How to integrate with Sparados API

Sparados API

Onboarding

Here's a short instruction for the essential steps before calling Sparados API:

Essential steps before calling Sparados API:

1. Create an x509 Certificate for Server-to-Server Communication

This API requires Mutual TLS authentication. You can use the same certificate for all mTLS-secured APIs exposed by Sparados. If you don't have one, follow this instruction: Connecting to server-to-server APIs

2. Receive Corporation Information from Sparados

- After Sparados signs the certificate, you will receive the following information from them:
- Corporation ID
- Balance ID
- Signed certificate (name: V-{Companyname}-BudgetControl)
- Optionally, you may also receive personalized visuals and their IDs for your cards.

3. API addresses

Instance	BETA	PROD
Cards API	https://sparados-bc-api.secure-verestro.dev/secure	https://sparados-bc-api.secure-verestro.com/secure
Transaction API	https://services.upaidtest.pl/thc	https://services.upaid.pl/thc

4. Assign Cards to Corporation Members

- Now, you can assign cards to your corporation members using the information received from Sparados.
- Remember to save the `approvalId` on your side because it will be needed in case of editing it

later.

By following these steps, you'll be ready to call Sparados API with the necessary credentials and information for your corporation's budget control and card management.

Swagger documentation is in the end of document

Values used in Sparados Cards API

Date format

We use ISO 8601 standard: year-month-day-T-hour-minute-second-Z

where T is constant separating the date from the time and Z is Zulu time (Greenwich Mean Time (GMT)).

Example: 2023-05-22T10:00:01.953Z

Minor values

All numeric values used in our API are in minor. If you want to e.g. assign card with limit 100.50 EUR you have to send us 10050. Currency is not needed - it's always the same as currency of account/balance.

Currency

Card/approval is always created in the currency of account. Account can have only one currency although it is possible to make transactions in other currencies.

ID

We always use UUID format for IDs in our project.

How to assign card?

Use the endpoint **POST /secure/approvals** to initiate the process of assigning a card and provide the necessary data listed below. Required fields are signed with *.

rules.budgetMinor *	Amount in minor that user can spend using a card provided in the currency of account/balance.
rules.validityPeriod *	validity of a card in UTC format
rules.timezone *	BC API will adjust startDate and EndDate to exact time in provided timezone
limits.amountMinor	Additional periodic limit value in minor provided in the currency of account/balance.

limits.timeUnit	Available values: daily, weekly, monthly
limits.type	Available values: general, ecommerce, atm, foreign_amount. Types are descibed below the table.
purpose.ecommerce *	Now it should be always as true.
purpose.allowChangeRequest *	True - user can send a request to change limits on card False - user can't send a request to change limits on card
user.email *	E-mail on which will be sent information about assigned card.
user.prefix *	Prefix of a phone number of a user that can be used for authorization purpose
user.phone *	Phone number of a user that can be used for authorization purpose
balance.Id *	Id of a balance created for you corporation
card.description	Description displayed on a card visual in user app
card.visual.Id	Id of visual provided by Sparados. Null - default Sparados visual.

Additional limits

It's possible to set additional limits on the card, which are optional. If left null, the card issuing service will set default values.

There are four types of additional limits:

1. **General limit** - periodic limit on all kinds of transactions. This value should be greater than remaining additional limits because it affects them. E.g. if the ATM limit is set to 100 EUR and the general limit is set to 50 EUR, the end user will be able to withdraw only 50 EUR from the ATM. Possible values: amount in minor or null.

2. **Online payment limit** - periodic limit on e-commerce payments. Possible values: 0, amount in minor or null.

3. **ATM limit** - periodic limit on ATM withdrawals. Possible values: 0, amount in minor or null.

4. **Foreign transaction limit** - periodic limit on transactions in a different currency from account currency. Limit is given in the currency of balance and it will be recalculated taking into account exchange rates and commission. Possible values: 0, amount in minor or null.

