Logo

Description automatically generated

**Primary API Onboarding**

***Index***

***Appendix***, 6

**Exchange Secret**, 2

**Fetch Stripe Token**, 3

**High Level interaction**, 1

**KYC Integration**, 6

**Stripe Integration**, 8

**Submit Investment Request**, 5

**Submit Investment Request – Field Description**, 12

***Introduction***

This document outlines the steps required to integrate and begin utilizing the Rialto Primary Invest Now platform. It serves as a quick setup guide, enabling you to efficiently build your integration and leverage the full capabilities of the Rialto Primary Market.

**High Level interaction**

From a high-level perspective, integrating a Primary Market security is a straightforward, single-step process. This involves submitting all required fields and ensuring compliance with the regulatory requirements.  
As you may already know, an API functions as a one-way service communication designed for a specific purpose.

In this context a primary investment involves the following interactions

* Exchange Secret: Authentication (*mandatory*).
* Fetch the security information (optional).
* Fetch the IDology (KYC) credentials (optional).
* Fetch the Stripe credentials (optional).
* Initiate the KYC process (mandatory)
* Submit the investment Request (*mandatory*).

As you can see there are only three mandatory activities, getting an API access token for security purposes and submitting the investment request.

However, the previous checklist just outlines the technical perspective. Onboarding a new security in the Rialto platform involves several regulatory steps, which are beyond the scope of this document. These steps include:

* Providing payment methods.
* Providing stock price and minimum investment.
* Providing a full list of payment methods.
* Providing a subscription agreement with the all the fields to be populated.
* Providing a wire check account or payment gateway integration (Stripe).
* Confirming with Rialto compliance team any other regulatory requirement.

For the purposes of this document, it is assumed that all prior steps have been completed and that the Primary Market security has been successfully onboarded onto the Rialto platform.

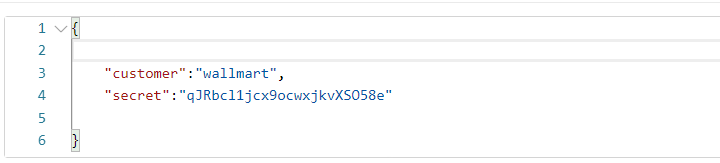
**Exchange Secret**

This is a necessary step to use the Rialto API. Your customer secret has to be exchanged for an API access token.

Make an HTTP call (POST) to this API:

***<env\_url>/secondarytrading/api/onboarding/authorize***

With the following payload:

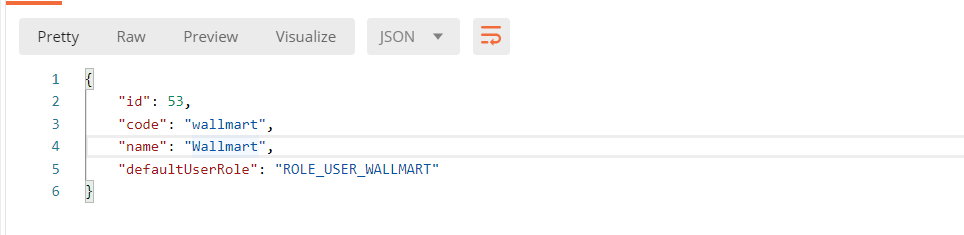


Where:

