## **Overview**

Finzly Connect is a universal payment API for developers to access all the payment rails without stressing over the network and messaging rules. Finzly Connects offers REST API and webhook notifications to access any Finzly-powered bank. Finzly has standardized the payment data and made the payment processing super simple, so a bank customer, whether a fintech, business or an individual, doesn't have to worry about the messaging rules. A developer can access ACH, Domestic Wires, International Wires, RTP, and FedNow using Finzly Connect and connect to their favorite Finzly Bank.

Finzly Connect offers instant payment access and notifications via simple REST API. It is designed to be simple to use and implement in any model computer language that allows you to generate web requests. It uses "REST-Like '' RPC-style operations with parameters URL encoded into the request, and its response is encoded in JSON.

In this section, we will give a brief overview of the Finzly APIs: OpenBanking API, UserAdmin API and PaymentStatusNotifications API. We'll get as much information as possible from the PDF and add some unique content.

This page will also have a menu, as shown below. Using this menu, readers can easily access a specific section of the API documentation.

### **Finzly APIs**

| [Learn About Finzly APIs](#_ux9uqmhg9s92) |
| --- |

### **Guides**

| [Check out the guides on how to use Finzly APIs](#_embqdcnhun5t) |
| --- |

### 

### **Authentication**

| [Create an API access token and authenticate your API requests](#_wa9nflgiwkrp) |
| --- |

### **Common Workflows**

| [Learn about common workflows](#_peo6vn933drm) |
| --- |

### **Errors**

| [Learn about API errors, how they’re structured and how to handle them.](#_bw1qz2hirheo) |
| --- |

### **Payment Networks**

| [Find out the different payment networks Finzly APIs support](#_mxbyhhmelu6c) |
| --- |

### **Webhooks**

| [Learn how to set up a webhook](#_js5w4gtmvay4) |
| --- |

### **Frequently Asked Questions**

| [Check out the frequently asked questions related to Finzly APIs](#_x5wwomgjgsio) |
| --- |

### **Changelog & Feedback**

| **API Changelog**  [Find out about what changed with the API](#_1b3io7e17du0) | **Feedback**  Share your feedback to improve the API and ensure it fits your needs. Feel free to send your queries & feedback to [connect@finzly.com](mailto:connect@finzly.com) |
| --- | --- |

## 

## **Finzly APIs**

Finzly provides a modular platform that enables businesses, developers, and fintech to consume our APIs directly through banks or embed them into their products.

Users can launch financial products and services via API integration into our FinzlyOS platform.

Our API consists of modern RESTful messages to connect and communicate with our FinzlyOS platform. For every message, the platform returns a REST response.

Webhooks are sent in response to third-party actions.

## **Open Banking API v2.1.0**

## 

## **Customers APIs**

Customer APIs entitle the downstream partners, Banks, and associates to formally onboard clients with all the required information to perform financial operations.

These APIs will also enable users to search & maintain these onboarded clients, update any modifications in client information and create/update their beneficiaries (contacts).

Following use cases helps you to understand the current customer capabilities exposed via APIs

1. Search Customer
2. Create Customer
3. Get Customer By CustomerUID
4. Update Customer

**Add Client (of Fintech)**

**Description -** Using this API, the user can add a CRM entity to FinzlyOS. This API allows them to create customer types such as Customer, Corporate, Downstream partner (FinTech), Third Party Customer (Customer of a FinTech), and Financial Institutions.

**HTTP Method:** POST

### **Supported API functions**

### **Search Customers**

**GET /v2/customers**

**HTTP Method: GET**

**Description** - A GET request containing a series of parameters and objects to the GET /v2/customers endpoint is required to search for customers. Sending a GET request to the Search Customers Endpoint will allow users to search for customers using parameters such as legal name, short name, etc. All parameters and objects are specified below:

**Request Parameters**

Page and Limits are the mandatory parameters that provide the list of Corporate Entities, like how many records we want to get through the API request. Here are some of the other optional Parameters which can also be passed as per the user requirements

| Name | In | Data Type | Required | Description |
| --- | --- | --- | --- | --- |
| legalName | query | string | false | Customer's legal name |
| customerType | query | string | false | Type of Customer |
| shortName | query | string | false | Customer’s short name |
| customerId | query | string | false | Customer ID |
| status | query | string | false | none |
| taxId | query | string | false | Federal Tax ID |
| achCompanyID | query | string | false | ACH company ID is a 10-digit identifier used by banks and Nacha, the operator of the ACH network, to identify the entity collecting payments or sending money via ACH (also referred to as an “originator”) |
| page | query | string | true | none |
| limit | query | string | true | none |

**JSON Response** for GET request to **/v2/customers** endpoint

A successful GET request to **/v2/customers** endpoint will return the following data -

**Response Code - 200**

| {  "externalReferenceId": "tran12780",  "legalName": "DEMO Bank",  "shortName": "DMOB",  "customerUID": 111222,  "customerType": "Consumer or Corporate or Downstream Partner or Processing Org or Financial Institution",  "taxId": "123456789",  "achCompanyID": "1234567899",  "customerId": "12347890",  "status": "Active, Suspended,New,Pending",  "lei": "ABC1234",  "swiftcode": "BOFAUS3N",  "language": "English",  "emailAddress": "abc@bank.com",  "phoneNumber": "1112223333",  "faxNumber": "1231249877",  "website": "http://www.bank.com",  "logoUrl": "https://www.google.com/images/branding/googlelogo/2x/googlelogo\_color\_92x30dp.png",  "linkedinUrl": "https://www.linkedin.com/company/finzly",  "twitterUrl": "https://twitter.com/bank",  "legalAddress": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "mailAddressSameAsLegal": true,  "mailingAddress": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "relationshipType": "Customer or Bank or ThirdParty or Self",  "partnerOrgShortName": "TESTBANK",  "processingOrgShortName": "POBANK",  "costCenter": "123123121",  "relationshipManager": "testuser",  "originatingOfficer": "testuser",  "parentEntityShortName": "TESTBANK",  "addtionalProfileDetails": [  {  "creditclient": true,  "accountanalysis": true,  "sso": true,  "billingcustomer": true,  "restrictedaccess": true,  "achPositivePay": true,  "achWhitelist": true,  "achNonPrefund": true,  "achNonPrefundLimit": "string",  "paymentFileAuthentication": true,  "detailsofCharge": "OUR",  "fxPricingTier": "string",  "crediValueAdjustment": "string"  }  ],  "secCodes": [  "WEB"  ],  "note": "Notes for the customer",  "legalEntityAccessSettings": {  "allowOnlineAccess": true,  "userPinForNewEntry": true,  "userDualApproval": true,  "userPinForApproval": true,  "notificationsDualApproval": true,  "customerPinForNewEntry": true,  "customerDualApproval": true,  "customerPinForApproval": true,  "benePinForNewEntry": true,  "beneDualApproval": true,  "benePinForApproval": true,  "paymentPinForNewEntry": true,  "paymentLimitforFirstApprover": 0,  "paymentDualApproval": true,  "paymentLimitforSecondApprover": 0,  "paymentNumberOfApprovers": 0,  "paymentPinForApproval": true,  "feeTierForApproval": true  } } |
| --- |

**API Response Objects-**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| pagination | [Pagination](https://apidocs.finzly.net/dashboard?java=#schemapagination) | false | none | none |
| data | [[GetCustomerDetailResponse](https://apidocs.finzly.net/dashboard?java=#schemagetcustomerdetailresponse)] | false | none | none |
|  |  |  |  |  |

**Get Customer Detail Response**

| {  "externalReferenceId": "tran12780",  "legalName": "DEMO Bank",  "shortName": "DMOB",  "customerUID": 111222,  "customerType": "Consumer or Corporate or Downstream Partner or Processing Org or Financial Institution",  "taxId": "123456789",  "achCompanyID": "1234567899",  "customerId": "12347890",  "status": "Active, Suspended,New,Pending",  "lei": "ABC1234",  "swiftcode": "BOFAUS3N",  "language": "English",  "emailAddress": "abc@bank.com",  "phoneNumber": "1112223333",  "faxNumber": "1231249877",  "website": "http://www.bank.com",  "logoUrl": "https://www.google.com/images/branding/googlelogo/2x/googlelogo\_color\_92x30dp.png",  "linkedinUrl": "https://www.linkedin.com/company/finzly",  "twitterUrl": "https://twitter.com/bank",  "legalAddress": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "mailAddressSameAsLegal": true,  "mailingAddress": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "relationshipType": "Customer or Bank or ThirdParty or Self",  "partnerOrgShortName": "TESTBANK",  "processingOrgShortName": "POBANK",  "costCenter": "123123121",  "relationshipManager": "testuser",  "originatingOfficer": "testuser",  "parentEntityShortName": "TESTBANK",  "addtionalProfileDetails": [  {  "creditclient": true,  "accountanalysis": true,  "sso": true,  "billingcustomer": true,  "restrictedaccess": true,  "achPositivePay": true,  "achWhitelist": true,  "achNonPrefund": true,  "achNonPrefundLimit": "string",  "paymentFileAuthentication": true,  "detailsofCharge": "OUR",  "fxPricingTier": "string",  "crediValueAdjustment": "string"  }  ],  "secCodes": [  "WEB"  ],  "note": "Notes for the customer",  "legalEntityAccessSettings": {  "allowOnlineAccess": true,  "userPinForNewEntry": true,  "userDualApproval": true,  "userPinForApproval": true,  "notificationsDualApproval": true,  "customerPinForNewEntry": true,  "customerDualApproval": true,  "customerPinForApproval": true,  "benePinForNewEntry": true,  "beneDualApproval": true,  "benePinForApproval": true,  "paymentPinForNewEntry": true,  "paymentLimitforFirstApprover": 0,  "paymentDualApproval": true,  "paymentLimitforSecondApprover": 0,  "paymentNumberOfApprovers": 0,  "paymentPinForApproval": true,  "feeTierForApproval": true  } } |
| --- |

**Data Returned Objects**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| externalReferenceId | string | false | none | Unique reference id from a system outside of finzly. The external reference id can be used by the finzly for the request tracing purpose (if needed). |
| legalName | string | false | none | Customer's legal name |
| shortName | string | false | none | Customer's short name |
| customerUID | string | false | none | The unique identifier associated with the customer record assigned by the Finzly system. |
| customerType | string | false | none | Type of the customers in the Finzly CRM system |
| taxId | string | false | none | A tax id that the US government uses to identify business entities located in the US number |
| achCompanyID | string | false | none | ACH company ID is a 10-digit identifier used by banks and Nacha, the operator of the ACH network, to identify the entity collecting payments or sending money via ACH (also referred to as an “originator”) |
| customerId | string | false | none | Customer id assigned by the user. |
| status | string | false | none | Current status of the customer. |
| lei | string | false | none | Legal Entity Identifier (LEI) associated with the customer |
| swiftcode | string | false | none | SWIFT code that identifies your bank, country, location, and branch |
| language | string | false | none | language |
| emailAddress | string | false | none | Customer email address |
| phoneNumber | string | false | none | Customer phone number |
| faxNumber | string | false | none | Customer fax number |
| website | string | false | none | Customer fax number |
| logoUrl | string | false | none | Customer logo url |
| linkedinUrl | string | false | none | Company linkedin URL |
| twitterUrl | string | false | none | Customer twitter url |
| legalAddress | [CustomerAddress](https://apidocs.finzly.net/dashboard?java=#schemacustomeraddress) | false | none | none |
| mailAddressSameAsLegal | boolean | false | none | none |
| mailingAddress | [CustomerAddress](https://apidocs.finzly.net/dashboard?java=#schemacustomeraddress) | false | none | none |
| relationshipType | string | false | none | Relationship Type |
| partnerOrgShortName | string | false | none | This is the shortname associated with the customer defined in the CRM system as 'Downstream Partner'. This is required only if the customer relationshipType is Third party |
| processingOrgShortName | string | false | none | This is the shortname associated with the processing org the customer is associated with. |
| costCenter | string | false | none | Cost Center associated with the customer entity |
| relationshipManager | string | false | none | Relationship manager name. It needs to be a valid user login id in the finzly system. |
| originatingOfficer | string | false | none | Originating Officer name. It needs to be a valid user login id in the finzly system. |
| parentEntityShortName | string | false | none | This is the short name associated with the existing customer legal entity act as a parent entity |
| addtionalProfileDetails | [[CustomerAdditonalProfileDetails](https://apidocs.finzly.net/dashboard?java=#schemacustomeradditonalprofiledetails)] | false | none | none |
| secCodes | string | false | none | Provide all valid SEC-Standard Entry Class codes allowed for the given customer entity |
| note | string | false | none | Free form text to leave any notes for the customer entity. |
| legalEntityAccessSettings | [LegalEntityAccessSettings](https://apidocs.finzly.net/dashboard?java=#schemalegalentityaccesssettings) | false | none | none |

*Note the following about the results:*

* *Request URL: ​Shows what executed.*
* *Response Headers:​ Contains messages about the success or failure of the request.*
* *Response Body: ​Contains a JSON array of the results.*
* *Curl Command Line: ​Contains a form of the request that you can copy and paste to execute at the command line or in scripts.*

### **Creating Customer**

**HTTP Method:** **POST**

**POST /v2/customers**

**Description -** Sending a POST request to Customers Endpoint will allow the users to create a customer as a Corporate Account for getting the features and using further options. Finzy is providing a dashboard where corporate customers can create their account and use the settings later on as per their convenience. In order to create a customer a POST request containing a series of parameters and objects to the POST /v2/customers endpoint is required. All parameters and objects are specified below -

**Request Parameters**

The entire data will be passed through the body section.

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| body | body | [CreateCustomerRequest](https://apidocs.finzly.net/dashboard?java#schemacreatecustomerrequest) | true | Customer Details |

**JSON Response** for POST request to **/v2/customers** endpoint will return the following data -

**Response Code - 200**

| 200 Response {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to create a customer due to the mandatory attribute is not provide.",  "data": {  "customerUID": "123",  "externalReferenceId": "string",  "shortName": "string",  "customerEntityStatus": "New"  } } |
| --- |

**Data Returned Objects**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [CreateUpdateCustomerResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemacreateupdatecustomerresponse_data) | false | none | none |

**Note the following about the results:**

* **Request URL: ​Shows what executed.**
* **Response Headers:​ Contains messages about the success or failure of the request.**
* **Response Body: ​Contains a JSON array of the results.**
* **Curl Command Line: ​Contains a form of the request that you can copy and paste to execute at the command line or in scripts.**

### **Getting Customer Information by customerUID**

**GET /v2/customers/{customerUID}**

**HTTP Method: GET**

**Description** - A GET request containing customerUID to the /v2/customers/**{customerUID}** endpoint is required to retrieve information related to customers. Sending a GET request to this Endpoint allows users to retrieve the customer information from the system using the unique identifier associated with the customer. **By passing the Customer ID in the path of the API request, we can get the full details in the response.**

**Request Parameters**

**Here we need to pass the CustomerID as a Path in the API request to get the response.**

| Name | In | Data Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true | none |

**Success Response**

A successful GET request to /v2/customers/**{customerUID}** endpoint returns the following data

**Response Code - 200**

| 200 Response [  {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to get the customer: ",  "pagination": {  "totalRecords": 100,  "returnedRecords": 10,  "pageReturned": 2,  "pageSize": 10  },  "data": [  {  "externalReferenceId": "tran12780",  "legalName": "DEMO Bank",  "shortName": "DMOB",  "customerUID": 111222,  "customerType": "Consumer or Corporate or Downstream Partner or Processing Org or Financial Institution",  "taxId": "123456789",  "achCompanyID": "1234567899",  "customerId": "12347890",  "status": "Active, Suspended,New,Pending",  "lei": "ABC1234",  "swiftcode": "BOFAUS3N",  "language": "English",  "emailAddress": "abc@bank.com",  "phoneNumber": "1112223333",  "faxNumber": "1231249877",  "website": "http://www.bank.com",  "logoUrl": "https://www.google.com/images/branding/googlelogo/2x/googlelogo\_color\_92x30dp.png",  "linkedinUrl": "https://www.linkedin.com/company/finzly",  "twitterUrl": "https://twitter.com/bank",  "legalAddress": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "mailAddressSameAsLegal": true,  "mailingAddress": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "relationshipType": "Customer or Bank or ThirdParty or Self",  "partnerOrgShortName": "TESTBANK",  "processingOrgShortName": "POBANK",  "costCenter": "123123121",  "relationshipManager": "testuser",  "originatingOfficer": "testuser",  "parentEntityShortName": "TESTBANK",  "addtionalProfileDetails": [  {  "creditclient": true,  "accountanalysis": true,  "sso": true,  "billingcustomer": true,  "restrictedaccess": true,  "achPositivePay": true,  "achWhitelist": true,  "achNonPrefund": true,  "achNonPrefundLimit": "string",  "paymentFileAuthentication": true,  "detailsofCharge": "OUR",  "fxPricingTier": "string",  "crediValueAdjustment": "string"  }  ],  "secCodes": [  "WEB"  ],  "note": "Notes for the customer",  "legalEntityAccessSettings": {  "allowOnlineAccess": true,  "userPinForNewEntry": true,  "userDualApproval": true,  "userPinForApproval": true,  "notificationsDualApproval": true,  "customerPinForNewEntry": true,  "customerDualApproval": true,  "customerPinForApproval": true,  "benePinForNewEntry": true,  "beneDualApproval": true,  "benePinForApproval": true,  "paymentPinForNewEntry": true,  "paymentLimitforFirstApprover": 0,  "paymentDualApproval": true,  "paymentLimitforSecondApprover": 0,  "paymentNumberOfApprovers": 0,  "paymentPinForApproval": true,  "feeTierForApproval": true  }  }  ]  } |
| --- |

### 

### **Updating Customers**

**PUT /v2/customers/{customerUID}**

**HTTP Method: PUT**

**Description -** A PUT request containing a series of parameters and objects to the v2/customers/{customerUID} endpoint is required to update customers. Sending a PUT request to the v2/customers/{customerUID} endpoint allows users to update information related to any existing customer type using parameters. All parameters and objects are specified below:

**Request Parameters**

**Here we need to Pass the Customer ID in the path of the API request and Data via the body section to update the record on behalf of the customerID.**

**nd data will be passed via the body section to update the record on the behalf of customerID.**

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true | none |
| body | body | [UpdateCustomerRequest](https://apidocs.finzly.net/dashboard#schemaupdatecustomerrequest) | true | Customer details |

**JSON Response**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to create a customer due to the mandatory attribute is not provide.",  "data": {  "customerUID": "123",  "externalReferenceId": "string",  "shortName": "string",  "customerEntityStatus": "New"  } } |
| --- |

| {  "customerUID": "123",  "externalReferenceId": "string",  "shortName": "string",  "customerEntityStatus": "New" } |
| --- |

**Data Returned Objects**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [CreateUpdateCustomerResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemacreateupdatecustomerresponse_data) | false | none | none |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| customerUID | string | false | none | Customer id within Finzly's CRM system |
| externalReferenceId | string | false | none | none |
| shortName | string | false | none | none |
| customerEntityStatus | string | false | none | Status of the customer |

**Add Customer Account**

This API allows the user to add a customer account to the customer profile in the CRM system. The user can use the account unique identifier (UID) in the payment request. The customer account can be a type of Checking, Saving or Virtual (Fintech account).

