esimba.ai

esimba.ai API Documentation

Dig into our API reference documentation and quickstarts. esimba.ai API documentation provides instructions on where to find API keys to access esimba.ai API. You're covered with everything from the query structure and API requests with detailed explanations on receiving lists of lines and bundles that belong to your Partner Account.

esimba.ai API is a primary way to get data in and out of the esimba.ai Console. You can use esimba.ai API to control your account, view balance reports, receive a list of lines that belong to your Partner Account, change line status, receive usage data, refill lines, etc.

Version 1.13

Revision History

Version	Date	Description of change
1.1	01.02.2022	Initial version
1.2	03.05.2022	Added more descriptions to swagger API
1.3	04.05.2022	Updated swagger API styles
1.4	20.01.2023	Added description about webhooks, auto-refill and bill protection
1.5	06.03.2023	Updated description about Payment Gateway
1.6	29.08.2023	API Document Updated Updated description about Partner payments setup Updated parameters description about Transactions Request
1.7	22.04.2024	Updated Authentication description Updated descriptions and FAQs for Lines Request Updated parameters description about Transactions Request Updated descriptions for Network providers and Countries Requests Added description about API Commands Usage Added description about Integration Process Added description and examples for Account Balance Request
1.8	28.10.2024	Updated description about GET\bundles command usage
1.9	02.02.2025	Updated description about Lifetime bundles in the GET/ product_types command
1.10	02.05.2025	Removed Payment Gateway from the API documentation
1.11	17.06.2025	Added the following webhooks stripe.paymentIntend.succeeded, simcard.usage.low_data, bundle.added, bundle.removed
1.12	01.07.2025	Added Webhook Signature Verification
1.13	19.08.2025	Added filter parameter to POST /line/create

Contents

01. Overview

We have designed the esimba.ai API in a RESTful way to ensure smooth communication between the client (frontend) and the server (backend). Anytime you request a server using a REST API, you receive responses via HTTP functions described in this documentation.

What is REST API?

API stands for “Application Programming Interface.” It’s a set of rules enabling programs to talk to each other, exposing data and functionality across the Internet in a consistent format.

REST stands for “Representational State Transfer.” This architectural pattern describes how distributed systems can expose a consistent interface. When people use the term ‘REST API,’ they generally refer to an API accessed using
the HTTPS protocol at a predefined set of URLs.

These URLs represent various resources — any information or content accessed at that location.
Often resources have one or more methods that can be performed over HTTPS.

At eSIMba , we use GET, POST, and PUT methods.

- **GET** — used to retrieve resources,
- **POST** — to create resources,
- **PUT** — to update resources

Connection

esimba.ai REST API is a server over HTTPS.
To ensure data privacy, unencrypted HTTP is not supported.

HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses and digitally sign those requests and responses.

HTTPS is far more secure than HTTP.

Timestamps

The server time zone is UTC.

There must be consistency across all date-related data.
All requests to a server must be aligned to UTC hour boundaries.

Countries

All country entries are in ISO 3166-1 alpha-2 format.

The purpose of ISO 3166 is to define internationally recognized codes of letters and/or numbers that we can use when we refer to countries and their subdivisions.

Currency

USD is the default and the only allowed currency.

Media types

All data is sent upon a user submitting GET/POST/PUT requests with parameters through the requested URL.

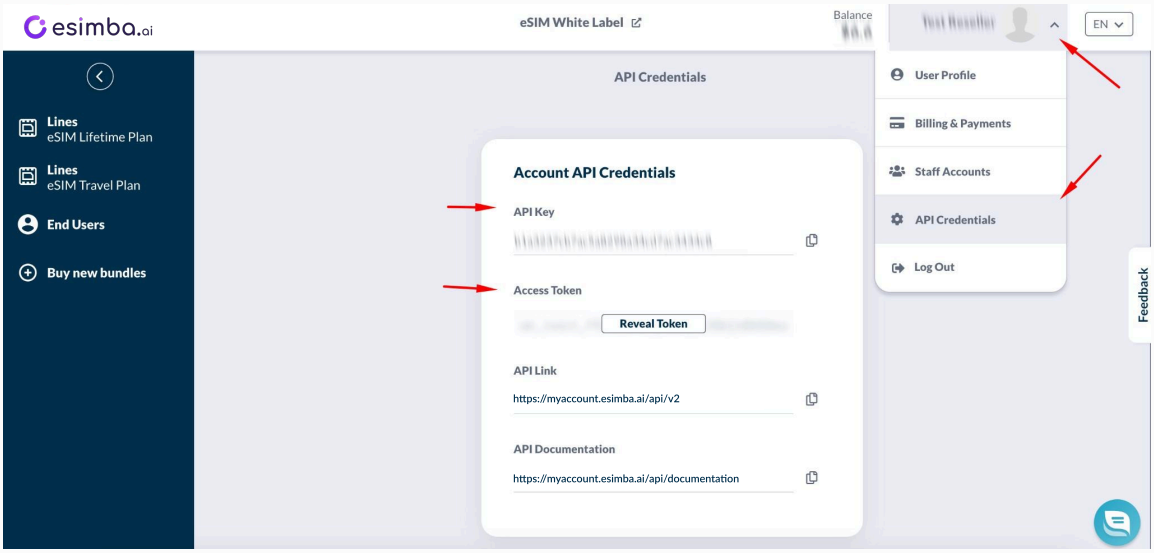
02. Authentication

esimba.ai authenticates your API requests using your account's API keys representing the required credentials. Each request has to include API Credentials