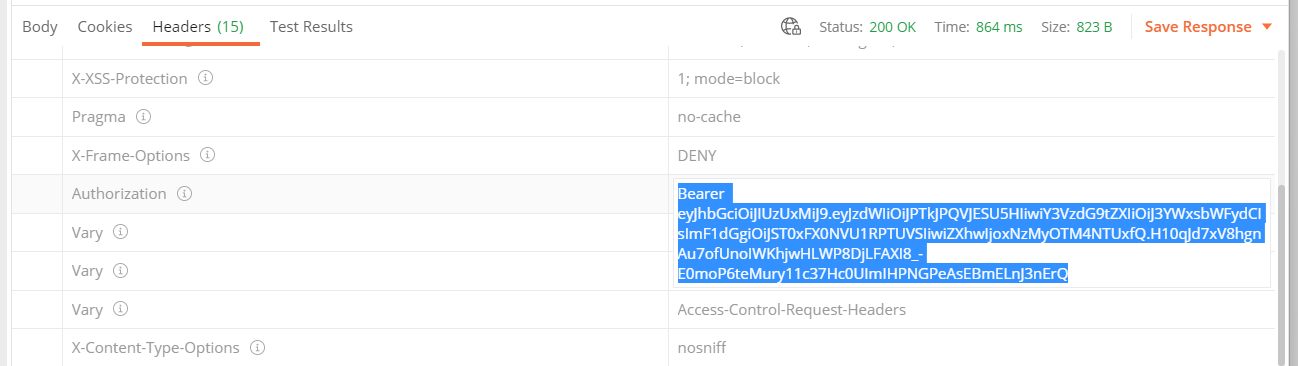
+ ***customer***: The Customer code provided by Rialto

+ ***secret***: The customer secret provided by Rialto

If the process is successful, a response similar to the following will be received:



The required access token can be found in the response headers, under the “Authorization” key.



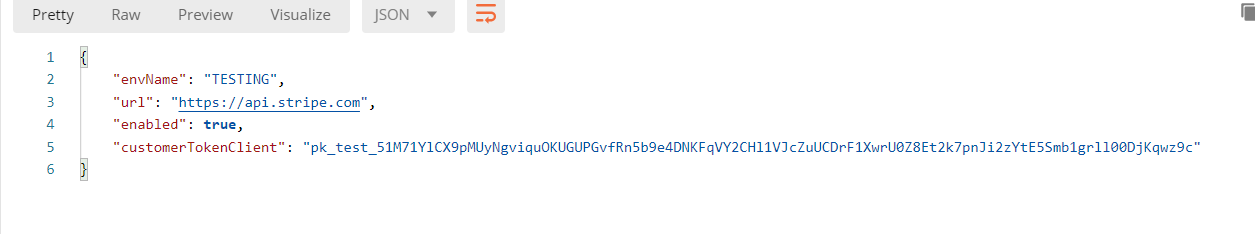
**Fetch Stripe Token**

Invoke the service to get access to the Rialto Stripe token that you can use to setup different ACH and Credit Card Stripe integrations.

Make an HTTP call (GET) to this API:

***<env\_url>*/api/external-settings/stripe**

With no payload and using the token acquired from the ***Exchange Token*** service as a Bearer token in the “Authorization” header, a response like the following JSON will be received.

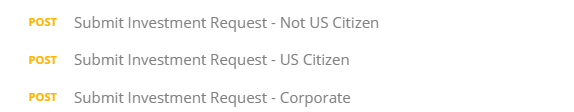


The ***customerTokenClient*** value in the response is used to set up payment integrations.

**Submit Investment Request**

There are essentially three ways to invoke the Investment Request service:

* US Citizen: See example in the Postman Collection as ***Submit Investment Request - US Citizen***
* Non-US Citizen: See example in the Postman Collection as ***Submit Investment Request - Not US Citizen***
* Corporate: See example in the Postman Collection as ***Submit Investment Request – Corporate***

****

All investment requests must be made through the Primary Security API.

