PIX2STABLE - OnChain Transfer

You have received your credentials, and now we'll walk through a complete use case that expands upon the basic PIX to STABLE and then after on-chain conversions and transfers flow:

Use Case: PIX to BRLA → USDC/USDT → Subaccounts → External Wallets

This flow starts with a deposit in local currency via PIX to the BRLA stablecoin. From there, BRLA is exchanged for another stable asset such as USDC or USDT.

Once the stable assets are available, subaccounts can be created to organize and segregate balances. Funds are then transferred from the main account to the respective subaccounts as needed.

Finally, tokens held in subaccounts can be transferred to external wallets.

What Will We Cover Here?

- Login
- KYC
- Subaccount creation
  - Subaccount KYC
- About Webhooks
- Deposit via PIX to BRLA (Funds credited to Main Account in BRLA Stablecoin)
  - Convert BRLA to USDC or USDT
- Transfer Tokens to Subaccounts
- Send Tokens from Subaccounts to External Wallets

Login

Once you receive your credentials, use the following endpoint to authenticate and receive your JWT.

INFO

We highly recommend `creating API Keys for security and convenience`.

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/auth/login
```

Sample JSON Body

```
{
  "email": "your.email@provider.com",
  "password": "UseAStrongPassword123!"
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/auth/login" \\
 -H "Content-Type: application/json" \\
 -d '{
     "email": "your.email@provider.com",
     "password": "UseAStrongPassword123!"
 }'
```

Upon a successful login **(HTTP 200)**, you will receive an email with a token to validate your login.

Validate Login

Using the token (emailToken) sent to your email, call the following endpoint to receive your authentication codes (JWT):

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/auth/validate-login
```

Sample JSON Body

```
{
  "email": "your.email@provider.com",
  "emailToken": "777777"
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/auth/validate-
login" \\
 -H "Content-Type: application/json" \\
 -d '{
     "email": "your.email@provider.com",
     "emailToken": "777777"
 }'
```

JSON Response

```
{
  "accessToken": "eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "refreshToken": "eyJhbdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
}
```

KYC Main Account

KYC (Know Your Customer) is essential to track who is moving or receiving funds.

INFO

We'll soon see that, to perform KYC on a subaccount, it's enough to include the subAccountId field as a parameter in the request.

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/kyc/level-1/api
```

Field	Type	Required	Description
fullName	string	Yes	The complete name of the individual.
dateOfBirth	string	Yes	The date of birth of the individual (Format: YYYY-MM-DD).
countryOfDocument	string	Yes	The country issuing the document.
documentType	string	Yes	The type of document (must be one of ID, Passport, Driver's License).
documentNumber	string	Yes	The number of the document.
countryOfTaxId	string	Yes	The country where the tax identification number was issued.
taxIdNumber	string	Yes	The tax identification number of the individual.
email	string	No	The email address of the individual.
phone	string	No	The phone number of the individual.
country	string	Yes	The country of residence.
state	string	Yes	The state/province of residence.
city	string	Yes	The city of residence.
zipCode	string	Yes	The postal code of the residence.
streetAddress	string	Yes	The street address of the residence.

INFO

All countries and states follow the ISO Alpha-3 standard (Example: USA-CA)

Sample JSON Body

```
{
  "fullName": "Jane Doe",
  "dateOfBirth": "1999-08-16",
  "countryOfDocument": "BRA",
  "documentType": "Passport",
  "documentNumber": "UJ252482",
  "countryOfTaxId": "BRA",
  "taxIdNumber": "75764220173",
  "country": "BRA",
  "state": "SP",
  "city": "SP",
  "zipCode": "12243010",
  "streetAddress": "Rua Madre Paula"
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/kyc/level-1/api"
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
 -H "Content-Type: application/json"
 -d '{
     "fullName": "Jane Doe",
     "dateOfBirth": "1999-08-16",
     "countryOfDocument": "BRA",
     "documentType": "Passport",
     "documentNumber": "UJ252482",
     "countryOfTaxId": "BRA",
     "taxIdNumber": "75764220173",
     "country": "BRA",
     "state": "SP",
     "city": "SP",
     "zipCode": "12243010",
     "streetAddress": "Rua Madre Paula"
 }
```

JSON Response

```
{
  "id": "1ee11163-9tjb-4389-9f84-074ccff7085d"
}
```

KYC - Track KYC Validation

Before proceeding, ensure that the KYC for the main account has been successfully validated. Use the following GET endpoint:

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/kyc/attempts/**YOUR-KYC-ID-
HERE**
```

cUrl Example

```
curl -X GET "https://api.sandbox.avenia.io:10952/v2/kyc/attempts/{kyc-
attempt-id}" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response - COMPLETED example

```
{
  "attempt": {
    "id": "e51359cd-04b0-4bbc-bf7d-0ee515112d74",
    "levelName": "level-1",
    "submissionData": null,
    "status": "COMPLETED",
    "result": "APPROVED",
    "resultMessage": "",
    "retryable": false,
    "createdAt": "2025-03-25T07:39:40.54713Z",
    "updatedAt": "2025-03-25T07:39:40.54713Z"
  }
}
```

JSON Response - PENDING example

