


Get started with Finxact Sandbox

Published on 07/24/24  10 min read

Start configuring and testing compelling financial products with Finxact Sandbox.

Sandbox is a pre-configured test environment allowing you to access Finxact's features in a secured, dedicated environment. Within Sandbox, you can start to design, configure, and test new financial product in a non-production environment. It comes pre-loaded with mock data, enabling you to test real-world use cases and simulate transactions using the [Finxact API](#). Scale and begin owning your financial product development roadmap by adding your team members to Sandbox.

Discover the full capabilities of Finxact Sandbox

Dedicated and secure environment: Develop and test [financial products](#) and experiences in a non-production environment for 30-days, giving you time to explore Finxact's capabilities.

Pre-configured setup: Start development immediately with pre-set foundational configurations that take as little effort as a few clicks.

Pre-loaded with mock data: Leverage retail and commercial customer [sample data](#), saving you setup time and effort in testing real-life use cases.

Unlimited product development with Launchpad: Create and refine as many banking products as needed, allowing for extensive experimentation and innovation.

Simulate real-world banking experiences: Use the [Finxact API](#) to test banking use cases with ready-to-use data, ensuring practical and effective solutions.

Collaborative team testing: Add team members to explore and validate functionalities together, enhancing productivity and team collaboration.

Four steps to deploying new financial products using Sandbox

1. **Request access:** Sign-up for a [Finxact.io account](#), and request access to a Sandbox environment within [Dashboard](#).
2. **Connect to the APIs:** Access your [API keys](#) in dashboard and connect to Sandbox. We recommend using [Postman](#), which takes a few seconds to set up.
3. **Configure your first product:** Use our accelerators and API collections to design and develop financial products.
4. **Simulate real life banking use cases:** Open accounts, simulate financial transactions, and test financial products without risk in Sandbox.

Sandbox API Quick Start

This guide will get you started by creating a financial product and opening an account using the Finxact API in just 20-minutes:

1. [Access the Sandbox API keys](#)
2. [Create a financial product](#)
3. [Retrieve a customer's information](#)
4. [Open a multi-position account](#)
5. [Fund the new account](#)
6. [Add a new savings position](#)



Don't have a Sandbox Environment? Get started by creating a [Finxact Account](#) and requesting a Sandbox.

Step 1 – Access Sandbox API Keys

You can access your Sandbox API keys (`clientID` and `clientSecret`) within **Dashboard**.

1. Access your sandbox API keys within the **My Account** page.
2. Select Download Postman Settings under My Environments.

https://images.prismic.io/finxact-learning/Zp6slx5LeNNTxYT2_ZpqmZx5LeNNTxUGI_Screenshot2024-07-19at12.44.16PM.png%3Fauto%3Dformat%2Ccompress.png?auto=format,compress

3. Upload the Sandbox Environment File to **Postman**.

4. Download the curated API collection below to complete the Sandbox exercise.

For more information about connecting to our API and Postman, reference the **Get Started with Finxact API** and **Authentication** document.

[Download - ZpqMWB5LeNNTxT8e_FinxactSandbox-APICollection.postman_collection.json](#)

Step 2 – Create a Financial Product

You can use the `/prod_bk` end-point to create a financial product, such as a savings, checking, or line of credit product, in Sandbox. The following API is pre-set to create a savings product.

1. Review the (`prod_bk`) endpoint and request body (provided below).
2. Replace "Base_Savings_Product" with a name for your savings product.
3. Select **Send** to execute the API request, then review the response body.