Customer Contacts

Customer Contact APIs help manage the customer beneficiaries (contacts) in the Finzly CRM system.

Once the contacts are defined as the template contact, UID can be used in the payment APIs as a receiverAccountId.

Following use cases helps you to understand the current customer contacts capabilities exposed via APIs

1. Search Customer Contact
2. Create Customer Contact
3. Get Customer Contacts By ContactUID
4. Update Customer Contact

### **Search Customer Contacts**

**GET /v2/customers/{customerUID}/contacts**

**HTTP Method: GET**

**Description -** A **GET** request containing a series of parameters and objects to the v2/customers/{customerUID}/contacts endpoint is required to search customer contacts. Sending a GET request to the **/v2/customers/{customerUID}/contacts** endpointallows Corporate Entities acting as customers to get the list of the details to those customers who are linked with the customerID. All parameters and objects are specified below:

**Request Parameters -**

**Here, the customerUID, Page and Limit are mandatory parameters. CustomerUID will be passed via the PATH of the API request and Page and Limit will be passed via the query section of the API request to fetch the information.**

| **Name** | **In** | **Data Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| customerUID | path | integer | true | none |
| contactName | path | string | false | none |
| accountNumber | query | string | false | none |
| page | query | string | true | none |
| limit | query | string | true | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to get a customer contact: ",  "data": {  "externalReferenceId": "string",  "templateName": "string",  "name": "string",  "address": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "emailAddress": "abc@abc.com",  "phoneNumber": "123-124-9877",  "contactBank": {  "country": "string",  "accountNo": "string",  "accountCcy": "string",  "name": "string",  "address": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "nationalCode": "0125487",  "swiftCode": "USBNX000"  },  "intermediaryBank": {  "country": "string",  "name": "string",  "nationalCode": "string",  "swiftCode": "string",  "coverMessageRequired": "string",  "address": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  }  },  "otherInfo": {  "beneficiaryNotes": "string",  "otherBeneBankInfo": "string"  },  "regulartoryReporting": [  {  "key": "string",  "value": "string"  }  ]  } } |
| --- |

**Data Returned Objects**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| pagination | Pagination | false | none | none |
| data | [[CustomerContact](https://apidocs.finzly.net/dashboard?java=#schemacustomercontact)] | false | none | none |

### **Creating Customer Contact**

POST /v2/customers/{customerUID}/contacts

**HTTP Method: POST**

**Description -** Sending a **POST** request to **v2/customers/{customerUID}/contacts** endpoint allows users to create a new contact for the existing customer. Corporate entities acting as customers can create a Customer, so thus in the future, they don't need to enter the details for payment/transaction purposes. It is a one-time registration for each corporate entity. All parameters and objects are specified below:

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| customerUID | path | integer | true | none |
| body | body | [CustomerContact](https://apidocs.finzly.net/dashboard#schemacustomercontact) | true | Account Info |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to create a customer contact: ",  "data": {  "customerContactUID": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [CreateCustomerContactResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemacreatecustomercontactresponse_data) | false | none | none |

### **Get Customer Contact Details**

GET /v2/customers/{customerUID}/contacts/{contactUID}

**HTTP Method: GET**

**Description -** A GET request containing contactUID to the /v2/customers/{customerUID}/contacts/{contactUID} endpoint is required to retrieve information related to customers. Sending a GET request to this Endpoint allows users to retrieve the customer information from the system using the unique identifier associated with the customer's contact.

**Parameters**

In this Query, we need to pass the two Parameters, CustomerID and ContactID, in the API’s Path to get the response of particular contacts.

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | integer | true | none |
| contactUID | path | [CustomerContact](https://apidocs.finzly.net/dashboard#schemacustomercontact) | true | none |

| {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to get a customer contact: ",  "data": {  "externalReferenceId": "string",  "templateName": "string",  "name": "string",  "address": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "emailAddress": "abc@abc.com",  "phoneNumber": "123-124-9877",  "contactBank": {  "country": "string",  "accountNo": "string",  "accountCcy": "string",  "name": "string",  "address": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  },  "nationalCode": "0125487",  "swiftCode": "USBNX000"  },  "intermediaryBank": {  "country": "string",  "name": "string",  "nationalCode": "string",  "swiftCode": "string",  "coverMessageRequired": "string",  "address": {  "addressLine1": "111 My Street",  "addressLine2": "Suite 210",  "city": "Charlotte",  "state": "North Carolina",  "postalCode": "28269",  "countryCode": "USA",  "country": "UNITED STATES OF AMERICA"  }  },  "otherInfo": {  "beneficiaryNotes": "string",  "otherBeneBankInfo": "string"  },  "regulartoryReporting": [  {  "key": "string",  "value": "string"  }  ]  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | CustomerContact | false | none | none |

### **Update Customer Contact Details**

**PUT /v2/customers/{customerUID}/contacts/{contactUID}**

**HTTP Method: PUT**

**Description -** Sending a PUT request to v2/customers/{customerUID}/contacts/{contactUID} allows users to update customer contact details. All parameters and objects are specified below:

**Request Parameters**

**Here we also need to pass two Parameters, customerUID, and contactUID, in the Path and the further account information for which they want to update the record in the Body section.**

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true | none |
| contactUID | path | string | true | none |
| body | body | [CustomerContact](https://apidocs.finzly.net/dashboard#schemacustomercontact) | true | Account Information |

**JSON Response**

| {  "status": "Success or Failure",  "code": "CUST001 etc.",  "message": "Failed to create a customer contact: ",  "data": {  "customerContactUID": "string",  "externalReferenceId": "string"  } |
| --- |

}

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | CreateCustomerContactResponse\_data | false | none | none |

Customer Bank Account

# Customer Accounts

Customer Accounts APIs help users get all the accounts associated with their customer in a single call. You can enhance the customer experience by providing them with the most updated balance. Besides, you can further add value by tracking payment transactions related to these individual accounts.

The following **supported API functions** helps you to understand the current customer accounts capabilities exposed via APIs:

1. Search Customer Account
2. Create Customer Account
3. Get Customer Account
4. Update Customer Account
5. Delete Customer Account

### **Supported Customer Accounts API functions**

### **Search Customer Account**

**HTTP Method: GET**

**Description** - Sending a GET request to the Search Customers Account Endpoint will allow users to search for customer accounts using parameters customerUID, accountType, and others. This can be used to search the details of customers' accounts registered with the corporate entity. The details will be provided in bulk and not specific. All the request parameters and objects are specified below:

**Request Parameters**

Customer UID must be passed in the path of the API request to search for customers. Rest of the parameters are optional

| Name | In | Data Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true |  |
| accountType | query | string | false | Type of Customer Account |
| status | query | string | false |  |
| currency | query | string | false | Customer ID |
| wireEnabled | query | string | false | none |
| accountNumber | query | string | false | Federal Tax ID |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "ACC001 etc.",  "message": "Failed to get an account: ",  "data": [  {  "accountUID": "100100",  "name": "Checking",  "accountNumber": "1122345",  "fboAccountNumber": "2233445",  "type": "DEPOSIT or LOAN",  "subType": "Checking, Saving, Psuedo",  "systemOfRecord": "Other Core or External",  "balanceType": "DDA etc.",  "ledgerBalance": "24000.00",  "availableBalance": "25000.00",  "availableBalanceStatus": "NSF",  "amountPrecision": "string",  "currency": "USD",  "country": "UNITED STATE OF AMERICA etc.",  "bankId": "11223498",  "bankName": "Chase etc.",  "bankIdType": "By ABA or By Address",  "enableWires": false,  "enableOffset": false,  "defaultOffset": false,  "feeAccount": false,  "returnAccount": false,  "status": "Active etc.",  "externalAccount": {  "externalProvider": "Plaid",  "externalId": "1234567",  "externalAuthToken": "1234567"  }  }  ] |
| --- |

**}**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [[CustomerBankAccount](https://apidocs.finzly.net/dashboard?java=#schemacustomerbankaccount)] | false | none | none |

**Customer Bank Account**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| accountUID | string | false | none | Account unique identifier assigned by the system |
| name | string | false | none | Account Name |
| accountNumber | string | false | none | Account Number |
| fboAccountNumber | string | false | none | FBO Account Number |
| type | string | false | none | Account Type |
| subType | string | false | none | Account Type |
| systemOfRecord | string | false | none | Identifier of the system of record |
| balanceType | string | false | none | Account balance type in the bank core system |
| ledgerBalance | string | false | none | Account ledger balance amount in the bank core system |
| availableBalance | string | false | none | Account balance amount in the bank core system |
| availableBalanceStatus | string | false | none | Account balance status in the bank core system |
| amountPrecision | string | false | none | Amount amountPrecision |
| currency | string | false | none | Currency code is ISO format |
| country | string | false | none | Country name |
| bankId | string | false | none | Bank routing number |
| bankName | string | false | none | Name of the bank associated with the bankId |
| bankIdType | string | false | none | Bank identifier type |
| enableWires | boolean | false | none | Wire status |
| enableOffset | boolean | false | none | Whether the accountNumber is a Offset account |
| defaultOffset | boolean | false | none | Default offset account |
| feeAccount | boolean | false | none | Whether the accountNumber is a Fee account |
| returnAccount | boolean | false | none | Whether the accountNumber is a Return account |
| status | string | false | none | Account Status |
| externalAccount | [AccountExternal](https://apidocs.finzly.net/dashboard?java=#schemaaccountexternal) | false | none | none |

### **Create Customer Account**

**POST /v2/customers/{customerUID}/accounts**

**HTTP Method: POST**

**Description -** Sending a POST request to /v2/customers/{customerUID}/accounts endpoint allows users to create a customer account using parameters including customerUID. All parameters and objects are specified below:

**Request Parameters**

**The customer uid must be passed in the path of API request. In the body section must pass the details of the user account to be createds**

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | integer | true | none |
| body | body | [CreateAccountRequest](https://apidocs.finzly.net/dashboard#schemacreateaccountrequest) | true | Account Info |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "ACC001 etc.",  "message": "Failed to create an account: ",  "data": {  "accountUID": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [CreateAccountResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemacreateaccountresponse_data) | false | none | none |

**Create Account Response Data**

| {  "accountUID": "string",  "externalReferenceId": "string" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| accountUUID | string | false | none | none |
| externalReferenceId | string | false | none | none |

### **Get Customers**

**GET /v2/customers/{customerUID}/accounts/{accountUID}**

**HTTP Method: GET**

**Description -** Sending a GET request to /v2/customers/{customerUID}/accounts/{accountUID} endpoint allows users to retrieve a customer account associated with parameters including customerUID and accountUID. All parameters and objects are specified below -

**Request Parameters**

**Two parameters need to be passed in the path of the API request, customerUID and account UID of the user registered with customer/corporate.**

| **Name** | **In** | **Data Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true | none |
| accountUID | path | string | true | none |
| includeAccountBalance | query | string | false | none |
| page | query | string | false | none |
| limit | query | string | false | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "ACC001 etc.",  "message": "Failed to get an account: ",  "data": {  "accountUID": "100100",  "name": "Checking",  "accountNumber": "1122345",  "fboAccountNumber": "2233445",  "type": "DEPOSIT or LOAN",  "subType": "Checking, Saving, Psuedo",  "systemOfRecord": "Other Core or External",  "balanceType": "DDA etc.",  "ledgerBalance": "24000.00",  "availableBalance": "25000.00",  "availableBalanceStatus": "NSF",  "amountPrecision": "string",  "currency": "USD",  "country": "UNITED STATE OF AMERICA etc.",  "bankId": "11223498",  "bankName": "Chase etc.",  "bankIdType": "By ABA or By Address",  "enableWires": false,  "enableOffset": false,  "defaultOffset": false,  "feeAccount": false,  "returnAccount": false,  "status": "Active etc.",  "externalAccount": {  "externalProvider": "Plaid",  "externalId": "1234567",  "externalAuthToken": "1234567"  }  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [CustomerBankAccount](https://apidocs.finzly.net/dashboard?java=#schemacustomerbankaccount) | false | none | none |

### 

### **Update Customer Account**

**PUT /v2/customers/{customerUID}/accounts/{accountUID}**

**HTTP Method: PUT**

**Description -** Sending a PUT request to v2/customers/{customerUID}/accounts/{accountUID} endpoint with parameters including customerUID and accountUID allows users to update customer accounts. All parameters and objects are specified below:

**Request Parameters**

**Two parameters need to be passed in the path of the API request, customerUID and account UID of the user registered with customer/corporate. Body section will include the details of the user account which we want to update.**

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true | none |
| accountUID | path | string | true | none |
| body | body | [UpdateAccountRequest](https://apidocs.finzly.net/dashboard#schemaupdateaccountrequest) | true | Account Info |

**JSON Response**

A successful request to this endpoint returns the following data -

Response Code - 200

| {  "status": "Success or Failure",  "code": "ACC001 etc.",  "message": "Failed to update an account: ",  "data": {  "accountUID": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [UpdateAccountResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemaupdateaccountresponse_data) | false | none | none |

**Update Account Response data**

| {  "accountUID": "string",  "externalReferenceId": "string" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| accountUUID | string | false | none | Account unique identifier assigned by the system |
| externalReferenceId | string | false | none | none |

### **Delete Customer Account**

**DELETE /v2/customers/{customerUID}/accounts/{accountUID}**

**HTTP Method: DELETE**

**Description - Sending a DELETE request to the** /v2/customers/{customerUID}/accounts/{accountUID} endpoint with parameters including CustomerUID and AccountUID allows users to delete a customer account. All parameters and objects are specified below:

**Request Parameters**

**Two parameters need to be passed in the path of the API request, customerUID and account UID of the user registered with customer/corporate to delete the account.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| customerUID | path | string | true | none |
| accountUID | path | string | true | none |

**JSON Response**

**A successful DELETE request to this endpoint will return the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "ACC001 etc.",  "message": "Failed to update an account: ",  "data": {  "accountUID": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [UpdateAccountResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemaupdateaccountresponse_data) | false | none | none |

**Update Account Response data**

| {  "accountUID": "string",  "externalReferenceId": "string" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| accountUUID | string | false | none | Account unique identifier assigned by the system |
| externalReferenceId | string | false | none | none |

# Payments

Payments APIs enable third parties, banks, and downstream partners to initiate and manage customer payments.

The single payment API supports the payment origination for the various payment rails such as ACH, FedWire, SWIFT, and Instant Payments based on speed preference.

The following use cases help you to understand the current payment capabilities exposed via APIs.

1. Search Payments
2. Create Credit Payment
3. Create Debit Payment

### **Supported Customer Accounts API functions**

**Search Payments**

**POST /v3/payments/search**

**HTTP Method: GET**

**Description -** Sending a GET request to /v3/payments/search endpoint allows users to retrieve a list of payments. All the request parameters and objects are specified below:

**Request Parameters**

***No argument is required. Only payment details will be passed in the body of the API request.***

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [PaymentSearch](https://apidocs.finzly.net/dashboard?http=#schemapaymentsearch) | false | none |

**JSON Response**

| {  "status": "Success or Failure",  "code": "PAYMENT001 etc.",  "message": "Failed to search a payment: ",  "data": [  {  "paymentUID": 12341,  "counterPartyId": "string",  "senderUID": "string",  "senderAccountNumber": "string",  "senderAccountName": "string",  "senderName": "string",  "senderAccountType": "string",  "ultimateSenderName": "string",  "ultimateSenderAccountNumber": "string",  "ultimateSenderAddress1": "string",  "ultimateSenderAddress2": "string",  "ultimateSenderTaxId": "string",  "ultimateSenderCity": "string",  "ultimateSenderState": "string",  "ultimateSenderZip": "string",  "ultimateSenderCountry": "string",  "beneficiaryUID": "string",  "beneficiaryName": "string",  "senderAmount": "string",  "paymentDate": "string",  "receiverCurrency": "string",  "senderCurrency": "string",  "receiverAmount": 0,  "deliveryMethod": "ACH",  "receiverAccountNumber": "string",  "ultimateReceiverName": "string",  "ultimateReceiverAccountNumber": "string",  "ultimateReceiverAddress1": "string",  "ultimateReceiverAddress2": "string",  "ultimateReceiverTaxId": "string",  "ultimateReceiverCity": "string",  "ultimateReceiverState": "string",  "ultimateReceiverZip": "string",  "ultimateReceiverCountry": "string",  "createdBy": "string",  "bankFeePayer": "remitter",  "paymentNotes": "payment for invoice ABC123",  "forexContract": "1325",  "occurrences": {  "frequency": "Daily",  "startDate": "2020-01-01",  "payUntil": "Cancelled",  "endDate": "2020-01-29",  "numberOfPayments": 10,  "status": "Active"  },  "status": "FUTURE\_DATED",  "displayStatus": "Scheduled",  "notification": {  "emailAddress": "me@gmail.com",  "phoneNumber": "111223333 or 111-22-3333",  "message": "Payment to XXX"  },  "audit": {  "submitterId": "1234567",  "submissionDate": "3-3-2020",  "approvers": [  {  "approverId": "12345",  "approvedDate": "3-3-2020"  }  ]  },  "creationDateTime": "string",  "customerType": "Corporate or Consumer",  "deliveryAgentType": "string",  "quoteId": "string",  "wireType": "string",  "counterParty": "string",  "businessUnit": "string",  "channel": "string",  "costCenter": "string",  "book": "string",  "businessUnitId": "string"  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [Payment] | false | none | none |

### **Create Credit Payment**

**POST /v3/payments/creditrequest**

**HTTP Method: POST**

**Description -** Sending a POST request to **/v3/payments/creditrequest** endpoint allows users to submit domestic or international one time or recurring payments. All the request parameters and objects are specified below:

Request Parameters

***All details of the users to which we want to make the payment will be passed in the body of the API request.***

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [CreditPaymentRequest](https://apidocs.finzly.net/dashboard?java=#schemacreditpaymentrequest) | true | Payment fields |

**JSON Response**

**A successful POST request to this endpoint will return the following data-**

**Success Code - 200**

| {  "status": "Success or Failure",  "code": "PAYMENT001 etc.",  "message": "Failed to create a payment: ",  "data": {  "paymentUID": "string",  "paymentStatus": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [BookTransferResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemabooktransferresponse_data) | false | none | none |