Example:

```
curl -X GET "https://{your_link}/api/v2/lines?per_page=3&page=2" -H "apiKey: Your_apiKey"
-H "accessToken: Your_accessToken"
```

You can find the API Keys under "API Credentials" in your account (see the image below).
If you cannot find your API Keys, please contact your Personal Manager.



Implement the following security measures to ensure the privacy and security of your users' eSIM data, when using esimba.ai API.

Each API request has to include API credentials, such as apiKey and accessToken, that you can find in your account.
Don't share this information with anyone. We don't save any payment information from users.
All API requests should use HTTPS URLs only.

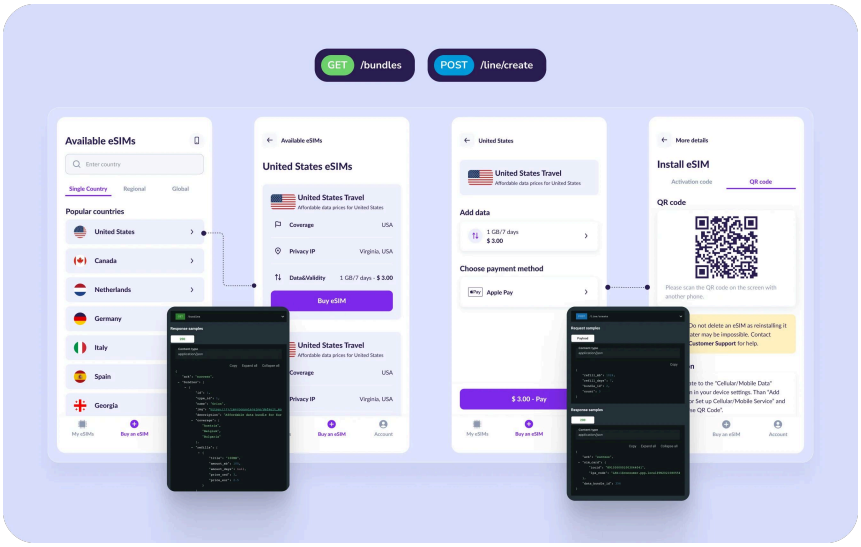
Use standard security measures: Keep passwords/keys safe and only request API via HTTPS.

03. API Commands Usage

Consider the following example demonstrating API Commands usage for eSIM purchase.

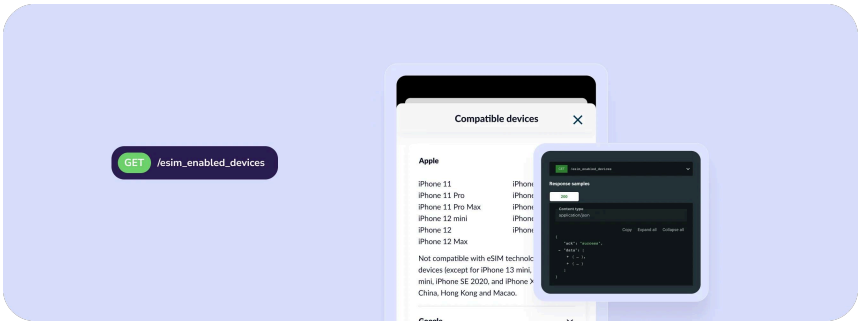
1. A user selects a country within the partner's app or platform and receives a list of bundles through the /bundles command.
2. They browse the eSIM bundles, select the desired bundles, and proceed to payment.
3. After a successful payment, the line is created using the /line/create command.
   The partner receives the payment, and funds are deducted from the account balance.
4. For eSIM installation, the user receives the LPA code (via the same API command), which can be transformed into a QR code.

*The visuals presented in the document are just one example of the development of API commands.
The actual implementation may vary depending on the partner's specific requirements.