```
{
  "attempt": {
    "id": "b83802a1-afe9-46ac-96d6-ade6c5961bd3",
    "levelName": "level-1",
    "submissionData": null,
    "status": "PENDING",
    "result": "",
    "resultMessage": "",
    "retryable": false,
    "createdAt": "2025-03-26T22:50:14.201695Z",
    "updatedAt": "2025-03-26T22:50:14.201695Z"
  }
}
```

JSON Response - REJECT example

```
{
  "attempt": {
    "id": "5bafd6cd-ec40-4dd3-83e0-a5af117c304a",
    "levelName": "level-1",
    "submissionData": null,
    "status": "COMPLETED",
    "result": "REJECTED",
    "resultMessage": "name does not match",
    "retryable": false,
    "createdAt": "2025-03-26T22:50:14.201695Z",
    "updatedAt": "2025-03-26T22:50:14.201695Z"
  }
}
```

Create Subaccount

To register your client, create a subaccount. Subaccounts allow the Main Account (you) to manage your clients.

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/account/sub-accounts
```

Field	Type	Description
accountType	string	Specify "INDIVIDUAL" for a personal subaccount.

Field	Type	Description
name	string	A name for the subaccount (e.g., "Jane Doe").

Sample JSON Body

```
{
  "accountType": "INDIVIDUAL",
  "name": "Jane Doe"
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/account/sub-
accounts" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" \
 -H "Content-Type: application/json"
 -d '{
     "accountType": "INDIVIDUAL",
     "name": "Jane Doe"
 }'
```

JSON Response

The response will include the ID of the created subaccount.

```
{
  "id": "1ee0a663-922b-4389-9f84-074ccff7085d"
}
```

KYC for Subaccount

KYC (Know Your Customer) is essential to track who is moving or receiving funds. For subaccounts, perform KYC using the API endpoint below. Be sure to pass the `subAccountId` parameter.

Since this operation is performed for a **subaccount**, you must include the field as an endpoint parameter **subAccountId**.

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/kyc/level-1/api?
subAccountId=1ee0a663-922b-4389-9f84-074ccff7085d
```

Field	Type	Required	Description
fullName	string	Yes	The complete name of the individual.
dateOfBirth	string	Yes	The date of birth of the individual (Format: YYYY-MM-DD).
countryOfDocument	string	Yes	The country issuing the document.
documentType	string	Yes	The type of document (must be one of ID, Passport, Driver's License).
documentNumber	string	Yes	The number of the document.
countryOfTaxId	string	Yes	The country where the tax identification number was issued.
taxIdNumber	string	Yes	The tax identification number of the individual.
email	string	No	The email address of the individual.
phone	string	No	The phone number of the individual.
country	string	Yes	The country of residence.
state	string	Yes	The state/province of residence.
city	string	Yes	The city of residence.

Field	Type	Required	Description
zipCode	string	Yes	The postal code of the residence.
streetAddress	string	Yes	The street address of the residence.

INFO

All countries and states follow the ISO Alpha-3 standard (Example: USA-CA)

Sample JSON Body

```
{
  "fullName": "Jane Doe",
  "dateOfBirth": "1999-08-16",
  "countryOfDocument": "BRA",
  "documentType": "Passport",
  "documentNumber": "UJ252482",
  "countryOfTaxId": "BRA",
  "taxIdNumber": "75764220173",
  "country": "BRA",
  "state": "SP",
  "city": "SP",
  "zipCode": "12243010",
  "streetAddress": "Rua Madre Paula"
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/kyc/level-1/api?
subAccountId=1ee0a663-922b-4389-9f84-074ccff7085d"
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
 -H "Content-Type: application/json"
 -d '{
     "fullName": "Jane Doe",
     "dateOfBirth": "1999-08-16",
     "countryOfDocument": "BRA",
     "documentType": "Passport",
     "documentNumber": "UJ252482",
     "countryOfTaxId": "BRA",
     "taxIdNumber": "75764220173",
     "country": "BRA",
     "state": "SP",
     "city": "SP",
     "zipCode": "12243010",
     "streetAddress": "Rua Madre Paula"
 }
```

JSON Response

```
{
  "id": "1ee11163-9tjb-4389-9f84-074ccff7085d"
}
```

Track KYC Subaccount Validation

Before proceeding, ensure that the KYC for the subaccount has been successfully validated. Use the following GET endpoint:

Since this operation is performed for a **subaccount**, you must include the field as an endpoint parameter **subAccountId**.

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/kyc/attempts/**YOUR-KYC-ID-HERE**?subAccountId=1ee0a663-922b-4389-9f84-074ccff7085d
```

cUrl Example

```
curl -X GET "https://api.sandbox.avenia.io:10952/v2/kyc/attempts/{kyc-attempt-id}?subAccountId=1ee0a663-922b-4389-9f84-074ccff7085d" \
 -H "Authorization: Bearer eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response - COMPLETED example