**Book Transfer Response Data**

| {  "paymentUID": "string",  "paymentStatus": "string",  "externalReferenceId": "string" } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| paymentUID | string | false | none | none |
| paymentStatus | string | false | none | none |
| externalReferenceId | string | false | none | none |

### **Create Debit Payment**

**POST /v3/payments/debitrequest**

**HTTP Method: POST**

**Description -** Sending a POST request to /v3/payments/debitrequest endpoint allows users to submit domestic or international one time or recurring payments. All the request parameters and objects are specified below:

**Request Parameters**

***All details of the users from which want to receive the payment will be passed in the body of the API request.***

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [DebitPaymentRequest](https://apidocs.finzly.net/dashboard?java=#schemadebitpaymentrequest) | true | Payment fields |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Success Code - 200**

| {  "status": "Success or Failure",  "code": "PAYMENT001 etc.",  "message": "Failed to create a payment: ",  "data": {  "paymentUID": "string",  "paymentStatus": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [BookTransferResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemabooktransferresponse_data) | false | none | none |

### **Book Transfer**

**POST /v3/payments/booktransfer**

**HTTP Method: POST**

**Description -** Sending a POST request to the  **/v3/payments/booktransfer**

endpoint allows users to Initiate Book Transfer. All the request parameters and objects are specified below:

**Request Parameters**

**All details will be passed in the body of the API request.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [BookTransferRequest](https://apidocs.finzly.net/dashboard?javascript#schemabooktransferrequest) | true | Payment fields |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Success Code - 200**

| {  "status": "Success or Failure",  "code": "BOOKTRF001 etc.",  "message": "Failed to create a book transfer payment: ",  "data": {  "paymentUID": "string",  "paymentStatus": "string",  "externalReferenceId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [BookTransferResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemabooktransferresponse_data) | false | none | none |

### **Get Payment by UIDV3**

**GET /v3/payments/{paymentUID}**

**HTTP Method: GET**

**Description -** Sending a GET Request to /v3/payments/{paymentUID} endpoint allows users to retrieve payment details using payment uid **in the path of the API request**. All the request parameters and objects are specified below:

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| paymentUID | path | string | true | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Success Code - 200**

| {  "status": "Success or Failure",  "code": "PAYMENT001 etc.",  "message": "Failed to retrieve a payment: ",  "data": {  "paymentUID": 12341,  "counterPartyId": "string",  "senderUID": "string",  "senderAccountNumber": "string",  "senderAccountName": "string",  "senderName": "string",  "senderAccountType": "string",  "ultimateSenderName": "string",  "ultimateSenderAccountNumber": "string",  "ultimateSenderAddress1": "string",  "ultimateSenderAddress2": "string",  "ultimateSenderTaxId": "string",  "ultimateSenderCity": "string",  "ultimateSenderState": "string",  "ultimateSenderZip": "string",  "ultimateSenderCountry": "string",  "beneficiaryUID": "string",  "beneficiaryName": "string",  "senderAmount": "string",  "paymentDate": "string",  "receiverCurrency": "string",  "senderCurrency": "string",  "receiverAmount": 0,  "deliveryMethod": "ACH",  "receiverAccountNumber": "string",  "ultimateReceiverName": "string",  "ultimateReceiverAccountNumber": "string",  "ultimateReceiverAddress1": "string",  "ultimateReceiverAddress2": "string",  "ultimateReceiverTaxId": "string",  "ultimateReceiverCity": "string",  "ultimateReceiverState": "string",  "ultimateReceiverZip": "string",  "ultimateReceiverCountry": "string",  "createdBy": "string",  "bankFeePayer": "remitter",  "paymentNotes": "payment for invoice ABC123",  "forexContract": "1325",  "occurrences": {  "frequency": "Daily",  "startDate": "2020-01-01",  "payUntil": "Cancelled",  "endDate": "2020-01-29",  "numberOfPayments": 10,  "status": "Active"  },  "status": "FUTURE\_DATED",  "displayStatus": "Scheduled",  "notification": {  "emailAddress": "me@gmail.com",  "phoneNumber": "111223333 or 111-22-3333",  "message": "Payment to XXX"  },  "audit": {  "submitterId": "1234567",  "submissionDate": "3-3-2020",  "approvers": [  {  "approverId": "12345",  "approvedDate": "3-3-2020"  }  ]  },  "creationDateTime": "string",  "customerType": "Corporate or Consumer",  "deliveryAgentType": "string",  "quoteId": "string",  "wireType": "string",  "counterParty": "string",  "businessUnit": "string",  "channel": "string",  "costCenter": "string",  "book": "string",  "businessUnitId": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [Payment](https://apidocs.finzly.net/dashboard?java=#schemapayment) | false | none | none |

### **Cancel Payment by paymentUID (v3)**

**PUT /v3/payments/{paymentUID}/cancel**

**HTTP Method: PUT**

**Description-** Sending a PUT Request to /v3/payments/{paymentUID}/cancel endpoint allows users to cancel a payment.using paymentUID. All the request parameters and objects are specified below -

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| paymentUID | path | string | true | none |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

**Success Code - 200**

| {  "status": "Success or Failure",  "code": "PAYMENT001 etc.",  "message": "Failed to cancel a payment: ",  "data": {  "paymentUID": "string"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [CancelPaymentResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemacancelpaymentresponse_data) | false | none | none |

Cancel Payment Response data

| {  "paymentUID": "string" } |
| --- |

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| paymentUID | string | false | none | none |

### **Return a payment by paymentUID (v3)**

**PUT/v3/payments/{paymentUID}/return**

**HTTP Method: PUT**

**Description -** Sending a PUT request to /v3/payments/{paymentUID}/return endpoint allows users to return a payment. All the request parameters and objects are specified below

**Request Parameters**

**In case of an incorrect transaction or refund, the users need to pass paymentUID in the API request path to return a payment.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| paymentUID | path | string | true | none |
| body | body | [ReturnPaymentRequest](https://apidocs.finzly.net/dashboard?java=#schemareturnpaymentrequest) | false | none |

JSON Response not available

### **Reverse Payment by paymentUID (v3)**

**PUT /v3/payments/{paymentUID}/reverse**

**HTTP Method: PUT**

**Description -** Sending a PUT request to /v3/payments/{paymentUID}/reverse endpoint allows users to reverse a payment. All the request parameters and objects are specified below

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| paymentUID | path | string | true | none |
| body | body | [ReversePaymentRequest](https://apidocs.finzly.net/dashboard?java=#schemareversepaymentrequest) | false | none |

**JSON Response not available**

## **Positive Pay**

Payments Positive Pay has two sets of APIs.

Positive Pay Rules APIs enable third parties, bank customers, and downstream partners to define positive rules to monitor ACH debits, set up Fedwire drawdown requests, and "request for pay - RFP" rules for instant payments.

**Positive Pay Decision APIs approve or reject the exceptions identified by the positive pay rules engine.**

### **API Functions**

### **Search Positive Pay Rules**

**POST /v2/positivepay/rules/search**

**HTTP Method: POST**

**Description -** Sending a POST request to /v2/positivepay/rules/search allows users to search positive pay rules for the given criteria. All the request parameters and objects are specified below

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| body | body | [ReversePaymentRequest](https://apidocs.finzly.net/dashboard?java=#schemareversepaymentrequest) | false | none |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to get positive rules: ",  "data": [  {  "ruleUID": 123456789,  "paymentRuleType": "ACH, FEDWIRE, RFP",  "customerName": "ABC Company",  "customerAccountNumber": "12387878",  "achCompanyID": 122334789,  "beneBankId": "12345678",  "beneName": "XYZ Name",  "beneBankName": "Bank Name",  "beneAccountNumber": "12345678",  "amount": 10,  "currency": "USD",  "amountType": "Exact or Maximum",  "frequency": "Onetime, Weekly etc.",  "secCode": "CCD, PPD etc.",  "status": "Active, Expired",  "effectiveDate": "0016-07-14",  "expiryDate": "0016-07-14"  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | Status of the API request either it will be a success or a failure |
| code | string | false | none | Code associated with the error. |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | [[PositivePayRule](https://apidocs.finzly.net/dashboard?java=#schemapositivepayrule)] | false | none | none |

**Positive Pay Rule**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| ruleUID | number(int64) | false | none | Positive rule unique reference id |
| paymentRuleType | string | false | none | Payment type associated with the rule such as ACH for Positive Pay, FEDWIRE for Drawdown, RTP for Request for Pay etc. |
| customerName | string | false | none | Customer name associated with the payment positive rule |
| customerAccountNumber | string | false | none | Bank Account Number |
| achCompanyID | number | false | none | ACH company ID is a 10-digit identifier used by banks and Nacha, the operator of the ACH network, to identify the entity collecting payments or sending money via ACH (also referred to as an “originator”) |
| beneBankId | string | false | none | Bank unique id that is used to identify a specific bank |
| beneName | string | false | none | A beneficiary is the person or entity that will receive the payment |
| beneBankName | string | false | none | Beneficiary bank name |
| beneAccountNumber | string | false | none | A beneficiary bank account number |
| amount | number | false | none | Payment amount |
| currency | string | false | none | Currency Code |
| amountType | string | false | none | Type of amount associated with the positive pay rule. |
| frequency | string | false | none | Positive pay rule frequency |
| secCode | string | false | none | SEC code associated with the ACH positive rule |
| status | string | false | none | Current state of the rule whether Active or Expired etc. |
| effectiveDate | string(date) | false | none | Rule effective date in mm-dd-yyyy format |
| expiryDate | string(date) | false | none | Rule effective date in mm-dd-yyyy format |

### **Update Positive Pay Rules(v2)**

**PUT /v2/positivepay/rules**

**HTTP Method: PUT**

**Description -** Sending a PUT request to /v2/positivepay/rules endpoint allows users to update Positive Pay Rule. All the request parameters and objects are specified below

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [PositivePayRule](https://apidocs.finzly.net/dashboard?java=#schemapositivepayrule) | false | none |

**JSON Response  
A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to update a positive rule: .",  "ruleUID": 123456789 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| ruleUID | number(int64) | false | none | Positive rule unique id |

### **Add PositivePay Rule (v2)**

**POST /v2/positivepay/rules**

**HTTP Method: POST**

**Description -**  Sending a PUT request to the **/v2/positivepay/rules** endpoint allows users to add a positive pay rule. All the request parameters and objects are specified below-

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [AddPositivePayRule](https://apidocs.finzly.net/dashboard?java=#schemaaddpositivepayrule) | false | none |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to add a positive rule: .",  "ruleUID": 123456789 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| ruleUID | number(int64) | false | none | Positive rule unique id |

### **Get PositivePay Rule By RuleUID (v2)**

**GET /v2/positivepay/rules/{ruleUID}**

**HTTP Method - GET**

**Description -** Sending a GET request to the **/v2/positivepay/rules/{ruleUID}** endpoint allows users to get PositivePay Rule by providing ruleUID in the path of the API request. All the request parameters and objects are specified below-

Request Parameters

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| ruleUID | path | string | true | Positive Pay Rule unique id |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to get positive rule: ",  "data": {  "ruleUID": 123456789,  "paymentRuleType": "ACH, FEDWIRE, RFP",  "customerName": "ABC Company",  "customerAccountNumber": "12387878",  "achCompanyID": 122334789,  "beneBankId": "12345678",  "beneName": "XYZ Name",  "beneBankName": "Bank Name",  "beneAccountNumber": "12345678",  "amount": 10,  "currency": "USD",  "amountType": "Exact or Maximum",  "frequency": "Onetime, Weekly etc.",  "secCode": "CCD, PPD etc.",  "status": "Active, Expired",  "effectiveDate": "0016-07-14",  "expiryDate": "0016-07-14"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | PositivePayRule | false | none | none |

### **Delete Rules**

**DELETE /v2/positivepay/rules/{ruleUID}**

**HTTP Method: Delete**

**Description -** Sending a GET request to the /v2/positivepay/rules/{ruleUID} endpoint allows users to Delete PositivePay Rule by providing ruleUID in the path of the API request. All the request parameters and objects are specified below-

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| ruleUID | path | number(int64) | true | Positive Pay Rule unique id |

**JSON Response**

**A successful DELETE request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to delete a positive rule: .",  "ruleUID": 123456789 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| ruleUID | number(int64) | false | none | Positive rule unique id |

### **Activate Rules**

**PUT /v2/positivepay/rules/{ruleUID}/activate**

**HTTP Method: PUT**

**Description -** Sending a PUT request to the **/v2/positivepay/rules/{ruleUID}/activate** endpoint allows users to activate Positive Pay Rule by providing ruleUID in the path of the API request. All the request parameters and objects are specified below-

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| ruleUID | path | string | true | Positive Pay Rule unique id |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to activate or suspend a positive rule: .",  "ruleUID": 123456789 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| ruleUID | number(int64) | false | none | Positive rule unique id |

### **Suspend Rules**

**PUT /v2/positivepay/rules/{ruleUID}/suspend**

**HTTP Method: PUT**

**Description -** Sending a PUT request to PUT /v2/positivepay/rules/{ruleUID}/suspend endpoint allows users to suspend Positive Pay Rules by providing ruleUID in the path of the API request. All the request parameters and objects are specified below-

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| ruleUID | path | string | true | Positive Pay Rule unique id |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PP001 etc.",  "message": "Failed to activate or suspend a positive rule: .",  "ruleUID": 123456789 } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| ruleUID | number(int64) | false | none | Positive rule unique id |

### **Search Positive Pay Expectations**

**POST /v2/positivepay/exception/search**

**HTTP Method: POST**

**Description -** Sending a POST request to /v2/positivepay/exception/search endpoint allows users to return positive pay expectations for a given criteria. All the request parameters and objects are specified below-

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [SearchPositivePayDecisionRequest](https://apidocs.finzly.net/dashboard?java=#schemasearchpositivepaydecisionrequest) | false | none |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PPD001 etc.",  "message": "Failed to get the positive pay exception information: .",  "data": [  {  "exceptionUID": "string",  "paymentUID": "100100",  "paymentRuleType": "ACH, FEDWIRE, RFP",  "amount": 123456789,  "achCompanyID": 122334789,  "customerAccountNumber": "12387878",  "beneName": "XYZ Name",  "exceptionReason": "Amount Mismatch",  "exceptionDateTime": "2019-08-24",  "expiryDateTime": "2019-08-24",  "positivePayDecisionStatus": "string",  "secCode": "CCD, PPD etc.",  "approvedBy": "user001",  "approvedDate": "2019-08-24"  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| ruleUID | [[PositivePayExceptionDecision](https://apidocs.finzly.net/dashboard?java=#schemapositivepayexceptiondecision)] | false | none | Positive rule unique id |

### **Approve PositivePay Exception**

**PUT /v2/positivepay/exception/{exceptionUID}/approve**

**HTTP Method: PUT**

**Description -** Sending a PUT request to the **/v2/positivepay/exception/{exceptionUID}/approve** endpoint allows users to approve Positive Pay Exception by passing the ExceptionUID in the API request path. All the request parameters and objects are specified below-

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| exceptionUID | path | string | true | A unique identifier associated with the positive pay rule exception |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PPD001 etc.",  "message": "Failed to activate or suspend a positive pay exception: .",  "exceptionUID": 123456789 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| exceptionUID | number(int64) | false | none | Unique identifier associated with positive pay exception |

### **Reject Exceptions**

**PUT /v2/positivepay/exception/{exceptionUID}/reject**

**HTTP Method: PUT**

**Description -** Sending a GET request to the /v2/positivepay/exception/{exceptionUID}/reject allows users to reject exceptions by passing the exceptionUID in the path of the API request.

All the request parameters and objects are specified below-

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| exceptionUID | path | string | true | A unique identifier associated with the positive pay rule exception |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PPD001 etc.",  "message": "Failed to activate or suspend a positive pay exception: .",  "exceptionUID": 123456789 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| exceptionUID | number(int64) | false | none | Unique identifier associated with positive pay exception |

### **Approve All Exceptions**

**PUT /v2/positivepay/exception/approveall**

**HTTP Method: PUT**

Description - Sending a PUT request to **/v2/positivepay/exception/approveall** endpoint allows users toapprove all positive pay exceptions via payment unique identifier in the API request path. All the request parameters and objects are specified below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [ApproveAllPositivePayExceptionRequest](https://apidocs.finzly.net/dashboard?java=#schemaapproveallpositivepayexceptionrequest) | false | none |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "PPD001 etc.",  "message": "Failed to approve all positive pay rule exception: ." } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |

### **Foreign Exchange**

Foreign Exchange (forex or FX) is trading one currency for another. For example, one can swap the U.S. dollar for the euro. Foreign Exchange Finzly FX APIs enable third parties, banks, and banks' downstream partners to initiate and manage FX trades for their customers.

