

readme.md for @apiclient.xyz/tink

an unofficial api abstraction for tink.com

Install

To install @apiclient.xyz/tink, use either `npm` or `yarn` as per your preference. Run the following command in your project directory:

```
npm install @apiclient.xyz/tink --save
```

Or if you prefer yarn, use:

```
yarn add @apiclient.xyz/tink
```

This will add the package to your project's dependencies.

Usage

The `@apiclient.xyz/tink` library provides an unofficial TypeScript API abstraction for interacting with tink.com. To make the most out of this library, it's recommended to have a basic understanding of TypeScript and asynchronous programming in JavaScript.

Getting Started

First, you need to import the necessary classes from the `@apiclient.xyz/tink` package. Here's how you can get started with creating a `TinkAccount` instance and utilizing it to perform operations:

```
import { TinkAccount, TinkUser, TinkProviderConsent } from '@apiclient.xyz/tink';

// Initialize a new TinkAccount with your clientId and clientSecret
const myTinkAccount = new TinkAccount('<YOUR_CLIENT_ID>', '<YOUR_CLIENT_SECRET>');
```

Creating and Managing Tink Users

To interact with user-specific APIs, create a `TinkUser` object. You can also retrieve an already existing user or create a new one as follows:

```
// Creating a new Tink User
const newUser = await myTinkAccount.createTinkUser('unique_external_user_id');
console.log('New TinkUser created:', newUser);

// Retrieving an existing Tink User
const existingUser = await myTinkAccount.getTinkUser('existing_unique_external_user_id');
console.log('Existing TinkUser retrieved:', existingUser);
```

Generating Tink Link for Account Connection

To connect bank accounts to the Tink platform, you will need to generate a Tink Link URL and present it to your users. Here's how you can obtain it:

```
const tinkLinkUrl = await existingUser.getTinkLinkForMarket({
  countryId: 'SE', // Specify the market code, e.g., SE for Sweden
  redirectUrl: 'https://yourapplication.com/callback', // Redirect URL after account
  connection
  customState: 'your_custom_state', // Optional custom state to be included in the callback
  testProviderBool: true // Set to true if using test providers
});
console.log('Tink Link URL:', tinkLinkUrl);
```

Fetching Bank Accounts and Transactions

After the user has connected their bank accounts through the Tink Link, you can retrieve the connected accounts and their transactions:

```
// Fetching provider consents to get bank accounts
const providerConsents = await existingUser.getProviderConsents();
for (const consent of providerConsents) {
  console.log('Provider consent:', consent);

  // For each consent, get associated bank accounts
  const bankAccounts = await consent.getBankAccounts();
  for (const account of bankAccounts) {
    console.log('Bank account:', account);
  }
}
```

```
// Fetch transactions for each bank account
const transactions = await account.getTransactions();
for (const transaction of transactions) {
  console.log('Transaction:', transaction);
}
}
```

Advanced Usage

The library offers advanced features like handling provider consents, refreshing data, and managing users. For instance, you can delete a user as follows:

```
await existingUser.delete();
console.log('User deleted successfully.');
```

To explore more about the library, including handling errors, refreshing tokens, and more nuanced data retrieval, consult the official API documentation provided by Tink.com and refer to the TypeScript definitions in this package.

Remember to secure your credentials and tokens appropriately and follow Tink's guidelines for data handling and user privacy.

License and Legal Information

This repository contains open-source code that is licensed under the MIT License. A copy of the MIT License can be found in the [license](#) file within this repository.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH and are not included within the scope of the MIT license granted herein. Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines, and any usage must be approved in writing by Task Venture Capital GmbH.

Company Information

Task Venture Capital GmbH

Registered at District court Bremen HRB 35230 HB, Germany

For any legal inquiries or if you require further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #7

Created 2026-03-28 10:48:12 UTC by foss.global Team

Updated 2026-03-28 12:13:22 UTC by foss.global Team