Make an HTTP call (POST) to this API:

***<env url>/api/primary-security/on-primary-market-investment-request***

For example

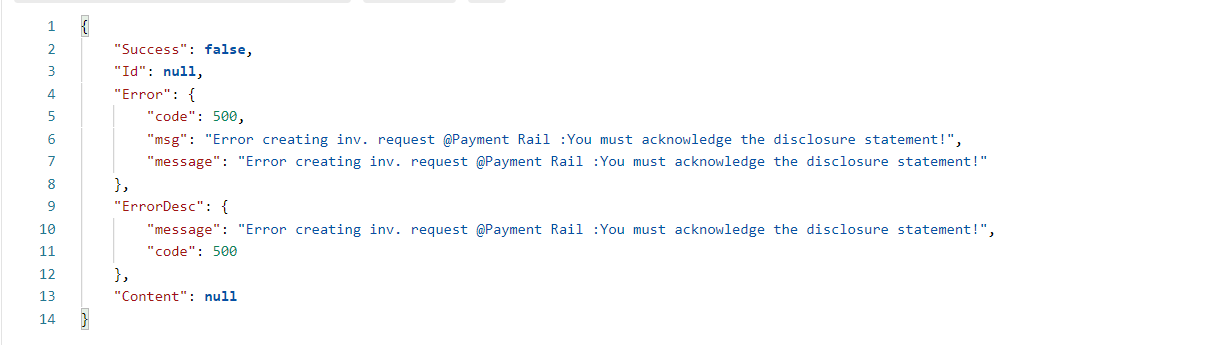
<https://rialto-primary-trading-service-dev01.services.rialtomarkets.io/api/primary-security/on-primary-market-investment-request>

You will need to use the same Bearer token acquired through the ***Exchange Secret*** service. If the process is successful, a success message like the following will be received.

****

Where that Id number is the investment request ID that uniquely identifies the investment request.

If an error occurs, the response will include an "Error" attribute with the error description, as shown below.



***Appendix***

**KYC Integration**

This is a high-level description of the steps required to integrate the KYC (Know Your Customer) verification process into your UI. The main goal of this process is to verify the investor's PII (Personally Identifiable Information) through our KYC vendor and obtain the ***IdologyIdentificationReference*** data, which will be submitted in the **Submit Investment Request service.**

1. *Initiate KYC Verification*

To begin the KYC verification process, use the **/api/kyc/initiate** endpoint. Submit the investor’s PII (Personally Identifiable Information) in the request body. Below is an example of the request payload:

POST /api/kyc/initiate

{

    "firstName": "Joseph",

    "lastName": "Gilbert",

    "address": "123 Main Street",

    "city": "Boston",

    "state": "MA",

    "stateName": "Massachusetts",

    "country": "USA",

    "zip": "02116",

    "ssn": "666000001",

    "passport": null,

    "email": "fake.email@gmail.com",

    "telephone": "666000001",

    "dob": "1943-04-08"

}

1. Handling KYC Outcomes

There are three possible outcomes when initiating KYC verification:

* HARD Fail: If the investor’s PII cannot be verified due to a business rule or if the investor is on a watchlist, the process will return an error response indicating a "HARD fail".

{

    "kycId": "5629758036",

    "key": "id.failure",

    "success": false,

    "message": "Something went wrong with your application. Please try again later.",

    "idScanUrl": ""

}

* Successful Verification: If the investor's PII is correctly verified, the process will return a successful response.

{

    "kycId": "5629762702",

    "key": "id.success",

    "success": true,

    "message": "PASS",

    "idScanUrl": ""

}

* SOFT Fail: If the investor's PII is mostly correct but contains minor inconsistencies (e.g., a single-digit error in the SSN), the response will indicate a "SOFT fail". In this case, the investor will need to complete further identity validation by uploading their ID or driver’s license and possibly taking a selfie. The response will include a link (idScanUrl) for the investor to complete this step.

{

    "kycId": "5629765748",

    "key": "id.failure",

    "success": false,

    "message": "FAIL",

    "idScanUrl": "https://web.idologylive.com/idscan/v2.jsp?id=nNyklf9H272162b722"

}

1. Finalizing the KYC Process (Optional)

For investors who received a "SOFT fail" and completed the additional identity verification through the provided link (idScanUrl), you can check the final status of the KYC process using the **/api/kyc/finalize** endpoint. This is an ***optional*** step.