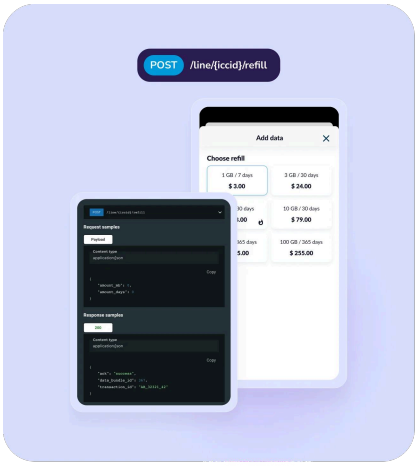


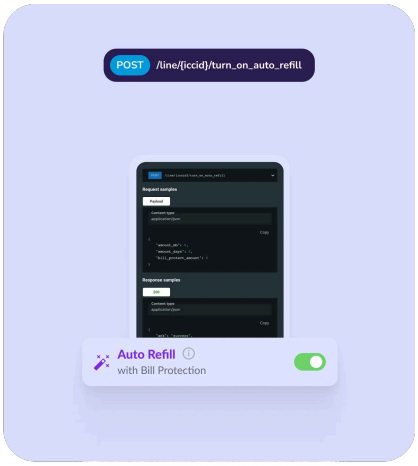
eSIM DEVICE COMPATIBILITY CHECK

You can offer users the option of checking that their device supports eSIM technology before purchasing an eSIM line to avoid installation issues. This is not mandatory but rather advised. /esim_enabled_devices API command offers an up-to-date list of eSIM-enabled devices that partners can display on their platform.



Users can refill their line with additional data amounts when they exhaust their available data. The available refills can be seen through the command /line/{ICCID}/available_refills. Although in most cases, they are standardized.
Users can activate the Auto-refill feature to ensure they never run out of balance. It automatically adds data to their line when it falls below 100 MB or when the data validity expires. /line/{ICCID}/turn_on_auto_refill. command operates similarly to line creation or refilling: the chosen refill is automatically added to the specific line (via ICCID).





04. Integration Process

For a streamlined integration of the esimba.ai API, it is recommended to develop an integration module incorporating
key API commands:

- /lines
- /line/{iccid}/get_details
- /line/{iccid}/available_refills
- /line/create
- /line/{iccid}/refill

This list encompasses essential commands to begin implementation. Additional API commands, though optional, can provide users with supplementary information. Detailed descriptions of each command and comprehensive information can be found in the API documentation below.

It is advised to have an independent backend to handle requests to the esimba.ai API. Direct client application requests are discouraged to prevent exposure of the partner key.

The recommended scheme involves routing requests: **mobile application >> partner API >> esimba.ai API.**

Organizations employing payment chains must confirm payment completion before submitting /line/create and /line/{iccid}/refill requests to us.

Activation following purchase or refill is automatic; no further activation steps are required.

05. Making requests

HTTP METHODS

Please use HTTPS requests because it's more secure than HTTP.
Permitted HTTP methods are:

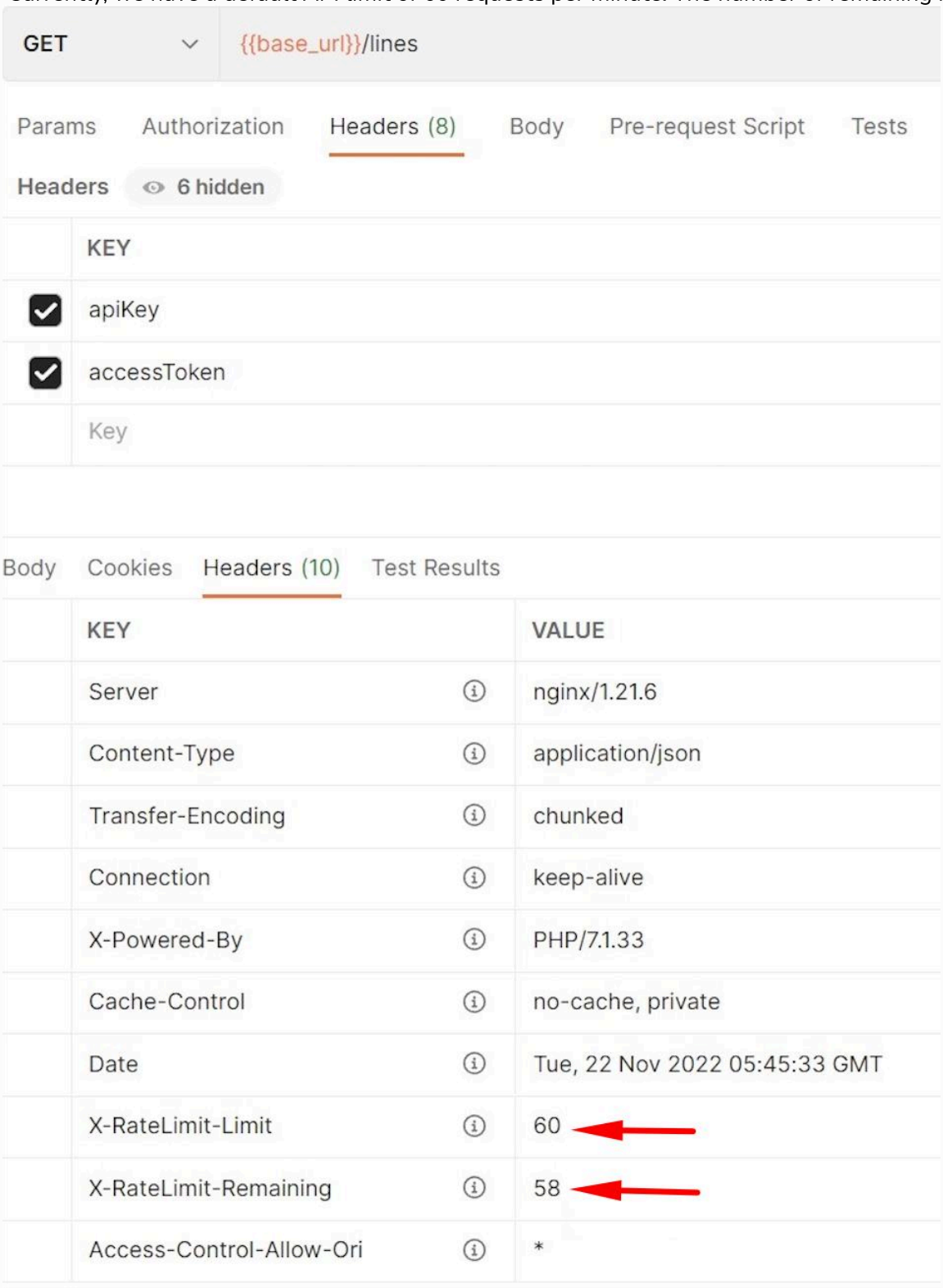
HTTP Method	Description
GET	The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.
POST	The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server.
PUT	The PUT method replaces all current representations of the target resource with the request payload.