Possible time units for all additional limits are daily, weekly, and monthly

Possible values od the additional limits:

0 - disabled. End user won't be able to perform online payments, withdraw from ATM or transactions in foreign currencies.

Value in minor - limited to requested value

Null - unlimited. End user will be able to perform online payments, withdraw from ATM or transactions in foreign currencies.

Difference between approval and reapproval:

Approval is created by the corporation in the process of assigning a card. We don't name it simply a card because a card is generated when the end-user redeems a card and the start date of approval occurs. Approval is created earlier, right after calling endpoint **PUT /secure/approvals/**.

Reapproval can be created by the corporation while editing data on approval (using method **PUT /secure/approvals/{id}**) in status Delivered or by mobile user when they request a higher limit or change of validity end date. Requesting changes can be disabled by putting the value false in purpose.allowChangeRequest field while creating an approval.

Ensure that you provide all the required fields and follow the instructions to successfully assign a card with the specified limits and rules.

The actions you can perform on an approval along with the corresponding statuses and available actions:

Statuses of approvals:

CREATED	Occur only when mobile user request changes on reapproval.
ACCEPTED	When an Approval is created. **It changes to DELIVERED status after registration is complete and card is redeemed by end-user.**
CANCELED	When an Approval or Reapproval with status CREATED, ACCEPTED or PREPARED is cancelled by corporation.

REJECTED	When a Reapproval with status CREATED is rejected by corporation.
EXPIRED	When a Reapproval with the status CREATED is not accepted/rejected or status of Approval is ACCEPTED but the card has not been redeemed by mobile user within the given Approval time (until the end of the ValidTo date).
PREPARED	When approval has been ACCEPTED, redeemed by the end-user and is waiting for card generation. Card is generated on startDate of Approval.
DELIVERED	When mobile user redeemed card. This is the state of approval in which transactions can be made with a card.
FINISHED	When an Approval with status DELIVERED exceeds the end date of assignment.
REAPPROVED	When a reapproval is created for a given Approval that is in Accepted/Delivered status.
LOCKED	This is a status of a **card (not an approval)** but we display it on the diagram below to show that Delivered is the only state of a approval on which we have action to lock and unlock card.

Statuses of an Approval:

image-1686858492185.png

Statuses of a Reapproval:

image-1686858497780.png

Actions available on approvals:

- Created: Accept, Reject, Edit, Cancel.
- Accepted: Cancel
- Preprared: Cancel
- Delivered: Lock, Unassign, Edit
- Locked: Unlock
- Rejected: —
- Expired: —
- Finished: —

Important! Reapproval Acceptance:

- When a reapproval is accepted, its status changes to "Delivered."
- The accepted reapproval replaces the previous approval. Therefore, it's crucial to save the ID of the accepted reapproval because, from that moment, it becomes the valid ID of this approval.
- Concurrently, the status of the original approval changes to "Reapproved."

Unassign card from enduser

If the user has already redeemed the card (status of approval is Delivered), use the **PUT /secure/approvals/{id}/unassign** method to take away the card from the end-user.

If the user hasn't redeemed the card yet (status of approval is Accepted or Prepared), use the **PUT /secure/approvals/{id}/cancel** method to take away the card.

Increase or decrease available limit on a card

To quickly increase the available limit on a card, use the PUT /secure/approvals/{id}/increase_budget method.

To add another 10 EUR to the available limit, put the value: "additionalBudgetMinor": 1000 (assuming the minor currency is used with 1 EUR = 100 minor).

To quickly decrease the available limit on a card, use the same method PUT /secure/approvals/{id}/increase_budget but with a negative value.

To subtract 10 EUR from the available limit, put the value: "additionalBudgetMinor": -1000.

What e-mails are sent to the end user?

E-mails can be adjusted to your corporation and sent in different language. Contact Sparados for details.