POST /api/kyc/finalize

{

    "kycId": "5629765748"

}

If the investor has not yet completed the identity verification, the response will indicate that the process is still pending.

{

    "kycId": "5629765748",

    "key": "id.scan.pending",

    "success": false,

    "message": "Please finish your identity verification before continuing."

}

If the process is completed, the response will provide the final result of the KYC verification.

{

    "kycId": "5629765748",

    "key": "result.id.scan.approved",

    "success": true,

    "message": "Approved"

}

1. Submitting the Investment Request

Once the KYC process is complete, include the IdologyIdentificationReference data in the Submit Investment Request service. If an idScanUrl was provided during the KYC process, include it as well.

"IdologyIdentificationReference": {

   "IdNumber": "124",

   "IdScanLink": "https://web.idologylive.com/idscan/v2.jsp?id=nNyklf9H272162b722"

}

**Stripe Integration**

This is a high-level description of the steps necessary to integrate *Credit Card* and *ACH* payments through *Stripe Gateway* into your UI. It serves as a general example of the three-step process required to place the Stripe popups in your screen. For a detailed and rigorous guide, please refer to the official Stripe documentation, as processes may change, or additional details may be required.

The primary goal is to set up the *StripeCreditCard* or *StripeACH* or StripeWallet **SetupIntent**, which will be submitted in the **Submit Investment Request service.**

1. First, create a Setup Intent. Instructions for creating a Setup Intent can be found at the following link: <https://docs.stripe.com/api/setup_intents/create>

In order to do this you will have to send your ***Stripe API key*** , like the following ***sk\_test\_4eC39HqLyjWDarjtT1zdp7dc***

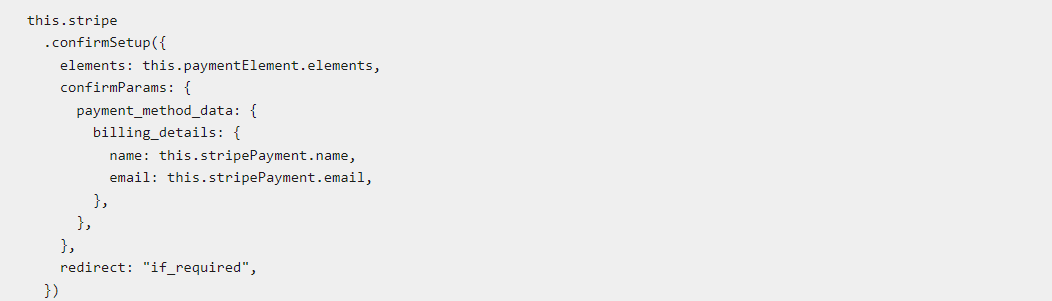
Then , the setup intent in the response will look as follows

seti\_1Mm8s8LkdIwHu7ix0OXBfTRG\_secret\_NXDICkPqPeiBTAFqWmkbff09lRmSVXe

1. Then you can use the previous setup intent to build the ngx Stripe component related to the chosen payment method selected. You can fetch more information about this on the following link: <https://ngx-stripe.dev/docs/payment-element>.
2. After triggering the form and submitting the payment method an event ***confirmSetup*** will be triggered. To execute the ***confirmSetup*** event we must build an object ***StripeFactoryService*** which is part of the Stripe library.

Once built this ***StripeFactoryService*** we trigger the final submit calling the service ***confirmSetup*** of the Stripe library, where we pass some values acquired in step 2.