```json
{
  "attempt": {
    "id": "e51359cd-04b0-4bbc-bf7d-0ee515112d74",
    "levelName": "level-1",
    "submissionData": null,
    "status": "COMPLETED",
    "result": "APPROVED",
    "resultMessage": "",
    "retryable": false,
    "createdAt": "2025-03-25T07:39:40.54713Z",
    "updatedAt": "2025-03-25T07:39:40.54713Z"
  }
}
```

JSON Response - PENDING example

```json
{
  "attempt": {
    "id": "b83802a1-afe9-46ac-96d6-ade6c5961bd3",
    "levelName": "level-1",
    "submissionData": null,
    "status": "PENDING",
    "result": "",
    "resultMessage": "",
    "retryable": false,
    "createdAt": "2025-03-26T22:50:14.201695Z",
    "updatedAt": "2025-03-26T22:50:14.201695Z"
  }
}
```

JSON Response - REJECT example

```json
{
  "attempt": {
    "id": "5bafd6cd-ec40-4dd3-83e0-a5af117c304a",
    "levelName": "level-1",
    "submissionData": null,
    "status": "COMPLETED",
    "result": "REJECTED",
    "resultMessage": "name does not match",
    "retryable": false,
    "createdAt": "2025-03-26T22:50:14.201695Z",
    "updatedAt": "2025-03-26T22:50:14.201695Z"
  }
}
```

About Webhooks

Before we begin the Operations section, we'll share a guide for integrating webhooks—this will allow you to receive real-time updates for every stage of a Ticket event. A Ticket event refers to any update related to a created ticket (i.e., a money movement) within your Account; it may also involve a SubAccount. Events can have multiple statuses, such as TICKET-CREATED, DEPOSIT-PROCESSING, and so on.

We recommend following these steps in order:

- Webhook Management
- Verifying Webhook Authenticity
- Webhooks Events

Pix to BRLA

We will now transfer balances to the main account via PIX, and BRLA Stable will be deposited.

WARNING

The quote is only valid for 15 seconds!

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate?
inputCurrency=BRL&inputPaymentMethod=PIX&outputAmount=100&outputCurrenc
y=BRLA&outputPaymentMethod=INTERNAL&inputThirdParty=false&outputThirdPa
rty=false
```

Field	Value	Description	Required
inputCurrency	BRL	The currency in which the mainAccount (in this case, Brazilian	yes

Field	Value	Description	Required
		Reais).	
inputPaymentMethod	PIX	The payment method in this case PIX method.	yes
outputAmount	100	The amount that the mainAccount receive (e.g., 98.58 = R$98.58).	yes
outputCurrency	BRLA	The currency in which the mainAccount will receive funds. (in this case BRLA stable)	yes
outputPaymentMethod	INTERNAL	The method by which the mainAccount will receive funds (as an internal balance).	yes
inputThirdParty	false	Since the information here is being deposit by de owner of accounts and subaccounts created by Avenia API, it will not be necessary to include this.	yes
outputThirdParty	false	Since the information here is being outputed in accounts and subaccounts created by Avenia API, it will not be necessary to include this.	yes

cUrl Example

```
curl -X GET
"https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate?
inputCurrency=BRL&inputPaymentMethod=PIX&outputAmount=100&outputCurrenc
y=BRLA&outputPaymentMethod=INTERNAL&inputThirdParty=false&outputThirdPa
rty=false" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "inputCurrency": "BRL",
  "inputPaymentMethod": "PIX",
  "inputAmount": "100.2",
  "outputCurrency": "BRLA",
  "outputPaymentMethod": "INTERNAL",
  "outputAmount": "100",
  "markupAmount": "0",
  "markupCurrency": "",
  "inputThirdParty": false,
  "outputThirdParty": false,
  "appliedFees": [
    {
      "type": "Markup Fee",
      "description": "Total markup fees represented in the input
currency.",
      "amount": "0",
      "currency": "BRL"
    },
    {
      "type": "In Fee",
      "description": "Fees due to input currency and input payment
method.",
      "amount": "0.2",
      "currency": "BRL"
    },
    {
      "type": "Conversion Fee",
      "description": "Fees due to conversion from input currency to
output currency.",
      "amount": "0",
      "currency": "BRL"
    },
    {
      "type": "Out Fee",
      "description": "Fees due to output currency and output payment
method.",
      "amount": "0",
      "currency": "BRL"
    },
    {
      "type": "Gas Fee",
      "description": "Fees due to blockchain transaction costs.",
      "amount": "0",
      "currency": "BRL"
    }
  ],
  "basePrice": "1",
  "pairName": "BRLBRLA"
}
```

Ticket - Closing the Order

With the quote in hand, we will initiate the order closing process—referred to as "Ticket"—where the quote is finalized and the requested operation is executed, which in our case is a transfer to our main account.