Following functions helps you to understand the current FX capabilities exposed via APIs

1. FX Quote
2. FX Quote Accept/Reject
3. Get Trade By ForexContractNo
4. Delete Trade By ForexContractNo

### **API Functions**

### **Get FX Quote**

**POST /v2/fx/quote**

**HTTP Method: GET**

**Description -** Sending a GET request to the **/v2/fx/quote** endpoint allows users to request a quote for FX trade. All the parameters and objects are specified below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [FXQuoteRequest](https://apidocs.finzly.net/dashboard?java=#schemafxquoterequest) | true | Quote request attributes to get the quote for a given ccy pair |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "FXQUOTE001 etc.",  "message": "Failed to get a FX Quote due to the mandatory attribute is not provide.",  "data": {  "externalReferenceId": "REQ123 or 1274878",  "quoteUUID": "123e4567-e89b-12d3-a456-426614174000",  "expirySeconds": 30,  "exchangeRate": 1.2,  "invertedRate": 0.83,  "sellCurrency": "EUR",  "buyCurrency": "INR",  "fundingAmount": 10000,  "transferAmount": 1000,  "productType": "SPOT or FORWARD",  "tenor": "SPOT, ON, TN, nW, nM, nY where n is a number",  "valueDate": "2020-01-01",  "externalPricingSystemRefId": "R1274878 etc."  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [FXQuoteResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemafxquoteresponse_data) | false | none | none |

### **Accept FX Quote**

**PUT /v2/fx/accept**

**HTTP Method: PUT**

**Description -** Sending PUT request to **/v2/fx/accept** endpoint allows users to accept FX Quote using quoteUUID in the API request path. All the request parameters and objects are specified below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [FXAcceptQuoteRequest](https://apidocs.finzly.net/dashboard?java=#schemafxacceptquoterequest) | false | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "FXQUOTE002 etc.",  "message": "Failed to accept a quote: ",  "data": {  "quoteUUID": "123e4567-e89b-12d3-a456-426614174000",  "forexContractNo": "123898",  "tradeStatus": "Booked etc.",  "tradeDate": "2020-01-01",  "ccyPair": "USDEUR",  "direction": "Buy or Sell",  "baseCcy": "USD",  "baseAmount": 1000,  "quoteCcy": "USD",  "quoteAmount": 900,  "tenor": "SPOT, ON, TN, nW, nM, nY where n is a number",  "exchangeRate": 1.2,  "valueDate": "2020-01-01",  "referenceSpotRate": 100,  "referenceFwdPoints": 100,  "executionTime": "2019-08-24",  "externalPricingSystemRefId": "R1274878 etc."  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [FXAcceptQuoteResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemafxacceptquoteresponse_data) | false | none | none |

FX Accept Quote Response Data

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| quoteUUID | string | false | none | FX Quote universally unique identifier (UUID) |
| forexContractNo | string | false | none | FX contract number assigned by the system after acceptance. |
| tradeStatus | string | false | none | Trade status after acceptance of fx quote |
| tradeDate | string(date) | false | none | The trade date is the date on which a transaction was executed. The date format is in yyyy-mm-dd. |
| ccyPair | string | false | none | A currency pair is the quotation of two different currencies, with the value of one currency being quoted against the other |
| direction | string | false | none | Buy or Sell |
| baseCcy | string | false | none | The base currency is the first currency stated in a currency pair quote |
| baseAmount | number | false | none | Base currency amount |
| quoteCcy | string | false | none | The second currency is the quote currency, which states how much of the quote currency is required to buy one unit of the base currency. |
| quoteAmount | number | false | none | Quote currency amount |
| tenor | string | false | none | The length of time remaining before a financial contract expires. Only one is needed between tenor or valueDate. |
| exchangeRate | number | false | none | The price of one currency in terms of the other |
| valueDate | string(date) | false | none | The delivery date on which counterparties to a transaction agree to settle their respective obligations by making payments and transferring ownership.Only one is needed between tenor or valueDate. The date format is in yyyy-mm-dd. |
| referenceSpotRate | number | false | none | the offered rate |
| referenceFwdPoints | number | false | none | The number of basis points added to or subtracted from the current spot rate of a currency pair to determine the forward rate for delivery on a specific value date |
| executionTime | string(date) | false | none | The execution time |
| externalPricingSystemRefId | string | false | none | Unique reference id returned by the external system provider. |

### **Reject FX Quote**

**PUT /v2/fx/{quoteUUID}/reject**

**HTTP Method: PUT**

**Description -** Sending a PUT request to PUT /v2/fx/{quoteUUID}/reject endpoint allows user to reject FX Quote using quoteUUID in the path of the API request. All the request parameters and objects are mentioned below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| quoteUUID | path | string | true | none |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "FXQUOTE003 etc.",  "message": "Failed to reject a quote: ",  "data": {  "quoteUUID": "123e4567-e89b-12d3-a456-426614174000",  "fxQuoteStatus": "Rejected"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [FXRejectQuoteResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemafxrejectquoteresponse_data) | false | none | none |

**FX Reject Quote Response Data**

| {  "quoteUUID": "123e4567-e89b-12d3-a456-426614174000",  "fxQuoteStatus": "Rejected" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| quoteUUID | string | false | none | FX Quote universally unique identifier (UUID) |
| fxQuoteStatus | string | false | none | FX Quote status after rejection |

### **Get FX trade information (v2)**

**GET /v2/fx/{forexContractNo}**

**HTTP Method: GET**

**Description -** Sending a GET request to **/v2/fx/{forexContractNo}** allows users to get FX trade information using forexContractNo in the path of the API request. All the request parameters and objects are mentioned below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | path | string | true | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "FX001 etc.",  "message": "Failed to get trade information: ",  "data": {  "forexContractNo": "123898",  "tradeDate": "2020-01-01",  "ccyPair": "USDEUR",  "direction": "Buy or Sell",  "baseCcy": "USD",  "baseAmount": 1000,  "quoteCcy": "USD",  "quoteAmount": 900,  "tenor": "SPOT, ON, TN, nW, nM, nY where n is a number",  "exchangeRate": 1.2,  "valueDate": "2020-01-01",  "referenceSpotRate": 100,  "referenceFwdPoints": 100,  "executionTime": "2019-08-24"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [FXTradeInfoResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemafxtradeinforesponse_data) | false | none | none |

**FX Trade Response Data**

| {  "forexContractNo": "123898",  "tradeDate": "2020-01-01",  "ccyPair": "USDEUR",  "direction": "Buy or Sell",  "baseCcy": "USD",  "baseAmount": 1000,  "quoteCcy": "USD",  "quoteAmount": 900,  "tenor": "SPOT, ON, TN, nW, nM, nY where n is a number",  "exchangeRate": 1.2,  "valueDate": "2020-01-01",  "referenceSpotRate": 100,  "referenceFwdPoints": 100,  "executionTime": "2019-08-24" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | string | false | none | FX contract number assigned by the system for a given trade. |
| tradeDate | string(date) | false | none | The trade date is the date on which a transaction was executed. The date format is in yyyy-mm-dd. |
| ccyPair | string | false | none | A currency pair is the quotation of two different currencies, with the value of one currency being quoted against the other |
| direction | string | false | none | Buy or Sell |
| baseCcy | string | false | none | The base currency is the first currency stated in a currency pair quote |
| baseAmount | number | false | none | Base currency amount |
| quoteCcy | string | false | none | The second currency is the quote currency, which states how much of the quote currency is required to buy one unit of the base currency. |
| quoteAmount | number | false | none | Quote currency amount |
| tenor | string | false | none | The length of time remaining before a financial contract expires. Only one is needed between tenor or valueDate. |
| exchangeRate | number | false | none | The price of one currency in terms of the other |
| valueDate | string(date) | false | none | The delivery date on which counterparties to a transaction agree to settle their respective obligations by making payments and transferring ownership.Only one is needed between tenor or valueDate. The date format is in yyyy-mm-dd. |
| referenceSpotRate | number | false | none | The price of one currency in terms of the other |
| valueDate | string(date) | false | none | The delivery date on which counterparties to a transaction agree to settle their respective obligations by making payments and transferring ownership.Only one is needed between tenor or valueDate. The date format is in yyyy-mm-dd. |
| referenceSpotRate | number | false | none | the offered rate |
| referenceFwdPoints | number | false | none | The number of basis points added to or subtracted from the current spot rate of a currency pair to determine the forward rate for delivery on a specific value date |
| executionTime | string(date) | false | none | The execution time |

### **Cover FX Trade**

**GET /v2/fx/{forexContractNo}/cover**

**HTTP Method: GET**

**Description -** Sending a GET request to the GET /v2/fx/{forexContractNo}/cover endpoint allows users to cover FX trade by passing the forexContractNo in the path of the API request. All the parameters and objects are specified below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | path | string | true | none |

**JASON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "FX001 etc.",  "message": "Failed to perform the cover trade: ",  "data": {  "forexContractNo": "123898",  "coverTradeNo": "990919",  "externalPricingSystemRefId": "R1274878 etc.",  "channel": "API",  "customerIdType": "CustomerUID, CustomerShortName,CustomerId, DDA",  "customerId": "112344 for DDA",  "tradeDate": "2020-01-01",  "valueDate": "2020-01-01",  "ccyPair": "USDEUR",  "direction": "Buy or Sell",  "baseCcy": "USD",  "baseAmount": 1000,  "quoteCcy": "EUR",  "quoteAmount": 955.08,  "tenor": "SPOT, ON, TN, nW, nM, nY where n is a number",  "allInRate": 0.98,  "referenceSpotRate": 100  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [FXCoverTradeResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemafxcovertraderesponse_data) | false | none | none |

**FX Cover Trader Response Data**

| {  "forexContractNo": "123898",  "coverTradeNo": "990919",  "externalPricingSystemRefId": "R1274878 etc.",  "channel": "API",  "customerIdType": "CustomerUID, CustomerShortName,CustomerId, DDA",  "customerId": "112344 for DDA",  "tradeDate": "2020-01-01",  "valueDate": "2020-01-01",  "ccyPair": "USDEUR",  "direction": "Buy or Sell",  "baseCcy": "USD",  "baseAmount": 1000,  "quoteCcy": "EUR",  "quoteAmount": 955.08,  "tenor": "SPOT, ON, TN, nW, nM, nY where n is a number",  "allInRate": 0.98,  "referenceSpotRate": 100 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | string | false | none | FX contract number assigned by the system for a given trade. |
| coverTradeNo | string | false | none | Unique cover trade number assigned by the system.Unique cover trade number assigned by the system. |
| externalPricingSystemRefId | string | false | none | Unique reference id provided by the external system |
| channel | string | false | none | Channel associated with the cover trade. |
| customerIdType | string | false | none | CustomerId type associated with the cover trade |
| customerId | string | false | none | The value based on the customerIdType associated with the cover trade |
| tradeDate | string | false | none | The trade date is the date on which a transaction was executed. The date format is in yyyy-mm-dd. |
| valueDate | number | false | none | The delivery date on which counterparties to a transaction agree to settle their respective obligations by making payments and transferring ownership.Only one is needed between tenor or valueDate. The date format is in yyyy-mm-dd. |
| ccyPair | string | false | none | A currency pair is the quotation of two different currencies, with the value of one currency being quoted against the other |
| direction | string | false | none | Buy or Sell |
| baseCcy | string | false | none | The base currency is the first currency stated in a currency pair quote |
| baseAmount | number | false | none | Base currency amount |
| quoteCcy | string | false | none | The second currency is the quote currency, which states how much of the quote currency is required to buy one unit of the base currency. |
| quoteAmount | number | false | none | Quote currency amount |
| tenor | string | false | none | The length of time remaining before a financial contract expires. Only one is needed between tenor or valueDate. |
| allInRate | number | false | none | The all in rate associated with the cover trade |
| referenceSpotRate | number | false | none | the offered rate |

### **Cancel FX Trade**

**DELETE /v2/fx/{forexContractNo}/cancel**

**HTTP Method: Delete**

**Description -** Sending a DELETE request to the /v2/fx/{forexContractNo}/cancel endpoint allows users to cancel FX trade by passing the forexContractNo in the path of the API request. All the parameters and objects are specified below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | path | string | true | none |

**JASON Response**

**A successful Delete request to this endpoint returns the following data -**

**Response Code - 200**

| {  "status": "Success or Failure",  "code": "FX001 etc.",  "message": "Failed to cancel a trade: ",  "data": {  "forexContractNo": "123898",  "tradeStatus": "Cancelled etc."  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [FXTradeCancelResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemafxtradecancelresponse_data) | false | none | none |

**FX Trade Cancel Response Data**

| {  "forexContractNo": "123898",  "tradeStatus": "Cancelled etc." } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | string | false | none | FX contract number assigned by the system for a given trade. |
| tradeStatus | string | false | none | Trade status after trade cancelation |

### **Bulk Payments**

Finzly system supports the ACH bulk payment processing in a NACHA format via FTP channel. The bank can create an FTP account to process the ACH bulk file in a Nacha format. The solution provided two types of FTP account.

**Following functions helps you to understand the current FX capabilities exposed via APIs**

**Get Supported File Types**

**GET /bulkpayment/files/supportedFileType**

**HTTP Method: GET**

**Description** - Sending a GET request to the endpoint allows users to get supported files in a list format.

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "trasactionTypesList": [  {  "trasactionType": "NACHA, FEDWIRE, FINZLY-CSV",  "fileType": ".csv or .txt etc"  }  ],  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  } } |
| --- |

### **Get Bulk File Payment Details**

**GET /bulkpayment/files/{fileNumber}**

**HTTP Method: GET**

**Description -** Sending a GET request to the **/bulkpayment/files/{fileNumber}** endpoint allows users to get bulk file payment details. The request parameters and objects are specified below-

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| forexContractNo | path | string | true | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

**Response Code - 200**

| {  "fileNumber": 0,  "fileName": "string",  "fileDateTime": "string",  "fileStatus": "string",  "fileType": "string",  "totalDebit": 0,  "totalCredit": 0,  "totalPayments": 0,  "validPayments": 0,  "errorPayments": 0,  "creditMemoPostStatus": "string",  "debitMemoPostStatus": "string",  "counterparty": "string",  "counterpartyId": 0,  "paymentDate": "2019-08-24",  "channel": "string",  "possibleUserAction": [  {  "id": "Approved or Rejected etc",  "displayName": "APPROVED / REJECTED etc",  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  }  }  ],  "paymentBatches": [  {  "batchId": "string",  "totalBatchDebit": 0,  "totalBatchCredit": 0,  "batchCurrency": "string",  "secCode": "PPD, CCD etc",  "status": "VALIDATED or ERROR",  "transactions": [  {  "paymentId": 0,  "transactionType": "string",  "amount": 0,  "currency": "string",  "deliveryMethod": "ACH",  "createdBy": "string",  "paymentType": "string",  "paymentDate": "2019-08-24",  "paymentStatus": "string",  "settlementDate": "2019-08-24",  "lastUpdatedBy": "string",  "creationDateTime": "string",  "lastUpdatedTime": "string",  "receiverName": "string",  "receiverAccountNumber": "string",  "senderAccountNumber": "string"  }  ]  }  ],  "messages": {  "uploadErrors": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ],  "uploadWarnings": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ]  },  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| fileNumber | integer | false | none | none |
| fileName | string | false | none | none |
| fileDateTime | string(dateTime) | false | none | none |
| fileStatus | string | false | none | none |
| fileType | string | false | none | none |
| totalDebit | number(double) | false | none | none |
| totalCredit | number(double) | false | none | none |
| totalPayments | number(integer) | false | none | none |
| validPayments | number(integer) | false | none | none |
| errorPayments | number(integer) | false | none | none |
| creditMemoPostStatus | string | false | none | none |
| debitMemoPostStatus | string | false | none | none |
| counterparty | string | false | none | none |
| counterpartyId | integer | false | none | none |
| paymentDate | string(date) | false | none | Payment date in mm/dd/yyyy format |
| channel | string | false | none | none |
| possibleUserAction | [[BulkfileActions](https://apidocs.finzly.net/dashboard?java=#schemabulkfileactions)] | false | none | none |
| paymentBatches | [[BulkPaymentBatches](https://apidocs.finzly.net/dashboard?java=#schemabulkpaymentbatches)] | false | none | none |
| messages | [BulkFile\_messages](https://apidocs.finzly.net/dashboard?java=#schemabulkfile_messages) | false | none | none |
| errors | [Error](https://apidocs.finzly.net/dashboard?java=#schemaerror) | false | none | none |

**Bulk File Actions**

| {  "id": "Approved or Rejected etc",  "displayName": "APPROVED / REJECTED etc",  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| id | string | false | none | none |
| displayName | string | false | none | none |
| errors | [Error](https://apidocs.finzly.net/dashboard?java=#schemaerror) | false | none | none |

**Error**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| code | string | false | none | Error code assigned by the system |
| description | string | false | none | Error description to provide additional details about the error code |

**Bulk Payment Batches**

| {  "batchId": "string",  "totalBatchDebit": 0,  "totalBatchCredit": 0,  "batchCurrency": "string",  "secCode": "PPD, CCD etc",  "status": "VALIDATED or ERROR",  "transactions": [  {  "paymentId": 0,  "transactionType": "string",  "amount": 0,  "currency": "string",  "deliveryMethod": "ACH",  "createdBy": "string",  "paymentType": "string",  "paymentDate": "2019-08-24",  "paymentStatus": "string",  "settlementDate": "2019-08-24",  "lastUpdatedBy": "string",  "creationDateTime": "string",  "lastUpdatedTime": "string",  "receiverName": "string",  "receiverAccountNumber": "string",  "senderAccountNumber": "string"  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| batchId | string | false | none | none |
| totalBatchDebit | number(double) | false | none | none |
| totalBatchCredit | number(double) | false | none | none |
| batchCurrency | string | false | none | none |
| secCode | string | false | none | none |
| status | string | false | none | none |
| transactions | [[BulkTransaction](https://apidocs.finzly.net/dashboard?java=#schemabulktransaction)] | false | none | none |

**Bulk Transaction**

| {  "paymentId": 0,  "transactionType": "string",  "amount": 0,  "currency": "string",  "deliveryMethod": "ACH",  "createdBy": "string",  "paymentType": "string",  "paymentDate": "2019-08-24",  "paymentStatus": "string",  "settlementDate": "2019-08-24",  "lastUpdatedBy": "string",  "creationDateTime": "string",  "lastUpdatedTime": "string",  "receiverName": "string",  "receiverAccountNumber": "string",  "senderAccountNumber": "string" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| paymentId | number | false | none | none |
| transactionType | string | false | none | none |
| amount | number(double) | false | none | none |
| currency | string | false | none | none |
| deliveryMethod | string | false | none | none |
| createdBy | string | false | none | none |
| paymentType | string | false | none | none |
| paymentDate | string(date) | false | none | Payment date in mm/dd/yyyy format |
| paymentStatus | string | false | none | none |
| settlementDate | string(date) | false | none | Settlement date in mm/dd/yyyy format |
| lastUpdatedBy | string | false | none | none |
| lastUpdatedTime | string(datetime) | false | none | none |
| receiverName | string | false | none | none |
| receiverAccountNumber | string | false | none | none |
| senderAccountNumber | string | false | none | none |

**Bulk File Messages**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| uploadErrors | [[UploadMessages](https://apidocs.finzly.net/dashboard?java=#schemauploadmessages)] | false | none | none |
| uploadWarnings | [[UploadMessages](https://apidocs.finzly.net/dashboard?java=#schemauploadmessages)] | false | none | none |

**Upload Messages**

| {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| bulkFileId | integer | false | none | none |
| fileNumber | integer | false | none | none |
| validationCode | string | false | none | none |
| displayMessage | string | false | none | none |
| validationType | string | false | none | none |
| creationDateTime | string(datetime) | false | none | none |
| approvedBy | string | false | none | none |
| approvedDateTime | string(datetime) | false | none | none |

**Error**

| {  "code": "invalidMethod",  "description": "Invalid HTTP method used" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| code | string | false | none | Error code assigned by the system |
| description | string | false | none | Error description to provide additional details about the error code |

### **Upload Bulk Files**

**POST /bulkpayment/files**

**HTTP Method: POST**

**Description -** Sending a POST request to /bulkpayment/files allows users to Upload files for bulk payment. The request parameters and objects are specified below-