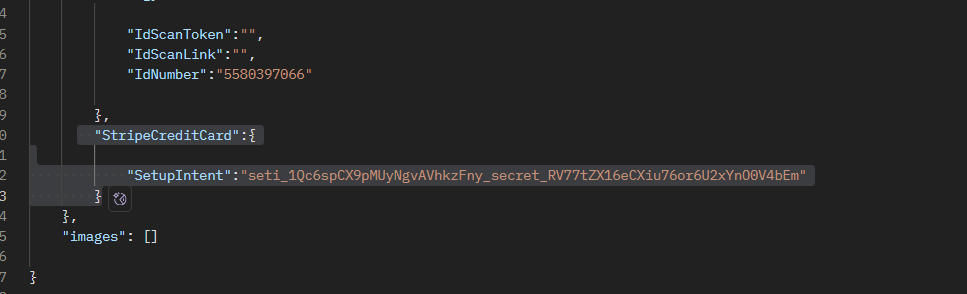
https://docs.stripe.com/js/setup\_intents/confirm\_setup



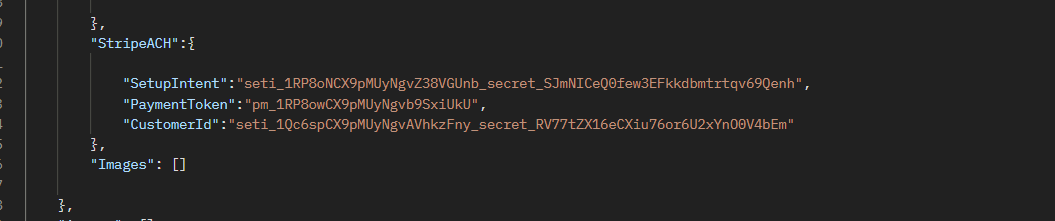
In the response of this service, we will get *C****lientSecret***and***StripeTokenClient.***

1. Finally, when invoking the **Submit Investment Request** you can provide the setup intent acquired in step 1. If provided (under the “*primaryInvestor”* node), it will be used for payment integration.

For example, for a Credit card:



And for an ACH payment :



Then Rialto settlement process will use these value to trigger the payment.

**Status Report**

***For regulatory purposes, this service should be enabled on explicit demand***

There are essentially two ways to invoke the Investment Request service:

* Status report: See example in the Postman Collection as ***Get Inv. Requests - Status Report***
* Status detail report: See example in the Postman Collection as ***Get Inv. Requests - Status Detail Report***

Texto

Descripción generada automáticamente

All investment requests must be made through the Primary Security API.

For the ***Status Report,*** make an HTTP call (POST) to this API:

***<env url>/api/primary-security/on-primary-market-investment-request-statuses-report***

For example

<https://rialto-primary-trading-service-dev01.services.rialtomarkets.io/api/primary-security/on-primary-market-investment-request>

You will need to use the same Bearer token acquired through the ***Exchange Secret*** service. If the process is successful, a success message like the following will be received.

***With the following payload:***

{

    "symbol": "ATL",

    "from": 1640995200,

    "to": 1646092799

}

If there is a security for the given symbol and there are investments requests for it, they can be found in the response. Like the following example.

Texto

Descripción generada automáticamente

For the ***Status Detail Report,*** make an HTTP call (POST) to this API:

***<env url>/api/primary-security/on-primary-market-investment-request-status-detail-report***

For example

<https://rialto-primary-trading-service-dev01.services.rialtomarkets.io/api/primary-security/on-primary-market-investment-request-status-detail-report>

You will need to use the same Bearer token acquired through the ***Exchange Secret*** service. If the process is successful, a success message like the following will be received.

***With the following payload:***

{

    "invRequestId": 47

}

If there is an investments request for the given id, its details can be found in the response. Like the following example.

Texto

Descripción generada automáticamente

**Submit Investment Request – Field Description**

This section summarizes each screen of the Invest Now process, including required fields, expected data types, and enumerated values where applicable.