Email name	Sending reason
BC_USER_PENDING_TERMS_AND_CONDITIONS	Sent while new terms and conditions are applied to the issuer to end users.
BC_USER_CARD_EXPIRING	Sent 1 day before card expiration to end user.
BC_USER_PENDING_CARD	Sent to end user when new card was assigned and is waiting for redemption
BC_USER_NEW_CARD	Sent to end user when new card was redeemed
BC_USER_CARD_EXPIRED	Sent to end user when approval is finished.
BC_USER_CARD_DELETED	Sent to end user while card/approval was uassigned or deleted by corporation.
BC_USER_CARD_LOCKED	Sent to end user when card was locked by corporation.

Sparados Transaction API

Description

Sparados Transaction History API consists of inbound and outbound APIs*(Outbound API is only tagged because OAS version which is used in this document is elder than OAS 3.1 in which webhooks was introduced)*

Inbound Authentication

This API requires Mutual TLS authentication. You can use the same certificate for all mTLS-secured APIs exposed by Sparados. If you don't have one, follow this instruction: Connecting to server-to-server APIs

Outbound Authentication

During outbound call, server will present own certificate which is signed by Verestro (technology provider) CA and will proof possession of the private key during TLS handshake. Client can put configuration to trust all certificates signed by given CA. Every environment has own CA certificate and client has to use corresponding to environment root CA certificate

Backward compatibility

Any additional development in service will be backward compatible. Changes below are considered to be backward compatible, and client should not brake if any change will appear in future:

- adding a new endpoint.
- adding a new, optional, request parameter to an existing endpoint. The parameter can be added as part of the request body, as a URL parameter, or an HTTP header
- adding a new enum value. The value is added either in the request or in the response.
- relaxing some of the constraints on an existing request parameter. For example, making it optional
- adding a new response parameter to the API response

Summing up: Client should ignore any unknown fields or enum values received as part of API responses

Outbound Retry Strategy

Outbound calls that fail with a timeout, connection failure, or an HTTP response code of 5xx, server will automatically retry 3 times with up to a 5-second wait between each try. If the call has not succeeded after the initial retries server will attempt a second round of 3 retries with increasing time intervals between each retry. Between attempts the system will wait 15 minutes, 30 minutes, and then 2 hours. If client server will not respond and retries are exhausted then call is dismissed

Transaction API

https://developer.verestro.com/books/transaction-history-api/page/technical-documentation-thc-external-api

Cards API Swagger Documentation

@swagger="https://sparados-bc-api.upaidtest.pl/api-secure.yaml"

Connecting to server-to-server APIs

Environments

We have three environments available for our partners.

- Sandbox - this is shared environment, available as demonstration. You can get access before signing a deal with us - product@sparados.com to get access.
- Beta - This is environment you're going to use during integration with our services.
- Production - This is live environment.

Authorization

Our APIs are secured with Mutual TLS Authentication. You will need certificate signed by us in order to connect. To get a certificate please send us an Certificate Signing Request (CSR). We will sign it and return a valid certificate in a response.

CSR should have following structure:

For Beta/Production

You will need separate certificates for Beta and Production environments, however they can be generated in exactly the same way.

Field	Example value
Common Name (CN)	`V-Sparados-{CompanyName}` Company Name should be replaced with name of your company.
Organization (O)	Name of your company
Organizational Unit (OU)	[UUID] corp_id sent by Sparados admin
Locality (L)	City where your organization is located
State/County/Region (ST)	State/County where your organization is located
Country (C)	Two-letter country code where organization is located (ISO 3166-1 alpha-2)

Field	Example value
Email Address	Email to be contacted in case of forced renewal

Example CSR generation command for Beta and Production

> **Remember to replace parameters and file names with your own data**
> Commas cannot be used in parameters
> Minimum key length is RSA 2048

```
openssl req -new -newkey rsa:4096 -keyout companyName.key -out companyName.csr -nodes -subj
'/C=US/ST=Florida/L=Miami/O=SomeCompany/OU=UUID/CN=V-Sparados-
CompanyName/emailAddress=example@user.com'
```