Field	Value	Descrip
quoteToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....	The quoteTo obtainec from the quote endpoin
ticketBlockchainOutput	beneficiaryWalletId	Contains details regardin the ticke blockcha output.
└ beneficiaryWalletId	00000000-0000-0000-0000-000000000000	Here we the ID of your ma account a zero id which makes Avenia A link this to your main

Field	Value	Descrip
		account who mac the requ

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/account/tickets
```

Sample JSON Body

Remember that here we pass a zeroed uuid because we are linking your main account to this deposit (who will receive the funds)

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "ticketBlockchainOutput": {
    "beneficiaryWalletId": "00000000-0000-0000-0000-000000000000"
  }
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/account/tickets" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" \
 -H "Content-Type: application/json" \
 -d '{
     "quoteToken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
     "ticketBlockchainOutput": {
     "beneficiaryWalletId": "00000000-0000-0000-0000-000000000000"
   }
  }
```

JSON Response

Here, we can see that the brCode is received, which represents the PIX payment code. The ticket will remain in a pending state until the PIX payment is completed.

> **INFO**
>
> If you've already registered a webhook for the `TICKET` event, you will start receiving webhook notifications from this point onward.

```
{
  "brCode": "00020126810014br.gov.bcb.pix01365c2c61a1-134b-4c34-958f-ea3122ac717f0219Avenia Ticket Payment5204000053039865406975.285802BR5917Avenia API Ltda6009Sao Paulo622905253Uu0qFigaAnwGdmmDJp3R9Yuz6304DE64",
  "expiration": "2025-04-14T13:58:47.609482542Z",
  "id": "b73767f7-1343-4176-9298-fffc85ea71a4"
}
```

> **INFO**
>
> In the sandbox environment, the PIX payment is simulated within a few seconds.

Verify status from Ticket

> **INFO**
>
> An alternative to continuously checking the ticket history to track its current stage is to use webhooks instead.

Next, verify if the ticket status is PAID by checking the tickets by id endpoint.

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/account/tickets/YOUR-TICKET-UUID-HERE
```

cUrl Example

```
curl -X GET
"https://api.sandbox.avenia.io:10952/v2/account/tickets/b73767f7-1343-
4176-9298-fffc85ea71a4" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response

INFO

Note: The status can be in: UNPAID PROCESSING PAID FAILED PARTIAL-FAILED

```
{
  "ticket": {
    "id": "b73767f7-1343-4176-9298-fffc85ea71a4",
    "workspaceId": "2ac803ad-faf7-489f-9c1a-c6a64072e699",
    "userId": "05505dde-c2e4-47c5-bd5c-071b4c4bb6a4",
    "status": "PAID",
    "reason": "",
    "failureReason": "",
    "createdAt": "2025-04-22T14:24:02.038578Z",
    "updatedAt": "2025-04-22T14:24:24.389526Z",
    "expiresAt": "2025-04-23T14:24:02.036433Z",
    "quote": {
      "id": "763fb47b-6bc4-4abc-acb3-800139ca9772",
      "ticketId": "0b695784-81de-4ba4-888a-947a5efe4f55",
      "inputCurrency": "BRL",
      "inputPaymentMethod": "PIX",
      "inputAmount": "100.2",
      "outputCurrency": "BRLA",
      "outputPaymentMethod": "INTERNAL",
      "outputAmount": "100",
      "markupCurrency": "",
      "markupAmount": "0",
      "sendMethod": "",
      "inputThirdParty": false,
      "outputThirdParty": false,
      "basePrice": "1",
      "appliedFees": [
        {
          "type": "Markup Fee",
          "amount": "0",
          "currency": "BRL",
          "rebatable": false,
          "description": "Total markup fees represented in the input
currency."
        },
        {
          "type": "In Fee",
          "amount": "0.2",
          "currency": "BRL",
          "rebatable": true,
          "description": "Fees due to input currency and input payment
method."
        },
        {
          "type": "Conversion Fee",
          "amount": "0",
          "currency": "BRL",
          "rebatable": true,
          "description": "Fees due to conversion from input currency to
output currency."
        },
        {
          "type": "Out Fee",
          "amount": "0",
          "currency": "BRL",
          "rebatable": true,
          "description": "Fees due to output currency and output
payment method."
        },
        {
          "type": "Gas Fee",
          "amount": "0",
          "currency": "BRL",
          "rebatable": false,
          "description": "Fees due to blockchain transaction costs."
        }
      ],
      "pairName": "BRLBRLA",
      "outputBrCode": "",
      "createdAt": "2025-04-22T14:24:01Z"
    },
    "rebate": {
      "id": "ce260673-1590-4a0f-84c5-e2bf3bd97cf3",
      "ticketId": "0b695784-81de-4ba4-888a-947a5efe4f55",
      "amount": "0.1",
      "currency": "BRLA",
      "destinationWalletAddress":
"0xb6e8860883039b6db937639b94e9a10ff7971bb6"
    },
    "brazilianFiatSenderInfo": {
      "id": "36428ccf-3be4-471c-a243-2635eea3e142",
      "ticketId": "0b695784-81de-4ba4-888a-947a5efe4f55",
      "name": "Ada Capital Gestao de Recursos Ltda",
      "taxId": "45981761000100",
      "bankCode": "20018183",
      "branchCode": "0001",
      "accountNumber": "5703785980624896",
      "accountType": "payment",
      "endToEndId": "e20018183202504221424gkdf7gyqoru"
    },
    "blockchainReceiverInfo": {
      "id": "2734ee95-62d5-4bd2-a909-7e9322f59404",
      "ticketId": "0b695784-81de-4ba4-888a-947a5efe4f55",
      "walletAddress": "0xe41A4a64564D19f98867a4b43E743a7D988c9d68",
      "walletChain": "INTERNAL",
      "walletMemo": "",
      "txHash":
"0xf1b98a5a5b179f219d9bc4df68df44af335bd412ce3c5025fdf3d87fbe9d64fd"
    },
    "brlPixInputInfo": {
      "id": "a5bfd2bb-784a-4989-a105-52da968de113",
      "ticketId": "0b695784-81de-4ba4-888a-947a5efe4f55",
      "referenceLabel": "xe1thXY2aBq2naewSPfn7Xhfr",
      "additionalData": "Avenia Ticket Payment",
      "brCode": "00020126810014br.gov.bcb.pix01365c2c61a1-134b-4c34-
958f-ea3122ac717f0219Avenia Ticket
Payment520400005303986540610 0.205802BR5917Avenia API Ltda6009Sao
Paulo62290525xe1thXY2aBq2naewSPfn7Xhfr63041FA5"
    }
  }
}
```

