URL:

Required Parameters (must be appended by iBank):

- **token** - The security token received from the token endpoint
- **accountNumber** - Customer account number (debit account) - **provided by iBank, must be appended**
- **mode** - Integration mode: **iframe** (for Internet Banking) or **native** (for Mobile App)

Recommended Parameters (should be appended by iBank):

- **clientNumber** - Client number for account transfer creation - **provided by iBank**
- **returnUrl** - URL to redirect after transaction completion (optional)

Implementation Steps for iBank:1. **Request Token** (specify which app to access):

```
POST /api/token/request
{
  "clientId": "ibank-client-id",
  "clientSecret": "ibank-client-secret",
  "app": "bill-payments" // or "airtime" for Airtime Top-Up app
}
```

App Options:

- **"airtime"** - For Airtime Top-Up App
- **"bill-payments"** - For Bill Payments App

2. **Receive Response** (includes baseUrl for the requested app):

```
{
  "success": true,
  "token": "jwt-token-here",
  "expiresIn": 3600,
  "baseUrl": "https://h5-getbucks-bill-payments.vercel.app",
  "app": "bill-payments"
}
```

Note: The **baseUrl** will match the app specified in the request:

- If **app: "airtime"** → **baseUrl: "https://h5-getbucks-airtime.vercel.app"**
- If **app: "bill-payments"** → **baseUrl: "https://h5-getbucks-bill-payments.vercel.app"**

3. **Construct Iframe URL** (iBank appends parameters):

```
// iBank constructs the complete URL
const accountNumber = getCurrentUserAccountNumber(); // From iBank system
const clientNumber = getCurrentUserClientNumber(); // From iBank system
```

```
const token = response.token; // From token endpoint
const baseUrl = response.baseUrl; // From token endpoint

const iframeUrl = `${baseUrl}?
token=${token}&mode=iframe&accountNumber=${accountNumber}&clientNumber=${cli
entNumber}`;
```

4. Embed in Iframe:

```
<iframe
  src="https://h5-getbucks-bill-payments.vercel.app/?token=jwt-token-
here&mode=iframe&accountNumber=00001203&clientNumber=023557"
  width="100%"
  height="600px"
  frameborder="0">
</iframe>
```

Example Complete URL (constructed by iBank):

```
https://h5-getbucks-bill-payments.vercel.app/?token=jwt-token-
here&mode=iframe&accountNumber=00001203&clientNumber=023557
```

Note:

- `accountNumber` and `clientNumber` are **provided by iBank** from their system (logged-in user's account details)
- These parameters **must be appended** to the URL by iBank when embedding the iframe
- The token endpoint only provides the `token` and `baseUrl` - iBank constructs the complete URL

BankWare/GetBucks Bank API Authentication

The applications need to authenticate with the GetBucks Bank Account Transfer API (BankWare) to process payments. We need to understand how iBank will provide these authentication credentials.

Required Information from iBank:

Please clarify how you will provide the OAuth credentials needed for BankWare API authentication:

1. OAuth Credentials Format:

- `username`: API username for BankWare
- `password`: API password for BankWare
- `systemId`: System identifier
- `grant_type`: OAuth grant type (typically "password")

2. Credential Delivery Method:

- Will these be provided directly to us for configuration?

- Should they be passed via URL parameters?
- Will iBank handle the OAuth token acquisition and pass the access token?
- Or another method?

3. Token Management:

- Who will manage OAuth token acquisition and refresh?
- Should the app request tokens from iBank, or will iBank provide pre-authenticated tokens?
- What is the expected token lifetime?

Current App Behavior: The apps are configured to obtain OAuth tokens from the GetBucks Bank `/token` endpoint using the credentials above. However, we need to know if iBank will:

- Provide the credentials for us to configure
- Handle OAuth token acquisition and pass tokens to the app
- Use a different authentication method

Please advise on your preferred approach for BankWare API authentication.

Questions for Clarification

1. **Token Lifetime:** Is 1 hour expiration acceptable, or do you prefer a different duration?
2. **Client Credentials Exchange:** How would you like to securely exchange the `clientId` and `clientSecret` for the iframe token endpoint? (Secure channel, encrypted email, etc.)
3. **BankWare API Authentication: Please clarify how iBank will provide OAuth credentials for BankWare/GetBucks Bank Account Transfer API:**
 - Will you provide the OAuth credentials (username, password, systemId) directly to us for configuration?
 - Or will iBank handle OAuth token acquisition and pass access tokens to the app?
 - What is your preferred method for providing these credentials?
4. **User Context:** Should tokens include user-specific information, or be session-based only?
5. **Token Refresh:** When a token expires, should iBank automatically request a new one, or handle it manually?
6. **Account Information:** Please confirm that iBank will provide `accountNumber` and `clientNumber` from the logged-in user's session and append them to the iframe URL as outlined above.

Next Steps

1. Please confirm the approach above
2. Let us know your preferences for the questions above
3. We'll provide the client credentials once ready
4. Coordinate testing once both sides are implemented

We're ready to proceed once we have your confirmation. Please let me know if you have any questions or concerns.