## Investor and Investment Information - Submit fields

All fields marked with an asterisk on the UI are mandatory.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Mandatory | Data Type | Comments |
| FirstName | Yes | String |  |
| MiddleName | No | String |  |
| LastName | Yes | String |  |
| Email | Yes | Email |  |
| UsState | Yes | String | Applies for US Citizens and NonUS. Place foreign state province here |
| Country | Yes | String |  |
| DOB | Yes | Date | Format: yyyy-MM-dd |
| IdNumber | Yes | String | SSN or national ID |
| MailingAddress | Yes | String |  |
| Apartment | No | String |  |
| City | Yes | String |  |
| ZipCode | Yes | String |  |
| PersonalNumber | Yes | String |  |
| DocType | Yes | Enum | Alternative document provided.  Must be Passport if non US citizen. See below |
| DocNumber | Yes | String |  |
| DocExpiry | Yes | Date | Format: yyyy-MM-dd |
| DocIssuingDate | Yes | Date | Format: yyyy-MM-dd |
| DocIssuingCountry | Yes | String | 3 letter : ISO 3166-1 alpha-2 |
| DocIssuingState | Yes | String | 2 letter: USPS State Abbreviation Codes for US states. Plain text for foreign provinces/states |
| AccreditedInvestor | Yes | Boolean |  |
| AccreditedInvestorType | Conditional | Enum | Required if AccreditedInvestor = true |
| NetWorth | Conditional | Decimal | Required if not AccreditedInvestor |
| YearIncome | Conditional | Decimal | Required if not AccreditedInvestor |
| Citizenship | Yes | Enum | US or NotUS |
| Employer | Yes | String |  |
| JobTitle | Yes | String |  |
| EmployerIsBrokerDealer | Yes | Boolean |  |
| notInvestedRegCF12Month | Yes | Boolean |  |
| acknowledgeDisclosureStatement | Yes | Boolean |  |
| acknowledgeEducationalMaterial | Yes | Boolean |  |
| acknowledgeOnlyElectronicCommunicationRialto | Yes | Boolean |  |
| acknowledgeRightCancel48hs | Yes | Boolean |  |
| acknowledgeFormCOffering | Yes | Boolean |  |
| BrokerDealerAgreementCheckbox | Yes | Boolean |  |
| SubscriptionAgreementCheckbox | Yes | Boolean |  |
| Signature | Yes | String |  |
| HearAboutUs | Yes | String |  |
| InvestingAs | Yes | Enum |  |
| AdvisorHelp | No | Boolean |  |
| AdvisorName | No | String |  |
| AdvisorAddress | No | String |  |
| Currency | Yes | String | 3-letter currency |
| StripeCreditCard.SetupIntent | Yes if PaymentMethod = CREDIT\_CARD\_STRIPE | String | Stripe SetupIntent secret |
| StripeACH.SetupIntent | Yes if PaymentMethod = ACH\_STRIPE | String | Stripe SetupIntent secret |
| StripeACH.PaymentToken | Yes if PaymentMethod = ACH\_STRIPE | String | Stripe Payment Token ID |
| StripeACH.CustomerId | Yes if PaymentMethod = ACH\_STRIPE | String | Stripe Customer ID |
| IdologyIdentificationReference.IdScanToken | Yes | String | Returned by Idology |
| IdologyIdentificationReference.IdScanLink | Yes | String | Returned by Idology |
| IdologyIdentificationReference.IdNumber | Yes | String | Returned by Idology |
|  |  |  |  |

## Enumerated Field Values

### InvestingAs

- a\_Company  
- an\_Individual  
- a\_Self\_Directed\_IRA

### PaymentMethod

- ALTO\_IRA  
- WIRE\_CHECK  
- CREDIT\_CARD\_STRIPE  
- WALLET\_STRIPE  
- SELF\_DIRECTED\_IRA  
- ACH\_STRIPE  
- CAMAPLAN\_IRA

### AccreditedInvestorType

- net\_worth\_spouse\_1\_million  
- net\_worth\_individual\_200k  
- series\_7\_65\_82\_license

### Citizenship

- US  
- NotUS

### DocType

- Passport  
- Driver License