Checking subAccount balance

Here, we'll retrieve all current balances for your or main account.

HTTP Get Request

```
https://api.sandbox.avenia.io:10952/v2/account/balances
```

Example Json Response

```
{
  "balances": {
    "BRLA": "165.00000",
    "USDC": "0",
    "USDT": "0",
    "USDM": "0"
  }
}
```

Convert BRLA to USDC or USDT

To decide where you want to transfer your BRLA to another stablecoin, simply set the outputCurrency to your chosen **one—USDT** or **USDC**.

WARNING

The quote is only valid for 15 seconds!

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate
```

Field	Value	Description	Required
inputCurrency	BRLA	The currency in which the mainAccount will receive funds. (in this case BRLA stable)	yes
inputPaymentMethod	INTERNAL	The method by which the mainAccount will receive funds (as an internal balance).	yes
outputAmount	100	The amount that the mainAccount receive (e.g., 98.58 = R$98.58).	yes
outputCurrency	USDT	The currency in which the mainAccount will receive funds. (in this case USDT stable)	yes

Field	Value	Description	Required
outputPaymentMethod	INTERNAL	The method by which the mainAccount will receive funds (as an internal balance).	yes
inputThirdParty	false	Since the mainAccount's funds are sourced from the own mainAccount, this field is not needed.	yes
outputThirdParty	false	For payments to a mainAccount, this should remain false (change to true if paying a third party).	yes
blockchainSendMethod	PERMIT	Defines the blockchain transaction type. I this case "PERMIT"	yes

cUrl Example

```
curl -X GET
"https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate?
inputCurrency=BRLA&inputPaymentMethod=INTERNAL&outputAmount=100&outputC
urrency=USDT&outputPaymentMethod=INTERNAL&inputThirdParty=false&outputT
hirdParty=false&blockchainSendMethod=PERMIT" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "inputCurrency": "BRLA",
  "inputPaymentMethod": "INTERNAL",
  "inputAmount": "582.446313",
  "outputCurrency": "USDT",
  "outputPaymentMethod": "INTERNAL",
  "outputAmount": "100",
  "markupAmount": "0",
  "markupCurrency": "",
  "blockchainSendMethod": "PERMIT",
  "inputThirdParty": false,
  "outputThirdParty": false,
  "appliedFees": [
    {
      "type": "Markup Fee",
      "description": "Total markup fees represented in the input
currency.",
      "amount": "0",
      "currency": "BRLA"
    },
    {
      "type": "In Fee",
      "description": "Fees due to input currency and input payment
method.",
      "amount": "0",
      "currency": "BRLA"
    },
    {
      "type": "Conversion Fee",
      "description": "Fees due to conversion from input currency to
output currency.",
      "amount": "3.494677",
      "currency": "BRLA"
    },
    {
      "type": "Out Fee",
      "description": "Fees due to output currency and output payment
method.",
      "amount": "0",
      "currency": "BRLA"
    },
    {
      "type": "Gas Fee",
      "description": "Fees due to blockchain transaction costs.",
      "amount": "0",
      "currency": "BRLA"
    }
  ],
  "basePrice": "5.7895",
  "pairName": "USDTBRLA"
}
```

Ticket - Closing the Order

With the quote in hand, we will initiate the order closing process—referred to as "Ticket"—where the quote is finalized and the requested operation is executed, which in our case is a transfer to our main account.

Field	Value	Descrip
quoteToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....	The quoteTol obtainec from the quote endpoin
ticketBlockchainOutput	beneficiaryWalletId	Contains details regardin the ticke blockcha output.
└ beneficiaryWalletId	00000000-0000-0000-0000-000000000000	Here we the ID of your ma account a zero id which makes Avenia A link this to your main account who mac the requ

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/account/tickets
```