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [BulkFilesRequest](https://apidocs.finzly.net/dashboard?java=#schemabulkfilesrequest) | false | none |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "bulkUploadedFiles": [  {  "fileNumber": 0,  "fileName": "string",  "fileDateTime": "string",  "fileStatus": "string",  "fileType": "string",  "totalDebit": 0,  "totalCredit": 0,  "totalPayments": 0,  "validPayments": 0,  "errorPayments": 0,  "creditMemoPostStatus": "string",  "debitMemoPostStatus": "string",  "counterparty": "string",  "counterpartyId": 0,  "paymentDate": "2019-08-24",  "channel": "string",  "possibleUserAction": [  {  "id": "Approved or Rejected etc",  "displayName": "APPROVED / REJECTED etc",  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  }  }  ],  "paymentBatches": [  {  "batchId": "string",  "totalBatchDebit": 0,  "totalBatchCredit": 0,  "batchCurrency": "string",  "secCode": "PPD, CCD etc",  "status": "VALIDATED or ERROR",  "transactions": [  {  "paymentId": 0,  "transactionType": "string",  "amount": 0,  "currency": "string",  "deliveryMethod": "ACH",  "createdBy": "string",  "paymentType": "string",  "paymentDate": "2019-08-24",  "paymentStatus": "string",  "settlementDate": "2019-08-24",  "lastUpdatedBy": "string",  "creationDateTime": "string",  "lastUpdatedTime": "string",  "receiverName": "string",  "receiverAccountNumber": "string",  "senderAccountNumber": "string"  }  ]  }  ],  "messages": {  "uploadErrors": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ],  "uploadWarnings": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ]  },  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  }  }  ],  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  },  "pagination": {  "totalRecords": 100,  "returnedRecords": 10,  "pageReturned": 2,  "pageSize": 10  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| bulkUploadedFiles | [[BulkFile](https://apidocs.finzly.net/dashboard?java=#schemabulkfile)] | false | none | none |
| errors | [Error](https://apidocs.finzly.net/dashboard?java=#schemaerror) | false | none | none |
| pagination | [Pagination](https://apidocs.finzly.net/dashboard?java=#schemapagination) | false | none | none |

**Bulk File**

| {  "fileNumber": 0,  "fileName": "string",  "fileDateTime": "string",  "fileStatus": "string",  "fileType": "string",  "totalDebit": 0,  "totalCredit": 0,  "totalPayments": 0,  "validPayments": 0,  "errorPayments": 0,  "creditMemoPostStatus": "string",  "debitMemoPostStatus": "string",  "counterparty": "string",  "counterpartyId": 0,  "paymentDate": "2019-08-24",  "channel": "string",  "possibleUserAction": [  {  "id": "Approved or Rejected etc",  "displayName": "APPROVED / REJECTED etc",  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  }  }  ],  "paymentBatches": [  {  "batchId": "string",  "totalBatchDebit": 0,  "totalBatchCredit": 0,  "batchCurrency": "string",  "secCode": "PPD, CCD etc",  "status": "VALIDATED or ERROR",  "transactions": [  {  "paymentId": 0,  "transactionType": "string",  "amount": 0,  "currency": "string",  "deliveryMethod": "ACH",  "createdBy": "string",  "paymentType": "string",  "paymentDate": "2019-08-24",  "paymentStatus": "string",  "settlementDate": "2019-08-24",  "lastUpdatedBy": "string",  "creationDateTime": "string",  "lastUpdatedTime": "string",  "receiverName": "string",  "receiverAccountNumber": "string",  "senderAccountNumber": "string"  }  ]  }  ],  "messages": {  "uploadErrors": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ],  "uploadWarnings": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ]  },  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  } } |
| --- |

**Data Returned Objects**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| fileNumber | integer | false | none | none |
| fileName | string | false | none | none |
| fileDateTime | string(datetime) | false | none | none |
| fileStatus | string | false | none | none |
| fileType | string | false | none | none |
| totalDebit | number(double) | false | none | none |
| totalCredit | number(double) | false | none | none |
| totalPayments | number(double) | false | none | none |
| validPayments | number(integer) | false | none | none |
| errorPayments | number(integer) | false | none | none |
| creditMemoPostStatus | string | false | none | none |
| debitMemoPostStatus | string | false | none | none |
| counterparty | string | false | none | none |
| counterpartyId | integer | false | none | none |
| paymentDate | string(date) | false | none | Payment date in mm/dd/yyyy format |
| channel | string | false | none | none |
| possibleUserAction | [[BulkfileActions](https://apidocs.finzly.net/dashboard?java=#schemabulkfileactions)] | false | none | none |
| paymentBatches | [[BulkPaymentBatches](https://apidocs.finzly.net/dashboard?java=#schemabulkpaymentbatches)] | false | none | none |
| messages | [BulkFile\_messages](https://apidocs.finzly.net/dashboard?java=#schemabulkfile_messages) | false | none | none |
| errors | [Error](https://apidocs.finzly.net/dashboard?java=#schemaerror) | false | none | none |

**Error**

| {  "code": "invalidMethod",  "description": "Invalid HTTP method used" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| code | string | false | none | Error code assigned by the system |
| description | string | false | none | Error description to provide addtional details about the error code |

**Pagination**

| {  "totalRecords": 100,  "returnedRecords": 10,  "pageReturned": 2,  "pageSize": 10 } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| totalRecords | number(integer) | false | none | Total number of records exist in the system for a given filter |
| returnedRecords | number(integer) | false | none | Total number of records returned |
| pageReturned | number(integer) | false | none | Current number of page returned |
| pageSize | number(integer) | false | none | Total number of records in a given pages |

### **Get Bulk Files**

**POST /bulkpayment/files/search**

**HTTP Method: POST**

Description - Sending a POST request to **/bulkpayment/files/search** allows users to search all files based on CounterParty (and Payment Date criteria). The request parameters and objects are specified below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [BulkFilesRequest](https://apidocs.finzly.net/dashboard?java=#schemabulkfilesrequest) | false | none |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

**Response Code - 200**

| {  "bulkUploadedFiles": [  {  "fileNumber": 0,  "fileName": "string",  "fileDateTime": "string",  "fileStatus": "string",  "fileType": "string",  "totalDebit": 0,  "totalCredit": 0,  "totalPayments": 0,  "validPayments": 0,  "errorPayments": 0,  "creditMemoPostStatus": "string",  "debitMemoPostStatus": "string",  "counterparty": "string",  "counterpartyId": 0,  "paymentDate": "2019-08-24",  "channel": "string",  "possibleUserAction": [  {  "id": "Approved or Rejected etc",  "displayName": "APPROVED / REJECTED etc",  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  }  }  ],  "paymentBatches": [  {  "batchId": "string",  "totalBatchDebit": 0,  "totalBatchCredit": 0,  "batchCurrency": "string",  "secCode": "PPD, CCD etc",  "status": "VALIDATED or ERROR",  "transactions": [  {  "paymentId": 0,  "transactionType": "string",  "amount": 0,  "currency": "string",  "deliveryMethod": "ACH",  "createdBy": "string",  "paymentType": "string",  "paymentDate": "2019-08-24",  "paymentStatus": "string",  "settlementDate": "2019-08-24",  "lastUpdatedBy": "string",  "creationDateTime": "string",  "lastUpdatedTime": "string",  "receiverName": "string",  "receiverAccountNumber": "string",  "senderAccountNumber": "string"  }  ]  }  ],  "messages": {  "uploadErrors": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ],  "uploadWarnings": [  {  "bulkFileId": 0,  "fileNumber": 0,  "validationCode": "string",  "displayMessage": "string",  "validationType": "string",  "creationDateTime": "string",  "approvedBy": "string",  "approvedDateTime": "string"  }  ]  },  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  }  }  ],  "errors": {  "code": "invalidMethod",  "description": "Invalid HTTP method used"  },  "pagination": {  "totalRecords": 100,  "returnedRecords": 10,  "pageReturned": 2,  "pageSize": 10  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| bulkUploadedFiles | [[BulkFile](https://apidocs.finzly.net/dashboard?java=#schemabulkfile)] | false | none | none |
| errors | [Error](https://apidocs.finzly.net/dashboard?java=#schemaerror) | false | none | none |
| pagination | [Pagination](https://apidocs.finzly.net/dashboard?java=#schemapagination) | false | none | none |

### **Approve File**

**PUT /bulkpayment/files/{fileNumber}/approve**

**HTTP Method: PUT**

**Description -** Sending a PUT request to PUT /bulkpayment/files/{fileNumber}/approve endpoint allows users to approve bulk payment files by using file number. The request parameters and objects are specified below -

**Request Parameters**

**The fileNumber must be passed in the path of API request.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| fileNumber | path | string | true | none |

**JSON Response(JSON schema NA in the API doc provided by the client)**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

### **Reject File**

**PUT /bulkpayment/files/{fileNumber}/reject**

**HTTP Method: PUT**

**Description -** Sending a PUT request to the bulkpayment/files/{fileNumber}/reject endpoint allows users to reject Bulk Payment File. The request parameters and objects are specified below -

**Request Parameters**

**The fileNumber must be passed in the path of API request.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| fileNumber | path | string | true | none |

**JSON Response(JSON schema NA in the API doc provided by the client)**

**A successful PUT request to this endpoint returns the following data -**

**Response Code - 200**

## **API Name UserAdmin**

### **Applications API**

### **Get Application**

**GET /applications**

**HTTP Method: GET**

**Description -** Sending a GET request to GET/applications endpoint allows users to fetch the application details using parameters “type” and “status” as a query in the API request.

Request Parameters

**The Application Type and status are optional parameters to be passed in the query of API request.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| type | query | string | false | Application Type |
| status | query | string | false | Application Status |

Enumerated Values

| **Parameter** | **Value** |
| --- | --- |
| type | BANK |
| type | CUSTOMER |
| status | ACTIVE |
| status | INACTIVE |

**JSON Response**

**A successful GET request to this endpoint will return the following data -**

| {  "status": "Success or Failure",  "code": "APP001 etc.",  "message": "Failed to get an application: .",  "data": [  {  "appCode": "finzly.payment etc.",  "description": "Payment Hub etc.",  "status": "ACTIVE or INACTIVE",  "type": "BANK or CUSTOMER",  "entilements": [  {  "UUID": "string",  "entitlementCode": "aBlkFil etc.",  "description": "string"  }  ]  }  ] } |
| --- |

**Data Returned**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [[SearchApplication](https://apidocs.finzly.net/dashboard?java=#schemasearchapplication)] | false | none | none |

Get Application Details By App Code

**GET /applications/appcode/{app-code}**

**HTTP Method: GET**

**Description-** Sending a GET request to **/applications/appcode/{app-code}** endpoint allows users to get application details by the application code by providing app-code in the path of the API request.

**Request Parameters**

The App-code needs to be passed in the path of the API request

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| app-code | path | string | true | none |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "APP001 etc.",  "message": "Failed to get an application: .",  "data": [  {  "appCode": "finzly.payment etc.",  "description": "Payment Hub etc.",  "status": "ACTIVE or INACTIVE",  "type": "BANK or CUSTOMER",  "entilements": [  {  "UUID": "string",  "entitlementCode": "aBlkFil etc.",  "description": "string"  }  ],  "userLimits": [  {  "limitCode": "ACH\_USER\_BULK\_FILE\_APPROVAL\_LIMIT etc.",  "value": "10000 etc.",  "description": "ACH user bulk file approval limit etc.",  "appCode": "finzly.payments etc.",  "linkedEntitlementCode": "aBlkFil etc."  }  ]  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [Application] | false | none | none |

### **Roles**

**POST/searchRoleV2**

**HTTP Method: POST**

**Description:** Sending a POST request to /searchRoleV2 endpoint allows users to search roles created by business customers to support their internal users. All the request parameters and objects are mentioned below -

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [V2GetRoleRequest](https://apidocs.finzly.net/dashboard?java=#schemav2getrolerequest) | true | Search the roles using various parameters |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "ROLE001 etc.",  "message": "Failed to get roles: .",  "data": [  {  "roleUUID": "123e4567-e89b-12d3-a456-426614174000 etc.",  "roleName": "User Admin etc.",  "description": "User Administrative role etc.",  "type": "BANK or CUSTOMER",  "customerUID": 0,  "customerShortName": "XYZCO etc.",  "applications": [  {  "appCode": "finzly.payment etc.",  "status": "ACTIVE or INACTIVE",  "entilements": [  {  "entitlementCode": "string"  }  ],  "userLimits": [  {  "limitCode": "ACH\_USER\_BULK\_FILE\_APPROVAL\_LIMIT etc.",  "value": "10000 etc.",  "description": "ACH user bulk file approval limit etc.",  "appCode": "finzly.payments etc.",  "linkedEntitlementCode": "aBlkFil etc."  }  ]  }  ]  }  ],  "pagination": {  "totalRecords": 100,  "returnedRecords": 10,  "pageReturned": 2,  "pageSize": 10  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [[V2Role](https://apidocs.finzly.net/dashboard?java=#schemav2role)] | false | none | none |
| pagination | [V2Pagination](https://apidocs.finzly.net/dashboard?java=#schemav2pagination) | false | none | none |

**V2Role**

| {  "roleUUID": "123e4567-e89b-12d3-a456-426614174000 etc.",  "roleName": "User Admin etc.",  "description": "User Administrative role etc.",  "type": "BANK or CUSTOMER",  "customerUID": 0,  "customerShortName": "XYZCO etc.",  "applications": [  {  "appCode": "finzly.payment etc.",  "status": "ACTIVE or INACTIVE",  "entilements": [  {  "entitlementCode": "string"  }  ],  "userLimits": [  {  "limitCode": "ACH\_USER\_BULK\_FILE\_APPROVAL\_LIMIT etc.",  "value": "10000 etc.",  "description": "ACH user bulk file approval limit etc.",  "appCode": "finzly.payments etc.",  "linkedEntitlementCode": "aBlkFil etc."  }  ]  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| roleUUID | string | false | none | A UUID (Universal Unique Identifier) associated with the role |
| roleName | string | false | none | Name of the role |
| description | string | false | none | Role details |
| type | string | false | none | Type of the Role |
| customerUID | number | false | none | A unique id associated with the CRM entity |
| customerShortName | string | false | none | Short name associated with the CRM entity |
| applications | [V2App] | false | none | none |

### **Add Role V2**

**POST /v2/roles**

**HTTP Method: POST**

**Description -** Sending a POST request to POST /v2/roles endpoint allows users to add a new role. All the request parameters and objects are mentioned below -

Request Parameters

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [V2CreateRoleRequest](https://apidocs.finzly.net/dashboard?java=#schemav2createrolerequest) | true | Search the roles using various parameters |

**JSON Response**

Sending a successful POST request to this endpoint returns the following data -

| {  "status": "Success or Failure",  "code": "ROLE001 etc.",  "message": "Failed to add a role: .",  "data": {  "externalReferenceId": "for e.g. 126879",  "roleUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [V2CreateRoleResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemav2createroleresponse_data) | false | none | none |

**V2 Create Role Response Data**

| {  "externalReferenceId": "for e.g. 126879",  "roleUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| externalReferenceId | string | false | none | An external reference id |
| roleUUID | string | false | none | A UUID associated with the role |

### **Get Role**

**GET /v2/roles/{roleUUID}  
HTTP Method: GET**

**Description -** Sending a GET request to **/v2/roles/{roleUUID}** allows users to retrieve a role details by a role id by passing roleUUID in the path of the API request. All the request parameters and objects are mentioned below -

**Request Parameters**

**RoleUUID needs to be passed in the path of the API request.**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| roleUUID | path | string | true | A UUID (Universal Unique Identifier) associated with a role |
| type | query | string | false | Type of the role |

**Enumerated Values**

| **Parameter** | **Value** |
| --- | --- |
| type | CUSTOMER |
| type | BANK |

**JSON Response**

**A successful GET request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "ROLE001 etc.",  "message": "Failed to find a role: .",  "data": {  "roleUUID": "123e4567-e89b-12d3-a456-426614174000 etc.",  "roleName": "User Admin etc.",  "description": "User Administrative role etc.",  "type": "BANK or CUSTOMER",  "customerUID": 0,  "customerShortName": "XYZCO etc.",  "applications": [  {  "appCode": "finzly.payment etc.",  "status": "ACTIVE or INACTIVE",  "entilements": [  {  "entitlementCode": "string"  }  ],  "userLimits": [  {  "limitCode": "ACH\_USER\_BULK\_FILE\_APPROVAL\_LIMIT etc.",  "value": "10000 etc.",  "description": "ACH user bulk file approval limit etc.",  "appCode": "finzly.payments etc.",  "linkedEntitlementCode": "aBlkFil etc."  }  ]  }  ]  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [V2Role](https://apidocs.finzly.net/dashboard?java=#schemav2role) | false | none | none |

**V2Role**

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| roleUUID | string | false | none | A UUID (Universal Unique Identifier) associated with the role |
| roleName | string | false | none | Name of the role |
| description | string | false | none | Role details |
| type | string | false | none | Type of the Role |
| customerUID | number | false | none | A unique id associated with the CRM entity |
| customerShortName | string | false | none | Short name associated with the CRM entity |
| applications | [V2App] | false | none | none |

### **Update Role V2**

**PUT /v2/roles/{roleUUID}**

**HTTP Method: PUT**

**Description** - Sending a PUT request to PUT /v2/roles/{roleUUID} allows users to update a role by passing roleUUID in the path of the API request.