POST | `core/v1/prod_bk`

```

1 {
2   "assetClass": 1,
3   "assetId": "USD",
4   "ccyCode": "USD",
5   "components": [
6     {
7       "componentClass": "componentInt",
8       "componentName": "Savings_Interest",
9       "version": "1"
10    }
11  ],
12  "desc": "Savings Product",
13  "glCat": 2,
14  "glSetCode": "PerSavLess100k",
15  "ifxAcctType": "SDA",
16  "isRegD": false,
17  "name": "Base_Savings_Product",
18  "posnClass": "posn_dep",
19  "prodGroup": "SAV",
20  "prodSubType": "Personal",
21  "prodType": "Deposit",
22  "stmtFreq": "0MAE"
}

```



Financial products require certain pre-requisites to build in Finxact. Sandbox comes pre-configured as a test institution with the pre-requisites complete. For more information on the pre-requisites, reference the [/prod_bk](#) documentation.

You can create financial products within **Dashboard** by using the **Product LaunchPad** feature in addition to the [/prod_bk](#) API call. LaunchPad offers pre-configured products accelerators and configuration wizards to streamline the design and development of financial products.

Step 3 – Retrieve a Customer’s Information

Sandbox comes pre-loaded with test customer data to simulate real-life use-cases. You can access a test customer’s information, **Thomas Payne’s**, with the **/customer** end-point.

1. Review the (**/customer**) endpoint. Notice, the API’s header includes **query** parameters to access Thomas Payne’s **customerId** : 98309 and **customerGroup** : 2
2. Select **Send** to execute the API request: **GET | /core/v1/customer?customerId=98309&customerGroup=2**
3. Review the response body. You’ll see the response includes the Thomas’ (**party_person._id**) and (**customer.customerId**). Those values have been set as variables in Postman, so they can be referenced when creating the account in step 4.



Users cannot create new customers or organizations in the Sandbox. However, Sandbox provides **sample** retail and commercial banking customers and accounts to execute test cases and integrations with the **Finxact API**. You can view the test customers at the bottom of this document. To gain full access to create customers, please connect with the Finxact team.

Step 4 – Open a Multi-Position Account

Finxact’s flexible, multi-position account structure enables the user to open multiple positions under one unified account. In this step, you can open a new multi-position account with a savings position for your test customer, **Thomas Payne**.

1. Review the (**customer/linkAccount**) endpoint and body (provided below).
2. Update the product name placeholder with savings product name created in step 2.
3. Select **Send** to execute the API request.
4. Review the response body. The response includes **partyId** of the owner and account number and account group of the multi-position account.

POST | core/v1/customer/linkAccount

```
1 {
2   "customer": {
3     "customerId": "{.thomaspaynecustomerId}",
4     "customerGroup": "2"
5   },
6   "newAccounts": [
7     {
8       "Acct": {
9         "acct_bk": {
10          "desc": "Personal Savings Account",
11          "acctGroup": 3,
12          "acctType": "Single",
13          "acctTitle": "Thomas Payne",
14          "isBrokered": false,
15          "tmZoneCode": "usnyc",
16          "stmtFreq": "0MAE"
17        },
18        "posn_depDtl": [
19          {
20            "prodName": "Base_Savings_Product",
21            "posn_dep": {
22              "ccyCode": "USD",
23              "acctgSeg": {
24                "CostCenter": "350",
25                "Product": "20"
26              }
27            }
28          }
29        ]
30      },
31      "acctPartyRel": {
32        "relType": "1",
33        "partyRelDesc": "Primary"
34      }
35    }
36  ]
}
```

Step 5 – Fund the New Account

After you create Thomas' account, it is time to fund it. You can facilitate a cash deposit to fund the new savings position by using the `/trn` end-point.

1. Review the (`trn`) endpoint and body (provided below).
2. Replace the \$5,000 placeholder with a funding amount for the savings position.
3. Select **Send** to send the API request.
4. Review the response body. The response includes the unique order id.