Sample JSON Body

Remember that here we pass a zeroed uuid because we are linking your main account to this deposit (who will receive the funds)

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "ticketBlockchainOutput": {
    "beneficiaryWalletId": "00000000-0000-0000-0000-000000000000"
  }
}
```

cUrl Example

```
curl -X POST "https://api.sandbox.avenia.io:10952/v2/account/tickets" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" \
 -H "Content-Type: application/json" \
 -d '{
     "quoteToken":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
     "ticketBlockchainOutput": {
     "beneficiaryWalletId": "00000000-0000-0000-0000-000000000000"
  }
 }
```

JSON Response

INFO

If you've already registered a webhook for the `TICKET` event, you will start receiving webhook notifications from this point onward.

```
{
  "id": "b73767f7-1343-4176-9298-fffc85ea71a4"
}
```

Verify status from Ticket

INFO

An alternative to continuously checking the ticket history to track its current stage is to use webhooks instead.

Next, verify if the ticket status is PAID by checking the tickets by id endpoint.

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/account/tickets/YOUR-TICKET-
UUID-HERE
```

cUrl Example

```
curl -X GET
"https://api.sandbox.avenia.io:10952/v2/account/tickets/b73767f7-1343-
4176-9298-fffc85ea71a4" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response

INFO

Note: The status can be in: UNPAID PROCESSING PAID FAILED PARTIAL-FAILED

```
{
  "ticket": {
    "id": "b993375e-c819-4a63-85a5-2df6e90dd976",
    "workspaceId": "fbe2f6a3-6604-4b12-aae5-9f2994aaa3c1",
    "userId": "94fdb114-189f-46d2-bce8-6ee6ba461d18",
    "status": "PAID",
    "reason": "",
    "failureReason": "",
    "createdAt": "2025-04-22T14:52:07.171269Z",
    "updatedAt": "2025-04-22T14:52:17.173258Z",
    "expiresAt": "2025-04-22T14:57:07.169645Z",
    "quote": {
      "id": "7bd064bc-51de-4afa-b2dc-9fb33f6930b0",
      "ticketId": "b993375e-c819-4a63-85a5-2df6e90dd976",
      "inputCurrency": "BRLA",
      "inputPaymentMethod": "INTERNAL",
      "inputAmount": "582.847867",
      "outputCurrency": "USDT",
      "outputPaymentMethod": "INTERNAL",
      "outputAmount": "100",
      "markupCurrency": "",
      "markupAmount": "0",
      "sendMethod": "PERMIT",
      "inputThirdParty": false,
      "outputThirdParty": false,
      "basePrice": "5.793499",
      "appliedFees": [
        {
          "type": "Markup Fee",
          "amount": "0",
          "currency": "BRLA",
          "rebatable": false,
          "description": "Total markup fees represented in the input
currency."
        },
        {
          "type": "In Fee",
          "amount": "0",
          "currency": "BRLA",
          "rebatable": true,
          "description": "Fees due to input currency and input payment
method."
        },
        {
          "type": "Conversion Fee",
          "amount": "3.497087",
          "currency": "BRLA",
          "rebatable": true,
          "description": "Fees due to conversion from input currency to
output currency."
        },
        {
          "type": "Out Fee",
          "amount": "0",
          "currency": "BRLA",
          "rebatable": true,
          "description": "Fees due to output currency and output
payment method."
        },
        {
          "type": "Gas Fee",
          "amount": "0",
          "currency": "BRLA",
          "rebatable": false,
          "description": "Fees due to blockchain transaction costs."
        }
      ],
      "pairName": "USDTBRLA",
      "outputBrCode": "",
      "createdAt": "2025-04-22T14:52:06Z"
    },
    "blockchainSenderInfo": {
      "id": "31cea8c0-f998-41ed-ba51-4ed3412e20c0",
      "ticketId": "b993375e-c819-4a63-85a5-2df6e90dd976",
      "walletAddress": "0xe6B0847cE60Dd81C9DC55a8CC69F37343bFE5eF4",
      "txHash":
"0x84435e1777ef3af19609447f79511918a0195c43c728613e9c7883357d3868a7"
    },
    "blockchainReceiverInfo": {
      "id": "8fe21974-ebee-4e7c-9058-abc39b1881a9",
      "ticketId": "b993375e-c819-4a63-85a5-2df6e90dd976",
      "walletAddress": "0xe6B0847cE60Dd81C9DC55a8CC69F37343bFE5eF4",
      "walletChain": "INTERNAL",
      "walletMemo": "",
      "txHash":
"0x84435e1777ef3af19609447f79511918a0195c43c728613e9c7883357d3868a7"
    },
    "blockchainInputInfo": {
      "id": "7ae3719a-8267-4a54-a8f1-659a6e62f3ea",
      "ticketId": "b993375e-c819-4a63-85a5-2df6e90dd976",
      "r": "",
      "s": "",
      "v": 0,
      "nonce": 0,
      "deadline": 0,
      "personalSignature": "",
      "personalSignatureDeadline": 0
    }
  }
}
```

Transfer Tokens to Subaccounts

To send funds to your SubAccount, the process is very simple—when creating the ticket, just provide your SubAccount ID as the recipient of the tokens.

WARNING

The quote is only valid for 15 seconds!

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate
```