**Request Parameters**

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| roleUUID | path | string | true | A UUID (Universal Unique Identifier) associated with a role |
| body | body | [V2CreateRoleRequest](https://apidocs.finzly.net/dashboard?java=#schemav2createrolerequest) | false | Type of the role |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "ROLE001 etc.",  "message": "Failed to add a role: .",  "data": {  "externalReferenceId": "for e.g. 126879",  "roleUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [V2CreateRoleResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemav2createroleresponse_data) | false | none | none |

### **Delete Role**

**DELETE /v2/roles/{roleUUID}**

**HTTP Method: DELETE**

**Description -** Sending a DELETE request to /v2/roles/{roleUUID} endpoint allows users to delete a role by passing roleUUID in the path of the API request. All the parameters and objects are mentioned below -

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [V2GetRoleRequest](https://apidocs.finzly.net/dashboard?java=#schemav2getrolerequest) | true | Search the roles using various parameters |

**JSON Response**

**A successful DELETE request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "ROLE001 etc.",  "message": "Failed to delete a role: .",  "data": {  "roleUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000"  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [V2DeleteRoleResponse\_data](https://apidocs.finzly.net/dashboard?java=#schemav2deleteroleresponse_data) | false | none | none |

**V2 Delete Roles Response**

| {  "roleUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000" } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| roleUUID | string | false | none | A UUID associated with the role |

**USERS**

**Search User**

**POST /v2/users/search**

**HTTP Method: POST**

**Description -** Sending a POST request to **/v2/users/search endpoint allows users to** search users by the given request criteria. This API will return user profile data without role and user limit details. All the request parameters and objects are mentioned below -

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| body | body | [V2SearchUserRequest](https://apidocs.finzly.net/dashboard?java=#schemav2searchuserrequest) | true | Customer User |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "USER001 etc.",  "message": "Failed to get the users: .",  "data": [  {  "userUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000",  "userName": "for e.g. user001",  "firstName": "for e.g. John",  "lastName": "for e.g. Doe",  "department": "for e.g. payments",  "otherDepartments": [  "for e.g. [admin,BSA]"  ],  "customerUID": "1223",  "emulationUser": "YES or NO",  "customerShortName": "DEMOBANK",  "status": "Active or Suspended etc.",  "email": "test@bank.com",  "mobile": "12345678",  "type": "BANK or CUSTOMER",  "createdBy": "ABC001",  "lastUpdatedBy": "XYZ001"  }  ],  "pagination": {  "totalRecords": 100,  "returnedRecords": 10,  "pageReturned": 2,  "pageSize": 10  } } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [[V2SearchUser](https://apidocs.finzly.net/dashboard?java=#schemav2searchuser)] | false | none | none |
| pagination | [V2Pagination](https://apidocs.finzly.net/dashboard?java=#schemav2pagination) | false | none | none |

### **Get User V2**

**GET /v2/users/{userUUID}**

**HTTP Method: GET**

**Description -** Sending a GET request to this ID /v2/users/{userUUID} allows users to get the user by userUUID. This API will return user profile data along with role and user limit details. All the request parameters and objects are mentioned below -

| **Name** | **In** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- | --- |
| userUUID | path | string | true | A universally unique identifier (UUID) associated with the user |
| type | query | string | true | Type of the user |

**Enumerated Values**

| **Parameter** | **Value** |
| --- | --- |
| type | CUSTOMER |
| type | BANK |

**JSON Response -**

**A successful GET request this endpoint returns the following data -**

| {  "userUUID": "for e.g. 123e4567-e89b-12d3-a456-426614174000",  "userName": "for e.g. user001",  "firstName": "for e.g. John",  "lastName": "for e.g. Doe",  "department": "for e.g. payments",  "otherDepartments": [  "for e.g. [admin,BSA]"  ],  "customerUID": "1223",  "emulationUser": "YES or NO",  "customerShortName": "DEMOBANK",  "status": "Active or Suspended etc.",  "email": "test@bank.com",  "mobile": "12345678",  "type": "BANK or CUSTOMER",  "createdBy": "ABC001",  "lastUpdatedBy": "XYZ001",  "roles": [  {  "roleUUID": "123e4567-e89b-12d3-a456-426614174000",  "name": "AdminRole",  "description": "Bank admin permission",  "applications": [  {  "appCode": "for e.g. finzly.payment",  "description": "for e.g. Payment Hub application to process multiple payment rails",  "entitlements": [  {  "entitlement": "for e.g. ApprovePayment",  "description": "Allow user to approve the payment"  }  ]  }  ]  }  ],  "userlimits": [  {  "limitCode": "ACH\_USER\_BULK\_FILE\_APPROVAL\_LIMIT etc.",  "value": "10000 etc.",  "description": "ACH user bulk file approval limit etc.",  "appCode": "finzly.payments etc.",  "linkedEntitlementCode": "aBlkFil etc."  }  ] } |
| --- |

| **Name** | **Type** | **Required** | **Restrictions** | **Description** |
| --- | --- | --- | --- | --- |
| userUUID | string | false | none | A universally unique identifier (UUID) assigned by the system |
| userName | string | false | none | Unique username assigned to the user |
| firstName | string | false | none | User First Name |
| lastName | string | false | none | User Last Name |
| department | string | false | none | Primary bank department user is associated with |
| otherDepartments | [string] | false | none | none |
| customerUID | string | false | none | Unique id associated with the CRM entity |
| emulationUser | string | false | none | If the user has an emulation capability. This is applicable for CUSTOMER user types. |
| customerShortName | string | false | none | Customer short name |
| status | string | false | none | User Status |
| email | string | false | none | Customer email address |
| mobile | string | false | none | Customer mobile contact number |
| type | string | false | none | Type of user |
| createdBy | string | false | none | User Id who created the user in the system |
| lastUpdatedBy | string | false | none | User who last updated the user in the system |
| roles | [[V2UserRole](https://apidocs.finzly.net/dashboard?java=#schemav2userrole)] | false | none | none |
| userlimits | [[V2UserLimits](https://apidocs.finzly.net/dashboard?java=#schemav2userlimits)] | false | none | none |

### **Update User**

**PUT /v2/users**

**HTTP Method: PUT**

**Description -** Sending a PUT request to PUT/v2/users allows users to update an existing user. The request parameters and objects are mentioned below -

**Request Parameters**

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| body | body | [V2UpdateUserRequest](https://apidocs.finzly.net/dashboard?java=#schemav2updateuserrequest) | false | Customer User |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "USER001 etc.",  "message": "Failed to add an user: .",  "data": {  "externalReferenceId": "string",  "userUUID": "123e4567-e89b-12d3-a456-426614174000"  } } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [V2AddUserResponseData](https://apidocs.finzly.net/dashboard?java=#schemav2adduserresponsedata) | false | none | none |

**V2 Add User Response**

| {  "externalReferenceId": "string",  "userUUID": "123e4567-e89b-12d3-a456-426614174000" } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| externalReferenceId | string | false | none | Unique reference id from a system outside of the Finzly system. |
| userUUID | string | false | none | A universally unique identifier (UUID) assigned to the user by the system |

### **Add User**

**POST /v2/users**

**HTTP Method: POST**

**Description -** Sending a POST request to POST/v2/users endpoint allows users to add a new user to the system. The request parameters and objects are mentioned below

Request Parameters

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| body | body | [V2UpdateUserRequest](https://apidocs.finzly.net/dashboard?java=#schemav2updateuserrequest) | false | Customer User |

**JSON Response**

**A successful POST request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "USER001 etc.",  "message": "Failed to add an user: .",  "data": {  "externalReferenceId": "string",  "userUUID": "123e4567-e89b-12d3-a456-426614174000"  } } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |
| data | [V2AddUserResponseData](https://apidocs.finzly.net/dashboard?java=#schemav2adduserresponsedata) | false | none | none |

**V2 Add User Response**

| {  "externalReferenceId": "string",  "userUUID": "123e4567-e89b-12d3-a456-426614174000" } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| externalReferenceId | string | false | none | Unique reference id from a system outside of the Finzly system. |
| userUUID | string | false | none | A universally unique identifier (UUID) assigned to the user by the system |

### **Activate User**

**PUT /v2/users/{userUUID}/activate  
HTTP Method: PUT**

**Description -** Sending a PUT request to the PUT /v2/users/{userUUID}/activate endpoint allows users to activate a user. The request parameters and objects are mentioned below -

**Request Parameters**

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| userUUID | path | string | true | A universally unique identifier (UUID) assigned to the user |
| type | query | string | true | user type |

**Enumerated Values**

| **Parameter** | **Value** |
| --- | --- |
| type | BANK |
| type | CUSTOMER |

**JSON Response**

**A successful PUT request to this endpoint returns the following data**

| {  "status": "Success or Failure",  "code": "USER001 etc.",  "message": "Failed to activate or suspend the user: ." } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |

**Suspend User V2**

**PUT /v2/users/{userUUID}/suspend**

**HTTP Method: PUT**

**Description -** Sending a PUT request to /v2/users/{userUUID}/suspend endpoint allows users to suspend a user by passing userUUID in the path of the API request. The request parameters and objects are mentioned below -

Request Parameters

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| userUUID | path | string | true | A universally unique identifier (UUID) assigned to the user |
| type | query | string | true | user type |

**Enumerated Values**

| **Parameter** | **Value** |
| --- | --- |
| type | BANK |
| type | CUSTOMER |

**JSON Response**

**A successful PUT request to this endpoint returns the following data -**

| {  "status": "Success or Failure",  "code": "USER001 etc.",  "message": "Failed to activate or suspend the user: ." } |
| --- |

| Name | Type | Required | Restrictions | Description |
| --- | --- | --- | --- | --- |
| status | string | false | none | This will return Success/Failed |
| code | string | false | none | This will indicate the error code in case of API error |
| message | string | false | none | This will be the detailed error message indicating what failed and how to fix the issue |

# **Payment Status Notifications v1.2.0**

Security protocols support basic authentication by Finzly. This API allows third parties and bank customers to receive payment details whenever there is an update to the payment system. Finzly can also share the FinzlyOS public IPs with third parties or bank customers to whitelist at their end for additional security.

Base URLs:

* [https://{tenant}-security-{environment}.{dns}/auth/realms/BANKOS.UAT.SANDBOX.CUSTOMER/protocol/openid-connect/token](about:blank)
  + **tenant** - Tenant Code Default: sandbox
  + **environment** - Environment Name Default: uat
  + **DNS** - DNS Name Default: [finzly.io](http://finzly.io/)

# Payment Notifications

Payment Notification to third parties whenever there is an update to the payment status in the Finzly payment system.

## **notifyPaymentStatus**

**POST /payments**

**HTTP Method: POST**

**Description -** Payment notification details, including payment Id, source reference, and status, are pushed via this webhook whenever there is an update to the payment status.

# Notify payment information to the client or third party.

| Name | In | Type | Required | Description |
| --- | --- | --- | --- | --- |
| body | body | [payment](https://apidocs.finzly.net/dashboard?java=#schemapayment) | true | none |

### **Errors**

**Http Response Codes**

| Code | Reason |
| --- | --- |
| 200 - OK | Successfully processed the request |
| 201 - Created | The object was successfully created |
| 204 - No Content | No Content, Successfully processed the request |
| 400 - Bad Request | The request was rejected, most likely due to invalid parameters |
| 401, 403 - Unauthorized, Forbidden | Unauthorized or forbidden request |
| 404 - Not Found | The resource does not exist |
| 409 - Conflict | This typically occurs when duplicate requests are received even while the original request is being processed. |
| 422 - Unprocessable Entity | The request body contains well-formed (i.e., syntactically correct), but semantically erroneous, instructions. |
| 500 - Internal Server Error | Something went wrong with Digital Disbursements and the request could not be processed. |

**400 Bad Request**

The 400 (Bad Request) status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

**402 Payment Required**

The 402 (Payment Required) status code is reserved for future use.

**403 Forbidden**

The 403 (Forbidden) status code indicates that the server understood the request but refuses to authorize it. A server intending to make the reason for the forbidden request public can describe it in the response payload (if any).

If authentication credentials were provided in the request, the server considers them insufficient to grant access. The client SHOULD NOT automatically repeat the request with the same credentials. The client MAY repeat the request with new or different credentials. However, a request might be forbidden for reasons unrelated to the credentials.

An origin server that wishes to "hide" the current existence of a forbidden target resource MAY instead respond with a status code of 404 (Not Found).

**404 Not Found**

The 404 (Not Found) status code indicates that the origin server did not find a current representation for the target resource or is not willing to disclose that one exists. A 404 status code does not indicate whether this lack of representation is temporary or permanent; the 410 (Gone) status code is preferred over 404 if the origin server knows, presumably through some configurable means, that the condition is likely to be permanent.

A 404 response is cacheable by default, i.e., unless otherwise indicated by the method definition or explicit cache controls (see [Section 4.2.2 of [RFC7234]](https://www.rfc-editor.org/rfc/rfc7234#section-4.2.2)).

**405 Method Not Allowed**

The 405 (Method Not Allowed) status code indicates that the method received in the request line is known by the origin server but not supported by the target resource. The origin server MUST generate an Allow header field in a 405 response containing a list of the target resource's currently supported methods.

A 405 response is cacheable by default, i.e., unless otherwise indicated by the method definition or explicit cache controls

**406 Not Acceptable**

The 406 (Not Acceptable) status code indicates that the target resource does not have a current representation that would be acceptable to the user agent, according to the proactive negotiation header fields received in the request and the server is unwilling to supply a default representation.

The server SHOULD generate a payload containing a list of available representation characteristics and corresponding resource identifiers from which the user or user agent can choose the most appropriate. A user agent MAY automatically select the most suitable choice from that list.

## **Guides**

### **Technical guides**

**Getting started with Finzly APIs**

Finzly Support creates your credentials, users in your team, and API Keys.

Using the API keys, you can access the Finzly API.

Now you can invoke the APIs to test the use cases. To commence your evaluation, you will require an evaluation and integration environment set up for you.

**Sandbox and testing**The sandbox environment is an environment you can play around with and use as a testbed for development. Customers must test their business flows in this environment before moving to the production environment. You must use test customers and test account numbers in the sandbox environment. Your business bank will onboard you into the sandbox environment and provide you with the API credentials.

After testing and validating the use cases to your satisfaction, you can sign off on evaluation.

Please contact the Finzly support team to validate the integration.

### **Use Cases**

Finzly APIs simplify the trading and payment of foreign currencies with an exceptional user experience. Banks can use Finzly APIs to integrate new products and services with the bank’s existing systems. That allows for quick and seamless integration. If a bank wishes to create a new product or service, Finzly APIs make it a much more straightforward process.

**Streamlining Customer Onboarding Processes for Banks -** Finzly APIs help the bank clients to accelerate their customer onboarding process.That allows our bank clients to win new customers, vastly enhance the experience of customers applying for their products and service, and sell further services promptly.

Customer onboarding is the next step after your customers have registered for the bank. Its objective is to allow your customers to feel comfortable utilizing your services and create a strong bond of trustworthiness. Usually, the customer onboarding procedure may include the following:

* Step-by-step instructions for account usage.
* Prefaces to your support team for further assistance.
* Landmark celebrations for customer engagement and satisfaction.

**Benefits of accelerated onboarding**

* Having clear policies and procedures highlights to customers that an institution is not only aware of and actively trying to win their customers but also that they are compliant with regulations and within the confines of the law.

Illustrating to customers how a bank is actively working with regulations secures its reputation as a trustworthy institution that will keep money and other assets safe.

* Quick and efficient onboarding allows financial institutions less lead-in time to begin selling their services.
* Once a bank has boarded a customer, they will look to sell them several different products across different service areas. Comprehensively onboarding a customer rigorously allows other service areas to offer their products as soon as the customer has been onboarded. That also impacts the reputation of the institution in the customer’s eyes.

**Insurance Claims**

Finzly APIs offer many ways to process premium payments and claims by embedding APIs. It offers real-time payment services for insurance firms, which help them convert repetitive payment tasks into reusable ones that foster productivity. The results are improved efficiency for insurers and a more satisfactory customer experience who benefit from speedier service enabled by streamlined payment and claims processes.

**Fintech Apps & Neobanks**

**Allow fintechs to Send & receive payments, create Virtual Accounts & manage accounts**

**Bank Account validation.**

Account Validation is a vital facet of an organization’s risk assessment. Understanding their customers’ banking status will allow our clients to eliminate fraud and counter risks related to payment acceptance.

With credit cards, companies can approve or reject a transaction depending on the credit limit during payment time and set aside funds to cover the payments.

**Gaming Platforms**

Power payments for gaming platforms by allowing payments through API

**E-com Payments**

Power instant checkouts through bank transfers by offering payment API for real-time payments and refunds.

The payment API is powered by the routing engine (rule-based workflow) to make it easy to understand. The consumer does not have to choose or decide on the payment rail or network (such as ACH, RTP FedNow, etc.) to create a payment. Instead, they have to choose from the simple options based on the payment speed and cost, for instance, "Economy" for less cost and 2 to 3 business days to transfer the money or "Instant" for a little higher price and instant money transfer.

**Features and Benefits**

* API supports both domestic as well as international payments using One API. Users can send & receive instant payments on the RTP & FedNow networks using one application. Offer value-added services using Request for Pay.
* Payment Scheduling - It allows the user to schedule a payment to a future date
* Repetitive Payments - It provides various options to set repetitive payments such as Daily, Weekly, Bi-Weekly, Monthly, etc.
* A simplified and improved payment experience with a single self-service portal with a customizable Super App for SMB, Commercial & Institutional clients.

**Invoice Payments**

Finzly empowers invoice processing firms to handle millions of invoices annually with a simple API integration to their existing channels

**Logistics Platforms**

Simplify supply chain payments by providing bank transfers, instant and worldwide payments.

The emergence of fast payments in many financial institutions and the increasing prevalence of new proposals and implementations show essential advancements in the demand for and supply of such payments. Finzly API makes the provision and use of fast payments increasingly viable. In addition, Finzly APIs have also redefined end-user expectations for the speed and convenience of payments.

**Incoming Debit Reviews**

Positive Pay services allow the bank and the bank's clients (via APIs or Finzly Treasury Experience) to enable rules to determine if ACH debits should be paid against the receiver's bank account. The service allows users to review all incoming debits before they're cleared to settle against their account or rejected and returned. The bank can use the Positive Pay feature to determine the "decision cutoff" time and a default decision to pay or reject any debits.

**Wealth Management**

Finzly APIs enable wealth management firms to offer direct account-to account payments without unnecessary interchange fees.

**Accounting Help**

Finzly APIs empower businesses to streamline reconciliation and enable real time cash flow visibility with instant payment notifications.

## **Payment Networks**

In this section, we’ll talk about the different payment networks Finzly APIs support.

* ACH (Automated Clearing House)
* WIRES
* FEDNOW (Fedwire Funds Services)
* RTP
* International Payments

### **ACH**

ACH (Automated Clearing House) is the most popular, and highly used payment network that was first launched in [xxxx]. ACH payments are exchanged in NACHA formatted files. NACHA sets the rules and guidelines for the processing of ACH payments. After the power of RESTful APIs were proved to be critical for a connected e-commerce and treasury, the need to offer ACH payments via REST API has become critical for the success of banks.

**ACH Terminologies**

While ACH payments are cheaper compared to other payment networks, they have most the complexity than other payment rails, as ACH has evolved over several decades to support various use cases of the industry.

**Files & Batches**

Each payment is created as a payment entry, and the entries are grouped in batches and the batches are grouped in a file. Each file and batch may be balanced or unbalanced.

**Balanced & Unbalanced Files**

In a balanced file, the outbound debits and credits are offset with a customer’s DDA account entry. In an unbalanced file, the bank would offset it with the customer’s default account.