POST | `core/v1/trn`

```
1 {
2   "mode": 3,
3   "trnCode": "514",
4   "network": "CORE",
5   "entries": [
6     {
7       "acctNbr": "{{.thomaspayneAcctNbr}}",
8       "posnAcctNbr": "{{.thomaspaynePosnAcctNbr}}",
9       "acctGroup": "{{.thomaspayneAcctGroup}}",
10      "trnAmt": 5000,
11      "isDr": false
12    },
13    {
14      "trnAmt": 100,
15      "isDr": true
16    }
17  ]
}
```



Finxact supports the leading **payment methods**, such as ACH and Wires. Sandbox is not connected to live payment rails, though comes pre-configured with payment networks **payment networks** and **transaction codes** to test various transaction types. For other account funding use-cases, access our **account funding tutorial**.

Step 6 – Add a New Savings Position

Over time a customer's needs may change, and they may require additional financial products. For instance, Thomas may need a new savings position with a goal to save for an upcoming vacation. The `/acct_bk` end-point can be used to open another savings position under Thomas' **multi-position account**:

1. Review the (`acct_bk`) endpoint and body (provided below). Notice the head is pre-set with Thomas' account group and account number of the previously opened account.
2. Update the product name with the savings product name you created in step 2.
3. Select **Send** to execute the API request.
4. Review the response body. The response includes `partyId` and the unique id generated for the linked position.

PUT | `/core/v1/acct_bk/{.thomaspayneAcctGroup}/{.thomaspayneAcctNbr}`

```
1 {
2   "posn_depDt1": [
3     {
4       "prodName": "Base_Savings_Product",
5       "posn_dep": {
6         "ccyCode": "USD",
7         "acctgSeg": {
8           "CostCenter": "350",
9           "Product": "20"
10        },
11        "posnName": "Vacation Savings Position",
12        "tmZoneCode": "usnyc",
13        "_class": "posn_dep"
14      }
15    }
16  ]
}
```


As a final step, you can transfer funds from Thomas' original savings position to his new savings position.

1. Review the (`trn`) endpoint and body (provided below).
2. Replace the \$1,000 placeholder with a funding amount for the savings position.
3. Select **Send** to send the API request.
4. Review the response body. The response includes the unique order id.

POST | `core/v1/order`

```
1 {
2   "orderType": 1,
3   "orderAmt": 1000,
4   "network": "CORE",
5   "originator": {
6     "acctNbr": "{{.thomaspayneAcctNbr}}",
7     "posnAcctNbr": "{{.thomaspaynePosnAcctNbr}}",
8     "acctGroup": "{{.thomaspayneAcctGroup}}"
9   },
10  "counterParty": {
11    "acctNbr": "{{.thomaspayneAcctNbr}}",
12    "posnAcctNbr": "{{.thomaspaynePosnAcctNbr2}}",
13    "acctGroup": "{{.thomaspayneAcctGroup}}"
14  }
}
```

Additional Use-Cases

You can continue to explore Finxact's capabilities, build products, and interact with the APIs in Sandbox. To learn more about additional use-cases, check-out the [tutorials](#) and the pre-loaded test data below.

Sandbox Test Data

Sandbox comes preloaded with test data to get you started. The test data includes a collection of fictitious retail and commercial banking customers and accounts to execute test cases and integrations with the Finxact API.



Sandbox is a non-production level environment that is intended execute test scenarios with fictions data. It is not intended to create new, real customers or for any performance testing. For more about Finxact's scalability, reference our [recent report](#).

Retail Customer Test Data

Customer Name	CustomerID	CustomerGroup
Maria Alvarez	98314	2
Pedro Alvarez	98313	2
Marvin Higgins	8421001	2
Anjit Mahajan	8421002	2
Thomas Payne	98309	2
Louise Payne	98310	2
Tyrice Washington	98311	2

Commercial Customer Test Data

Business Name	CustomerID	CustomerGroup	Type
Cold Brew Coffee Company	1002020311	2	Sole Proprietorship
Gleichner-Sawayn	1002020301	2	Corporation
Grady PLC	1002020303	2	Corporation
Willms Group	1002020304	2	Corporation
Von Group	1002020312	2	Limited Liability Company

Discover learning content designed for you

Enroll to access customized docs

Enroll now

Log in