Field	Value	Description	Required
inputCurrency	USDT	The currency in which the mainAccount will receive funds. (in this case USDT stable)	yes
inputPaymentMethod	INTERNAL	The method by which the mainAccount will receive funds (as an internal balance).	yes
outputAmount	100	The amount that the mainAccount receive (e.g., 98.58 = R$98.58).	yes
outputCurrency	USDT	The currency in which the mainAccount will receive funds. (in this case USDT stable)	yes
outputPaymentMethod	INTERNAL	The method by which the mainAccount will receive funds (as an internal balance).	yes
inputThirdParty	false	Since the mainAccount's funds are sourced from the own mainAccount, this field is not needed.	yes
outputThirdParty	false	For payments to a mainAccount, this should remain false (change to true if paying a third party).	yes

Field	Value	Description	Required
blockchainSendMethod	PERMIT	Defines the blockchain transaction type. I this case "PERMIT"	yes

cUrl Example

```
curl -X GET
"https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate?
inputCurrency=USDT&inputPaymentMethod=INTERNAL&outputAmount=100&outputC
urrency=USDT&outputPaymentMethod=INTERNAL&inputThirdParty=false&outputT
hirdParty=false&blockchainSendMethod=PERMIT" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "inputCurrency": "USDT",
  "inputPaymentMethod": "INTERNAL",
  "inputAmount": "100",
  "outputCurrency": "USDT",
  "outputPaymentMethod": "INTERNAL",
  "outputAmount": "100",
  "markupAmount": "0",
  "markupCurrency": "",
  "blockchainSendMethod": "PERMIT",
  "inputThirdParty": false,
  "outputThirdParty": false,
  "appliedFees": [
    {
      "type": "Markup Fee",
      "description": "Total markup fees represented in the input
currency.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "In Fee",
      "description": "Fees due to input currency and input payment
method.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "Conversion Fee",
      "description": "Fees due to conversion from input currency to
output currency.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "Out Fee",
      "description": "Fees due to output currency and output payment
method.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "Gas Fee",
      "description": "Fees due to blockchain transaction costs.",
      "amount": "0",
      "currency": "USDT"
    }
  ],
  "basePrice": "1",
  "pairName": "USDTUSDT"
}
```

Ticket - Closing the Order

With the quote in hand, we will initiate the order closing process—referred to as "Ticket"—where the quote is finalized and the requested operation is executed, which in our case is a transfer to our sub account.

Field	Value	Descrip
quoteToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....	The quoteTo obtainec from the quote endpoin

Field	Value	Descrip
ticketBlockchainOutput	beneficiaryWalletId	Contains details regardin the ticke blockch output.
└ beneficiaryWalletId	1ee0a663-922b-4389-9f84-074ccff7085d	Here we the ID of your sub account

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/account/tickets
```

Sample JSON Body

Remember to provide the ID of the SubAccount you want to transfer the balances to.

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "ticketBlockchainOutput": {
    "beneficiaryWalletId": "1ee0a663-922b-4389-9f84-074ccff7085d"
  }
}
```

JSON Response

INFO

If you've already registered a webhook for the `TICKET` event, you will start receiving webhook notifications from this point onward.

```
{
  "id": "b73767f7-1343-4176-9298-fffc85ea71a4"
}
```

Checking subAccount balance

Here, we'll retrieve all current balances for your or subaccount.

> **INFO**
>
> Keep in mind that the operation is still being performed by the subaccount, so it's also necessary to include the subAccountId parameter here.

HTTP Get Request

```
https://api.sandbox.avenia.io:10952/v2/account/balances?
subAccountId=YOUR-SUBACCOUNT-ID-HERE
```

Example Json Response

```
{
  "balances": {
    "BRLA": "0",
    "USDC": "0",
    "USDT": "100",
    "USDM": "0"
  }
}
```

Send tokens from Subaccounts to External Wallets

Before starting the process of sending tokens to external wallets, we first need to identify those external wallets. That's where the topic of Beneficiary Wallets

Create Beneficiary Wallet

Registering a new beneficiary wallet is straightforward. Let's look at the required fields:

Fields

Field	Type	Required	Description
alias	string	Yes	A custom name for the wallet to help identify it.
description	string	No	A brief description of the wallet.
walletAddress	string	Yes	The blockchain wallet address.
walletChain	string	Yes	The blockchain network of the wallet (e.g., **POLYGON**, **CELO**, **ETHEREUM**, **GNOSIS**, **MOONBEAM** and **TRON**).
walletMemo	string	No	Memo for the wallet.

INFO

To perform this operation for a sub-account, pass the **subAccountId** field as a parameter to this endpoint.