**SEC Codes**

Each payment entry must be categorized under one of the several Service Entry Class Codes, aka SEC Codes. Finzly supports all the SEC Codes as listed below

| **SEC Code** | **Description** | **Use** | **Debit / Credit** | **Consumer / Corporate** |
| --- | --- | --- | --- | --- |
| ARC | Accounts Receivable Entries | A single ACH debit used by originator for the conversion of an eligible source document received via the U.S. mail or delivery service; at a lockbox location; or in person at a manned location for the payment of a bill. | Debit | Consumer to Corporate |
| ACK | ACH Payment Acknowledge | RDFI to acknowledge the receipt of ACH credit payments originated using the CCD formats | Non-Monetary | N/A |
| ADV | Automated Accounting Advice | identifies automated accounting advice of ACH accounting information in machinereadable format to facilitate the automation of accounting information for participating DFIs | Non-Monetary | N/A |
| ATX | Financial EDI Acknowledgment | RDFI to acknowledge the receipt of ACH credit payments originated using the CTX formats | Non-Monetary | N/A |
| BOC | Back Office Conversion | A single ACH debit initiated by an originator based on an eligible source document provided at the point-of-purchase or at a manned bill payment location for subsequent conversion during back-office conversion. | Debit | Consumer to Corporate |
| CCD | Corporate Credit or Debit | A single or a recurring ACH credit or debit originated to a corporate account. They are commonly used by Originators to pay vendors, concentrate funds from outlying accounts (cash concentration), to fund payroll, petty cash, or other disbursement accounts. | Credit / Debit/ Nonmonetary | Corporate to Corporate |
| CIE | Customer Initiated Entries | A credit entry initiated by consumer through a bill payment service provider to pay bills. | Credit | Consumer to Corporate |
| CTX | Corporate Trade Exchange | A single or a recurring ACH credit or debit originated to a corporate account that supports up to 9,999 addenda records. They are commonly used in trading partner relationships because a full ANSI ASC X12 message or payment-related UN/EDIFACT information can be sent with the CTX entry. | Credit / Debit/ Nonmonetary | Corporate to Corporate |
| IAT | International ACH Transaction | An Entry that is part of a payment transaction involving a financial agency’s office that is not located within the territorial jurisdiction of the United States. | Credit / Debit | Corporate to Consumer/ |
| POP | Point of Purchase | A single ACH debit initiated by an originator based on an eligible source document provided at a point-of-purchase or manned bill payment location. | Debit | Corporate to Consumer/ |
| POS | Point of Sale | Initiated at an electronic terminal using merchant issued physical card or other access device | Debit | Consumer |
| PPD | Pre-arranged Payment or Deposit | A single or a recurring ACH credit or debit sent by an originator to a consumer account to make or collect a payment, where authorization is obtained in writing. | Credit / Debit | Corporate to Consumer |
| RCK | Re-presented Check Entries | Originators can re-present a check that has been processed through the check collection system as a paper or image item, and returned because of insufficient or uncollected funds | Debit | Corporate to Consumer |
| TEL | Telephone Initiated Entries | A single or a recurring ACH debit that occurs when the consumer’s authorization for a transfer of funds is received orally via the telephone. | Debit | Corporate to Consumer |
| WEB | Internet-Initiated/Mobile Entries | Credit WEB Entries: A Person-to-Person entry transmitted on behalf of one natural person to another natural person, or between accounts belonging to the same natural person.   Debit WEB Entries: ACH entry initiated according to an authorization obtained via the internet or a wireless network (e.g. mobile device) except for an oral authorization via a telephone call. | Credit / Debit | Corporate to Consumer |
| DNE | Death Notification Entry | Federal Govt. Agency Use Only | Non-Monetary | Consumer |

***Note: Finzly payment hub system support all above sec codes from RDFI perspective. However, from the ODFI perspective the following sec codes are supported BOC, CCD, CIE, CTX, IAT, POP, POS, PPD, TEL and WEB.***

**Addenda Record**

Each payment entry carries the additional information required to process the payment, and these addenda records are different for various SEC Codes.

**ODFI**

The financial institution that originates or sends the payment file to FED is called Originating Depository Financial Institution.

**RDFI**

The financial institution that receives the payment file from FED is called Receiving Depository Financial Institution.

**ACH Transactions**

**Credit Request**

Credit Request allows ODFI to send a payment to RDFI. RDFI upon receiving the credit request would either accept the payment and credit its customer or return the payment.

**Debit Request**

Debit Request allows ODFI to debit request to RDFI. RDFI upon receiving the debit request would either accept the request and debit its customer or return the request.

**Notice of Change**

In case the receiving party’s information is changed for a Credit or Debit request, RDFI may send Notice of Change (NOC) to ODFI.

**Returns**

When the RDFI returns the credit or debit request, the ODFI may accept the return or dishonor the return.

|  | **Return Code** |  |  | **Return Title** |  |  | **Return Description** |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | R01 |  |  | Insufficient Funds |  |  | The available and/or cash reserve balance is not |  |
|  |  |  |  |  | sufficient to cover the dollar value of the debit Entry. |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| R02 | | |  | Account Closed | |  | A previously active account has been closed by action | |
|  |  | of the customer or the RDFI | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  | The account number structure is valid and it passes |  |
|  |  |  |  | No Account/ Unable to Locate |  |  | the Check digit validation, but the account number |  |
|  | R03 |  |  |  |  | does not correspond to the individual identified in the |  |
|  |  |  | Account |  |  |  |
|  |  |  |  |  |  | Entry, or the account number designated is not an |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | existing account |  |
|  |  |  |  |  |  |  |  |  |
| R04 | | |  | Invalid Account Number Structure | |  | The account number structure is not valid | |
|  |  |  |  |  |  |  |  | |
|  |  |  |  | Unauthorized Debit to Consumer |  |  | CCD or CTX debit Entry was transmitted to a Consumer |  |
|  | R05 |  |  |  |  | Account of the Receiver and was not authorized by the |  |
|  |  |  | Account Using Corporate SEC Code |  |  |  |
|  |  |  |  |  |  | Receiver |  |
|  |  |  |  |  |  |  |  |
| R06 | | |  | Returned per ODFIs Request | |  | The ODFI has requested that the RDFI return an | |
|  |  | Erroneous Entry. | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  | The RDFIs customer (the Receiver) revoked the |  |
|  | R07 |  |  | Authorization Revoked by Customer |  |  | authorization previously provided to the Originator for |  |
|  |  |  |  |  |  |  | this debit Entry. |  |
| R08 | | |  | Payment Stopped | |  | The Receiver has placed a stop payment order on this | |
|  |  | debit Entry | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  | A sufficient ledger balance exists to satisfy the dollar |  |
|  | R09 |  |  | Uncollected Funds |  |  | value of the transaction, but the available balance is |  |
|  |  |  |  |  |  |  | below the dollar value of the debit Entry |  |
|  |  |  |  | Customer Advises Not Authorized, | |  | The RDFI has been notified by the Receiver that the | |
| R10 | | |  | Improper, Ineligible, or Part of an | |  | Entry is unauthorized, improper, ineligible, or part of | |
|  |  |  |  | incomplete transaction | |  | an incomplete transaction. | |
|  |  |  |  |  | |  |  |  |
|  |  |  |  | Customer Advises entry not in |  |  | Error or defect in payment between originator and |  |
|  | R11 |  |  | accordance with the terms of |  |  |  |
|  |  |  |  |  | receiver |  |
|  |  |  |  | Authorization |  |  |  |
|  |  |  |  |  |  |  |  |
| R12 | | |  | Account Sold to Another DFI | |  | A financial institution received an Entry to an account | |
|  |  | that was sold to another financial institution. | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |
|  |  |  |  | Representative Payee Deceased or |  |  | The representative payee is either deceased or unable |  |
|  | R14 |  |  |  |  | to continue in that capacity. The beneficiary is NOT |  |
|  |  |  | Unable to Continue in that Capacity |  |  |  |
|  |  |  |  |  |  | deceased. |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  | Beneficiary or Account Holder | |  | (1) The beneficiary is deceased, or (2) The account | |
| R15 | | |  | (Other Than a Representative | |  |
|  |  | holder is deceased. | |
|  |  |  |  |  | |  |

| R16 | Account Frozen/Entry Returned per OFAC Instruction | OFAC has instructed the RDFI to return the Entry |
| --- | --- | --- |
| R17 | File Record Edit Criteria | An RDFI would use R17 to return an entry that does not have a valid account number and the RDFI believes the entry was initiated under questionable circumstances. |
| R20 | Non-Transaction Account | ACH Entry to a Non-Transaction Account |
| R21 | Invalid Company Identification | The identification number used in the Company Identification Field is not valid |
| R22 | Invalid Individual ID Number | The Receiver has indicated to the RDFI that the number with which the Originator was identified is not correct. |
| R23 | Credit Entry Refused by Receiver | Any credit Entry that is refused by the Receiver may be returned by the RDFI. |
| R24 | Duplicate Entry | The RDFI has received what appears to be a duplicate entry; i .e., the trace number, date, dollar amount and/or other data matches another transaction. |
| R26 | Mandatory Field Error | Improper information in one of the mandatory fields |
| R29 | Corporate Customer Advises Not Authorized | he RDFI has been notified by the Receiver (nonconsumer) that the Originator of a given transaction has not been authorized to debit the Receiver account. |
| R31 | Permissible Return Entry (CCD and CTX only) | The RDFI may return a CCD or CTX Entry that the ODFI agrees to accept. |
| R61 | Misrouted Return | The financial institution preparing the Return Entry (The RDFI of the original Entry) has placed the incorrect Routing Number in the Receiving DFI Identification field. |
| R67 | Duplicate Return | The ODFI has received more than one Return for the same Entry. |
| R68 | Untimely Return | The Return Entry has not been sent within the timeframe established by these Rules. |
| R69 | Field Error(s) | One or more of the field requirements are incorrect |
| R70 | Permissible Return Entry Not Accepted/ Return Not Requested by ODF | The ODFI has received a Return Entry identified by the RDFI as being returned with the permission of, or at the request of, the ODFI, but the ODFI has not agreed to accept the Entry or has not requested the return of the Entry. |
| R71 | Misrouted Contested Dishonored Return | The financial institution preparing the dishonored Return Entry (the ODFI of the original Entry) has placed the incorrect Routing Number in the Receiving DFI Identification field. |
| R72 | Untimely Dishonored Return | The dishonored Return Entry has not been sent within the designated timeframe. |
| R73 | Timely Original Return | The RDFI is certifying that the original Return Entry was sent within the timeframe designated in these Rules. |
| R74 | Corrected Return | The RDFI is correcting a previous Return Entry that was dishonored using Return Reason Code R69 (Field Errors) because it contained incomplete or incorrect information. |
| R75 | Return Not a Duplicate | The Return Entry was not a duplicate of an Entry previously returned by the RDFI. |
| R76 | No Errors Found | The original Return Entry did not contain the errors indicated by the ODFI in the dishonored Return Entry. |

**On-us Transactions**

On-us transactions in ACH is when a bank’s customer initiates ACH transfers for which the bank is both ODFI and RDFI. For such transactions, bank can settle within its ledger without having to clear thru ACH network.

**Same-day and Standard ACH**

ACH payments are usually processed with 2 days settlement, which means if a payment is initiated, it would take two business days for the payment to get credited in the customer account. In 20xx, NACHA required the clearing houses to support same-day ACH payments to allow same day processing and settlement. For some weird reason, several financial institution’s ACH system does not support same day ACH and can only process regular ACH payments.

**Cutoff Times**

The Federal Reserve processes the files on all bank working days from x:00 to x:00. Same day ACH files are transmitted at the following times

Standard ACH files are transmitted at the following times…

**ACH Positive Pay**

Positive Pay services allow the bank, and the bank's clients (via APIs or Finzly Treasury Experience) to enable rules to determine if ACH debits should be paid against the receiver's bank account. The service allows users to review all incoming debits before they’re cleared to pay against their account or rejected and returned. The bank can define a “decision cutoff” time, along with a default decision to pay or reject any debits that have not been decisioned.

**Block List**

The Block List is a service which provides the user the ability to identify specific ACH debit entries to be "stopped" and either returned automatically or to be manually decisioned as to whether the item should be returned or posted.

**Prefund/Non-Prefund**

To support the ACH Prefunding requirements for ACH Credits, Payment Galaxy provides a Prefunding capability, which causes the originator's offset account to be debited on the day their ACH file is processed, instead of waiting until the effective date of the ACH credits. Prior to routing the ACH credits to the FedACH network, the system will verify sufficient funds are available in the originator's account to fund the ACH credits. Alternatively, prefunding can be disabled, allowing for settlement to the originator's account on the effective date. For non-prefunded originators, you can assign exposure limits to monitor the overall amount of exposure to the bank.

**ACH Bulk Upload**

Finzly system supports the ACH bulk payment processing in a NACHA format via FTP channel. The bank can create an FTP account to process the ACH bulk file in a Nacha format. The solution provided two types of FTP account.

**Customer:** To be used by the corporate or Fintech to upload the bulk payments

**Channel:** To be used by the bank’s system to upload the payments in bulk associated with the multiple customers.

**Summary**

NACHA has various rules around each transaction processing. ACH is very complex to develop than other payment rails, but Finzly has made them simple with APIs. Finzly also offers file uploads of NACHA formatted files and returns NACHA formatted response files.

### **FEDWIRE**

Fedwire Funds Services (Fedwire) is a real-time gross settlement transfer system that allows participating financial institutions to send and receive same-day fund transfers. Federal Reserve operates the FEDWIRE payment network and is typically used to process high value payments or the payments that are required to be delivered faster, typically within an hour. Note: There could be delay due to compliance and other bank policies.

The Fedwire Funds Service is a credit transfer service. Participants originate funds transfers by instructing a Federal Reserve Bank to debit funds from its own account and credit funds to the account of another participant. Participants may originate funds transfers online, by initiating a secure electronic message, or offline, via Teller, Phone-in.

Fedwire funds are sent directly from the originator's bank account to the receiver's bank account. Wires are instant, but the fee can be a configurable data for both sender and receiver. The transaction is permanent and irrevocable with an option to request for Reversal (Return of Funds).

FEDWIRE is a message-oriented payment system. Each payment travels in a single message and is gross settled, whereas ACH payments are net settled.

Fedwire operates weekdays, from 9 p.m. the prior calendar day, to 7 p.m. Eastern Time. The system is closed during weekends and federal holidays, with the exception being that service for Mondays are operational from 9 p.m. on the preceding Sunday. Though operational until 7 p.m. ET, the deadline for initiating a payment for a bank’s customer, is 6 p.m. ET.

Reversals - Cut off time for Reversals are separately maintained as configurable property. Any Reversal after that cut off will not be eligible and will have to reverse the next business day

**Business Codes**

FEDWIRE payments are categorized into the following business codes.

* CTR - Customer or Corporate Credit Transfer (Where Beneficiary is a Customer - Consumer

/Corporate)

* BTR - Bank to Bank Transfer (where Beneficiary is Bank)
* CTP - Customer Transfer Plus (Where Beneficiary is a Customer - Consumer /Corporate with Additional remittance information)
* DRC - Customer or Corporate Drawdown Request
* DRB - Bank to Bank Drawdown Request
* DRW - Drawdown Payment
* SVC - Service message

### **Wire Origination**

Fedwire Funds service supports the following origination gateways for processing through BankOS

- API – Finzly Connect APIs

* Supports the initiation of
  + Customer or Corporate credit transfer
  + Customer Transfer Plus
  + Customer or Corporate Drawdown Request

- Web - Online Digital Treasury Portal (CashOS)

* Supports the initiation of domestic wire transfer
  + This will be defined based on the Processing speed as Express (Configurable) and Fees to be applied.

- Branch - Teller Application (Wire Room)

* Supports the initiation of Wire from branch using the teller application with an option as
  + Walk-in
  + Phone-in

- Bulk uploads

* Supports the initiation of wire with an option from Bulk File processing
  + Fedwire message upload
    - This will be based on the actual raw Fedwire message structure.
    - Multiple Fedwire message can be uploaded in a File

- Finzly CSV

* This is customed CSV formatted file for processing the Fedwire payments with required information as
  + Sender (Originator details)
  + Sender A/c o Receiver (Beneficiary details)
  + Receiver A/c
  + Additional Notes
* Finzly CSV supports the following Business Function code
  + Customer or Corporate credit transfer
  + Customer or Corporate Drawdown Request

### **Payment Status Updates**

### Payment Status Notification for every event on the wire life cycle will be notified. This will be applied for both outgoing and incoming payments.

### Validation Failed - Fedwire Payment with the Exception status will be notified.

### OFAC Review/Reject - Fedwire Payment with OFAC compliance Exception will be notified.

### Memo Post Status - Fedwire Payment with memo post status failure will be notified.

### Processed - Notification back to the Customer on payment Processing.

### **Drawdown Initiation**

### Capability to initiate a Drawdown request for both Customer and as well as for the Bank for their treasury Funds/Fed Transfers.

* + Drawdown initiation is possible for customer using the open Banking API and Bulk Files.
  + Every Drawdown request initiated will be received as a Separate drawdown payment linking the original payment.

### **Notifications**

Fedwire Incoming payment notification will be triggered with a source of Webhook notification. Customers can configure to receive these notifications. All the Inbound wire can be transmitted with a Raw Wire File message file received from Fed through the FTP source. Customer can configure an Inbound FTP account to receive the actual raw messages.

Inbound Drawdown request are triggered with E-mail notifications for the Bank to take an action on the drawdown requests which does not match based on the agreement set by the customer as part of the Finzly positive rules feature. This notification will be user friendly action for Bank to take an action on the Pending request which does not match to automatically create a drawdown payment out.

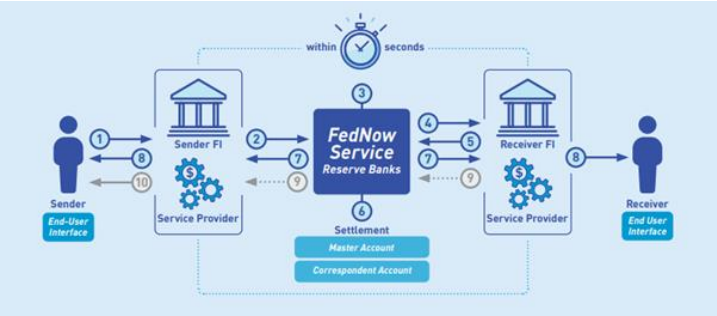
### **Instant Payments**

While ACH is used for high volume or low value payments, and FEDWIRE is used for faster or high value payments, the new instant payment networks such as RTP and FEDNOW are emerging to enable instant payments that are processed in real-time within seconds and are irrevocable. It is important to note that Debit cards enable instant payments, those transactions are comparatively expensive than RTP or FEDNOW payments and requires a debit card in addition to the bank account. Finzly offers instant payments from a bank account to another bank account via either RTP or FEDNOW, based on the sender and receiver bank’s connectivity.

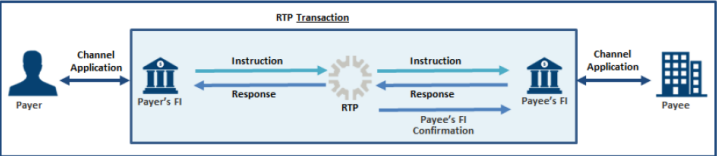
While there are other payment networks such as Zelle, Venmo, Cash App enable real-time settlement of funds at user’s accounts, these payment rails operate on top of other clearing networks such as ACH, RTP, FEDNOW or Card networks.

The following diagram explains the fund flow in FedNow and RTP payment network.

**FedNow:**



**RTP:**

****

Finzly Connect offers instant payment access and notifications via simple REST API. Develop new financial products such as Point of Sale, QR based invoices, and get paid instantly from your customer’s bank and eliminate card transaction fees.

### **International Payments**

The commerce is now global, and the businesses have to routinely send and receive payments to and from global vendors and customers. Finzly Connect allows making global payments, tracking and payment notifications via API easier.

The domestic payment networks such as ACH, FEDWIRE, RTP and FEDNOW makes interborder settlement easier, but the international payments have more stringent compliance requirements and is more complex than domestic payments.

To process international payments, your bank should either be connected with a correspondent bank who have SWIFT connection, or your bank should have direct SWIFT connectivity. Finzly directly connects with SWIFT to settle and clear international payments for your bank.

Finzly Connect, with a single API, powers you to send international payments to any recipient in the globe and receive payment notifications from your bank account for easy and automatic reconciliation.

## **Common Workflows**

### **Book Transfers**

Book transfers are used to move money within a bank or within a ledger. These transfers do not require using any payment rail.

#### **Book Transfer Request**:

| **Attribute** | **Type** | |  | **Required** | | | **Restrictions** | | | **Notes** |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| externalReferenceId | string | |  | TRUE | |  | Max Limit – 50 | | | Unique reference id from a system |  |
|  |  |  |  |  |  |  | chars | |  | outside of finzly. The external |  |
|  |  |  |  |  |  |  |  |  |  | reference id can be used by the |  |
|  |  |  |  |  |  |  |  |  |  | finzly for the request tracing (if |  |
|  |  |  |  |  |  |  |  |  |  | needed). |  |
| senderAccount | [BookTransferSender](https://apidocs.finzly.net/dashboard#schemav3booktransfersender) | | | TRUE | |  | Required | |  | Sender account details | |
|  |  |  | |  |  |  |  |  |  |  |  |
| receiverAccount | [BookTransferReceive](https://apidocs.finzly.net/dashboard#schemav3booktransferreceiver) | | | TRUE | |  | Required | |  | Receiver account details |  |
|  | [r](https://apidocs.finzly.net/dashboard#schemav3booktransferreceiver) | |  |  |  |  |  |  |  |  |  |
| paymentAmount | number | |  | TRUE | |  | Greater than | |  | Payment amount | |
|  |  |  |  |  |  |  | Zero | |  |  |  |
| paymentDate | string(date) | |  | TRUE | |  | Valid date | |  | Payment date in mm-dd-yyyy |  |
|  |  |  |  |  |  |  | format mm- | |  | format. |  |
|  |  |  |  |  |  |  | dd-yyyy | |  |  |  |
| channel | string | |  | FALSE | |  | Valid channel | | | Channel associated with the | |
|  |  |  |  |  |  |  | such as API etc. | | | payment. If not provided it will | |
|  |  |  |  |  |  |  |  |  |  | default to API | |

### **BookTransferSender**

| **Attribute** | **Type** |  | **Required** | |  | **Restrictions** | |  |  | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| accountUID | string |  | FALSE |  |  | Valid account id | |  | Sender account unique identifier. If | |
|  |  |  |  |  |  | exists in the | |  | accountUID is provided then other | |
|  |  |  |  |  |  | system. Max | |  | attributes are not required such as | |
|  |  |  |  |  |  | Length 17 chars | |  | accountNumber, accounttype, currency | |
| accountNumber | string |  | FALSE |  |  | Valid account | |  | Sender account number | |
|  |  |  |  |  |  | number | |  |  |  |
| accountType | string |  | FALSE |  |  | DDA or GL | |  | Account type | |
| currency | string |  | FALSE |  |  | Valid 3-digit | |  | Sender Currency | |
|  |  |  |  |  |  | currency code | |  |  |  |
|  |  |  |  |  |  | such as USD | |  |  |  |

### **BookTransferReceiver**

| **Attribute** | **Type** | **Required** | **Restrictions** | **Notes** |
| --- | --- | --- | --- | --- |
| accountUID | string | FALSE | Valid account id | Receiver account unique identifier. If |
|  |  |  | exists in the | accountUID is provided then other |

|  |  |  | system. Max | attributes are not required such as |
| --- | --- | --- | --- | --- |
|  |  |  | Length 17 chars | accountNumber, accounttype, currency |
| accountNumber | string | FALSE | none | Receiver account number |
|  |  |  |  |  |
| accountType | string | FALSE | DDA or GL | Account type |
| currency | string | FALSE | Valid 3-digit | Sender Currency. |
|  |  |  | currency code |  |
|  |  |  | such as USD |  |

### **Book Transfer Response:**

| **Attribute** | **Type** | **Required** | **Restrictions** | **Notes** |
| --- | --- | --- | --- | --- |
| status | string | FALSE | Success or failure | Status of the API request either it will be a success or a failure |
| code | string | FALSE | Error code such as BOOK001 | Code associated with the error. |
| message | string | FALSE | none | Error message corresponding to the error code indicating the issue in API call and an indication on how to resolve it. |
| data | Book Transfer Response | FALSE | none | none |

### **Book Transfer Response**

| **Attribute** | **Type** | **Required** | **Restrictions** | **Notes** |
| --- | --- | --- | --- | --- |
| paymentUID | string | FALSE | none | Payment unique identifier. |
|  |  |  |  |  |
| externalReferenceId | string | FALSE | none | External reference id provided by the |
|  |  |  |  | origination system |
|  |  |  |  |  |
| paymentStatus | String | FALASE | Valid Payment | Payment status |
|  |  |  | Status |  |

## **Sandbox**

The sandbox environment is an environment that you can play around with, use as a testbed for development. Customers must test their business flows in this environment prior to moving to production environment. Your business bank will onboard you into the sandbox environment and provide you with the API credentials. You must use test customers and test account numbers in the sandbox environment.

**Base URL:**

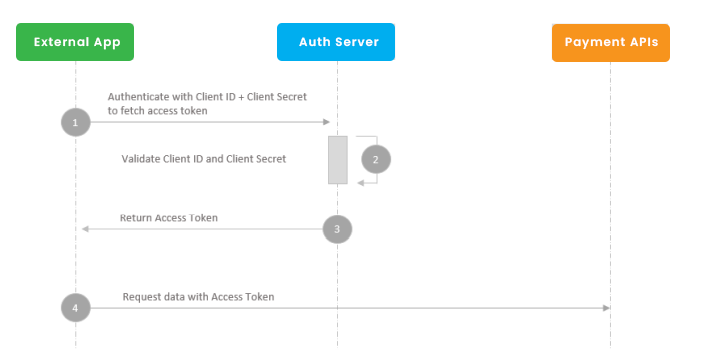
[**https://sandbox-digitalbanking-uat.finzly.io/api/openbanking**](https://sandbox-digitalbanking-uat.finzly.io/api/openbanking)

**Sandbox OAuth Token URL:**

[**https://sandbox-security-uat.finzly.io/auth/realms/BANKOS.UAT.SANDBOX.CUSTOMER/protocol/openid-connect/token**](https://sandbox-security-uat.finzly.io/auth/realms/BANKOS.UAT.SANDBOX.CUSTOMER/protocol/openid-connect/token)

## **Authentication**

Our APIs use oAuth 2.0 authentication protocol to provide the secure access. You will use our API keys with grant type “Client Credentials” to fetch an access token. This token is digitally signed and contains access information such as user role mapping etc. Access token is a short-lived token (30 sec). To authenticate and authorize your API calls, you will have to pass the access token in authorization header with “bearer” prefix.



**Sample Request**

| **curl -X POST [URL]  -H "Content-Type: application/x-www-form-urlencoded"  -H 'Authorization: Basic [Base64(apikey:apisecert)]'  -d 'grant\_type=client\_credentials'** |
| --- |

**Sample Response**

| **{  "access\_token": "",  "expires\_in": 14400,  "refresh\_expires\_in": 1800,  "refresh\_token":"",  "token\_type": "bearer", }** |
| --- |

## **Webhooks**

Webhooks are used to get notified by Finzly bank whenever a payment event happens in your account. Your bank will help you setup a webhook for your accounts. The following events are sent to Webhooks.

**Payment Webhooks**

1. Outgoing credit payment

2. Outgoing debit payment

3. Incoming payments

4. Incoming drawdowns

5. Incoming debit requests

6. Incoming returns

7. Incoming NOCs

8. Incoming RFPs

| **Attribute** | **Type** | **Restrictions** | **Notes** |
| --- | --- | --- | --- |
| paymentId | integer | none | Payment unique identifier assigned by the |
|  |  |  | system to the payment object. |
|  |  |  |  |
| sourceReferenceId | string | none | Unique identifier assigned by th external |
|  |  |  | system to the payment object. |
|  |  |  |  |
| paymentStatus | string | none | Payment status such as VALIDATED, |
|  |  |  | VALIDATION\_FAILURE, TRANSMITTED, |
|  |  |  | PROCESSED, CANCELLED etc |
|  |  |  |  |
| bulkFileName | string | none | Name of the file, if the payment is uploaded |
|  |  |  | from the file to the finzly payment hub system |
|  |  |  | via FTP or API channel. |
| counterParty | string | none | Shortname associated with the legal entity |
|  |  |  | defined in the CRM system. |
|  |  |  |  |
| deliveryDate | string(date) | none | Delivery date refers to the date on which |
|  |  |  | payment is settle |
| deliveryMethod | string | none | The delivery method used to setlle the |
|  |  |  | payment such as ACH, WIRE, RTP, |
|  |  |  | FEDNOW, SWIFT, etc. |
| feeAmount | number(decimal) | none | Fee amount |
|  |  |  |  |
| paymentDate | string(date) | none | Date on which the payment is originated in |
|  |  |  | the system. |
| channel | string | none | Channel represents the system that originates |
|  |  |  | the payment for example API, TELLER, |
|  |  |  | CASHOS, ONLINE etc. |
| paymentIOType | string | none | Whether the payment is Inbound or Outbound |
| paymentNote1 | string | none | Payment notes provided by the external |
|  |  |  | system |

| paymentNote2 | string | none | Additional payment notes |
| --- | --- | --- | --- |
| paymentRemittanceInfo | string | none | Payment remittance information associated |
|  |  |  | with the payment. |
|  |  |  |  |
| paymentType | string | none | The payment type associated with the |
|  |  |  | payment such as Same Day ACH, Regular |
|  |  |  | ACH, WIRE, SWIFT etc. |
| forexContract | string | none | FX contract number associated with the |
|  |  |  | SWIFT payment |
| paymentNocDetails | [Object](https://apidocs.finzly.net/dashboard#schemapayment_paymentnocdetails) | none | The purpose of a NOC to notify you that |
|  |  |  | something about your customer's bank |
|  |  |  | account has changed. |
| paymentNocDetails: nocCode | string | none | Code associated with NOC |
|  |  |  |  |
| paymentNocDetails: | string | none | Original account number |
| originalAccountNumber |  |  |  |
|  |  |  |  |
| paymentNocDetails: | string | none | Corrected account number |
| correctedAccountNumber |  |  |  |
|  |  |  |  |
| paymentNocDetails: | string | none | Original routing number |
| originalRoutingNumber |  |  |  |
|  |  |  |  |
| paymentNocDetails: | string | none | Corrected routing number |
| correctedRoutingNumber |  |  |  |
|  |  |  |  |
| paymentNocDetails: | string | none | Original tran code |
| originalTranCode |  |  |  |
|  |  |  |  |
| paymentNocDetails: | string | none | Corrected tran code |
| correctedTranCode |  |  |  |
|  |  |  |  |
| paymentNocDetails: | string | none | NOC received date and time |
| receivedDateTime |  |  |  |
|  |  |  |  |
| imad | string | none | It is a unique number given to each Fedwire |
|  |  |  | payment. An IMAD/OMAD number can be |
|  |  |  | used to investigate and track wire transfers. |
|  |  |  |  |
| omad | string | none | It is a unique number given to each Fedwire |
|  |  |  | payment. An IMAD/OMAD number can be |
|  |  |  | used to investigate and track wire transfers. |
|  |  |  |  |
| receiverAccountNumber | string | none | Account number associated with the receiver |
|  |  |  |  |
| receiverAccountType | string | none | Receiver account type |
| receiverAddress1 | string | none | Receiver address |
|  |  |  |  |
| receiverAmount | number(decimal) | none | Payment amount received by the recipient |
| receiverBankName | string | none | Recipient's bank name |
|  |  |  |  |
| receiverBankState | string | none | Recipient's bank state |
| receiverDiscretionaryData | string | none | Receiver's discretionary data |
|  |  |  |  |
| receiverEmail | string(email) | none | Receiver's email |
| receiverId | string | none | Receiver unique identifier exist in the Finzly |
|  |  |  | CRM system |
| receiverIdType | string | none | Receiver identifier type |
| receiverName | string | none | Receiver name associated with the payment. |
|  |  |  |  |
| originalReceiverName | string | none | The original receiver name received from the |
|  |  |  | payment network for the inbound payment |
|  |  |  | request. |
| receiverPhoneNumber | string | none | Receiver phone number |
|  |  |  |  |
| receiverRoutingNumber | string | none | Receiver bank routing number |

| receiverState | string | none | Receiver state |
| --- | --- | --- | --- |
|  |  |  |  |
| receiverTemplateName | string | none | If the template name is associated with the |
|  |  |  | receiver in the payment hub system. |
|  |  |  |  |
| receiverType | string | none | Receiver type |
|  |  |  |  |
| receiverZip | string | none | Receiver's zipcode |
| ultimateReceiverName | string | none | Ultimate receiver name is associated with the |
|  |  |  | payment. |
| ultimateReceiverAccountNumber | string | none | The ultimate receiver account number |
| ultimateReceiverAddress1 | string | none | The ultimate receiver address 1 |
|  |  |  |  |
| ultimateReceiverAddress2 | string | none | The ultimate receiver address 2 |
| ultimateReceiverTaxId | string | none | The ultimate receiver tax number |
|  |  |  |  |
| ultimateReceiverCity | string | none | The ultimate receiver city |
| ultimateReceiverState | string | none | The ultimate receiver sate |
|  |  |  |  |
| ultimateReceiverZip | string | none | The ultimate receiver zipcode |
| ultimateReceiverCountry | string | none | The ultimate receiver country |
|  |  |  |  |
| senderAccountNumber | string | none | The sender account number |
| senderAccountType | string | none | The sender account type |
|  |  |  |  |
| senderAddress1 | string | none | The sender address 1 |
| senderAmount | number(decimal) | none | The sender amount |
|  |  |  |  |
| senderBankName | string | none | The sender bank name |
| senderBankState | string | none | The sender bank State |
|  |  |  |  |
| senderDiscretionaryData | string | none | The sender discretionary data |
| senderEmail | string(email) | none | The sender email |
|  |  |  |  |
| senderId | string | none | The sender unique identifier exist in the |
|  |  |  | system |
| senderIdType | string | none | The sender identifier type |
|  |  |  |  |
| senderName | string | none | The sender’s name |
| senderRoutingNumber | string | none | The sender routing number |
|  |  |  |  |
| senderState | string | none | The sender state |
| senderType | string | none | Type of the Sender for e.g., Corporate or |
|  |  |  | Consumer |
| senderZip | string | none | The sender zipcode |
| ultimateSenderName | string | none | The ultimate sender’s name |
|  |  |  |  |
| ultimateSenderAccountNumber | string | none | The ultimate sender account number |
| ultimateSenderAddress1 | string | none | The ultimate sender address 1 |
|  |  |  |  |
| ultimateSenderAddress2 | string | none | The ultimate sender address 2 |
| ultimateSenderTaxId | string | none | The ultimate receiver tax number |
|  |  |  |  |
| ultimateSenderCity | string | none | The ultimate sender city |
| ultimateSenderState | string | none | The ultimate sender sate |
|  |  |  |  |
| ultimateSenderZip | string | none | The ultimate sender zipcode |
| ultimateSenderCountry | string | none | The ultimate sender country |
|  |  |  |  |
| bulkFileNumber | integer | none | Bulk file number |
| entryType | string | none | Whether the payment is Credit or Debit |
|  |  |  |  |
| paymentReturnDetails | Object | This will be | Payment returns details. |
|  |  | populated |  |
|  |  | when the |  |
|  |  | payment is |  |
|  |  | returned. |  |

| paymentReturnDetails: code | string | none | Return code associated with the payment |
| --- | --- | --- | --- |
|  |  |  |  |
| paymentReturnDetails: | string | none | Return code description |
| description |  |  |  |
|  |  |  |  |
| creationDateTime | string(date-time) | none | Date and Time payment is created |
|  |  |  |  |
| lastUpdatedTime | string(date-time) | none | The date and Time of payment got updated |
|  |  |  | last |

## **Frequently Asked Questions**

* **What is Finzly Connect?**

Finzly Connect is the universal payment API for developers to access all the payment rails without worrying about the network and messaging rules.

* **What does Finzly Connect do?**

Finzly Connects offers REST API and webhook notifications to access any Finzly-powered bank. Finzly has normalized the payment data and made the payment processing super simple, so a bank customer, whether a fintech, business or an individual, doesn’t have to fret over the messaging rules. A developer can access ACH, Domestic Wires, International Wires, RTP, and FedNow using Finzly Connect and connect to their favorite Finzly Bank.

* **Does Finzly API support Bulk Payment Processing?**

Finzly system supports the ACH bulk payment processing in a NACHA format via the FTP channel. The bank can create an FTP account to process the ACH bulk file in a Nacha format. The solution provided two types of FTP accounts.

Customer: To be used by the corporate or Fintech to upload the bulk payments

Channel: To be used by the bank’s system to upload the payments associated with multiple customers.

NACHA has various rules around each transaction processing. ACH is very complicated to develop than other payment rails, but Finzly has simplified them with APIs. Finzly also offers file uploads of NACHA formatted files and returns NACHA formatted response files.

* **How does Finzly API offer instant payment access?**

Finzly Connect offers instant payment access and notifications via simple REST API. Develop new financial products such as Point of Sale, QR based invoices, and get paid instantly from your customer’s bank and eliminate card transaction fees

* **How does Fizly Connect secure payment access?**

Our APIs use the oAuth 2.0 authentication protocol to provide secure access. You will use our API keys with grant type “Client Credentials” to fetch an access token. This token is digitally signed and contains access information such as user role mapping. The access token is short-lived (30 sec). To authenticate and authorize your API calls, you must pass the access token in the authorization header with the “bearer” prefix.

* **Who should use Finzly APIs?**

Banks, and Fintechs seeking a modern, flexible, API-centric, end-to-end solution to support their BaaS initiatives and become a developer and fintech-friendly bank.

* **How can I test Finzly APIs?**

Users can perform your testing in Sandbox Mode. This mode provides a testing environment that helps you create an account enabled for making Finzly API calls.

## **API Changelog**

All notable changes to the Finzly APIs will be displayed on this page.