HTTP Post Request

```
https://api.sandbox.avenia.io:10952/v2/account/beneficiaries/wallets?
subAccountId=SUB-ACCOUNT-ID-HERE
```

Sample JSON Body

```json
{
  "alias": "ExampleAlias",
  "description": "Example description",
  "walletAddress": "0xXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "walletChain": "POLYGON",
  "walletMemo": "memooo"
}
```

cUrl Example

```
curl -X POST
"https://api.sandbox.avenia.io:10952/v2/account/beneficiaries/wallets?
subAccountId=SUB-ACCOUNT-ID-HERE" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" \
-d '{
  "alias": "ExampleAlias",
  "description": "Example description",
  "walletAddress": "0xXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "walletChain": "POLYGON",
  "walletMemo": "memooo"
}'
```

JSON Response

```
{
  "id": "76971925-a1ca-423f-badf-0b3f2b03c51c"
}
```

Sending Tokens to a Beneficiary Wallet

Now that everything is ready and the beneficiary wallet has been registered, we just need to generate the quote.

INFO

Remember, the outputPaymentMethod must match the chain of the registered wallet.

WARNING

The quote is only valid for 15 seconds!

HTTP GET Request

```
https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate?
```

```
subAccountId=SUB-ACCOUNT-ID-HERE
```

Field	Value	Description	Required
inputCurrency	USDT	The currency in which the mainAccount will receive funds. (in this case USDT stable)	yes
inputPaymentMethod	INTERNAL	The method by which the mainAccount will receive funds (as an internal balance).	yes
outputAmount	50	The amount that the mainAccount receive (e.g., 98.58 = R$98.58).	yes
outputCurrency	USDT	The currency in which the mainAccount will receive funds. (in this case USDT stable)	yes
outputPaymentMethod	POLYGON	The method by which the external wallet will receive funds (must be the same as the beneficiary wallet registered chain).	yes
inputThirdParty	false	Since the mainAccount's funds are sourced from the own mainAccount, this field is not needed.	yes
outputThirdParty	false	For payments to a mainAccount, this should remain false (change to true if paying a third party).	yes
blockchainSendMethod	PERMIT	Defines the blockchain transaction type. I this case "PERMIT"	yes

INFO

> To perform this operation for a subAccount, pass the **subAccountId** field as a parameter to this endpoint.

cUrl Example

```
curl -X GET
"https://api.sandbox.avenia.io:10952/v2/account/quote/fixed-rate?
inputCurrency=USDT&inputPaymentMethod=INTERNAL&outputAmount=100&outputC
urrency=USDT&outputPaymentMethod=POLYGON&inputThirdParty=false&outputTh
irdParty=false&blockchainSendMethod=PERMIT&subAccountId=1ee0a663-922b-
4389-9f84-074ccff7085d" \
 -H "Authorization: Bearer
eyJhdXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

JSON Response

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "inputCurrency": "USDT",
  "inputPaymentMethod": "INTERNAL",
  "inputAmount": "100.050089",
  "outputCurrency": "USDT",
  "outputPaymentMethod": "POLYGON",
  "outputAmount": "100",
  "markupAmount": "0",
  "markupCurrency": "",
  "blockchainSendMethod": "PERMIT",
  "inputThirdParty": false,
  "outputThirdParty": false,
  "appliedFees": [
    {
      "type": "Markup Fee",
      "description": "Total markup fees represented in the input
currency.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "In Fee",
      "description": "Fees due to input currency and input payment
method.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "Conversion Fee",
      "description": "Fees due to conversion from input currency to
output currency.",
      "amount": "0",
      "currency": "USDT"
    },
    {
      "type": "Out Fee",
      "description": "Fees due to output currency and output payment
method.",
      "amount": "0.050025",
      "currency": "USDT"
    },
    {
      "type": "Gas Fee",
      "description": "Fees due to blockchain transaction costs.",
      "amount": "0.000063",
      "currency": "USDT"
    }
  ],
  "basePrice": "1",
  "pairName": "USDTUSDT"
}
```

Ticket - Closing the Order

With the quote in hand, we will initiate the order closing process—referred to as "Ticket"—where the quote is finalized and the requested operation is executed, which in our case is a transfer to our sub account.

Field	Value	Descrip
quoteToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....	The quoteTo obtaine from the quote endpoin
ticketBlockchainOutput	beneficiaryWalletId	Contain details

Field	Value	Descrip
		regardin the ticke blockcha output.
└ beneficiaryWalletId	1ee0a663-922b-4389-9f84-074ccff7085d	Here we the ID of your benefici wallet registrec

HTTP POST Request

```
https://api.sandbox.avenia.io:10952/v2/account/tickets
```

Sample JSON Body

Remember to provide the ID of the SubAccount you want to transfer the balances to.

```
{
  "quoteToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.quoteToken....",
  "ticketBlockchainOutput": {
    "beneficiaryWalletId": "76971925-a1ca-423f-badf-0b3f2b03c51c"
  }
}
```

JSON Response

INFO

If you've already registered a webhook for the `TICKET` event, you will start receiving webhook notifications from this point onward.

```json
{
  "id": "b73767f7-1343-4176-9298-fffc85ea71a4"
}
```