RATE LIMIT

All calls within the esimba.ai API are allotted a specific number of requests per refresh period. Each API request returns
the following header information regarding rate limits and the number of requests left.

Depending on the endpoint you are trying to reach, it will have a specific number of allowed requests per refresh period.

Currently, we have a default API limit of 60 requests per minute. The number of remaining requests can be found



STATUS AND ERROR CODES

Sometimes, your API call may generate an error. Here you will find additional information about what to expect if you don't format your request correctly or if we fail to process your request properly.

Name	Description
200 OK	The request was successful
401 UNAUTHORIZED	The supplied credentials are insufficient to access the resource
403 FORBIDDEN	The response status code indicates that the server understands the request but refuses to authorize it
404 NOT FOUND	Resource was not found
405 METHOD NOT ALLOWED	Resource can not be accessed with this method (GET, POST, etc.)
500 SERVER ERROR	Representation could not be returned due to an internal server error
503 SERVICE UNAVAILABLE	Representation can not be returned due to a temporary service outage

FORMATTING YOUR REQUEST

Accept Header

The API provides JSON responses and requires the accept header. The API uses application/json.

Example:

```
curl -X GET "https://{your_link}/api/v2/lines?page=1&per_page=25" -H "accept: application/json"
```

Request Body

When submitting data to a resource via POST or PUT, you must submit your payload in JSON

GET	curl -L -X GET "https://{your_link}/api/v2/lines" -H " apiKey: Your_apiKey" -H "accessToken: Your_accessToken" -H "Content-Type: application/json" --data-raw "{ "per_page": 3, "page": 2 }"
POST	curl -L -X POST "https://{your_link}api/v2/line/create" -H "apiKey: Your_apiKey" -H "accessToken: Your_accessToken" -H "Content-Type: application/json" --data-raw "{ "refill_mb": 1024, "refill_days": 7, "bundle_id": 13 }"
PUT	curl -L -X PUT "https://{your_link}/api/v2/line/8934071179000613677/ update_notes" -H "apiKey: Your_apiKey" -H "accessToken: Your_accessToken" -H "Content-Type: application/json" --data-raw "{ "notes": "updated notes" }"

PAGINATION

You can get paginated results (e.g., you can get the first 25 elements, the second 25 elements, etc.).
All details can be found in the request section. Below is a basic example of sending a request to get paginated results.

Example:

```
curl -X GET "https://{your_link}/api/v2/lines?page=1&per_page=25" -H "accept: application/json"
```

SUCCESSFUL REQUESTS

Below is a general overview of what resource objects return with successful API requests.

HTTP Verb	Resource object returned
GET	Returns a single resource object or array of resource objects
PUT	Returns the updated resource object
POST	Returns the newly created resource object

06. Webhooks

Webhooks are a convenient way to keep track of line usage and changes in the status of your esimba.ai lines. You can subscribe to events for your products and payments by activating webhooks that will push notifications to a given URL. We use HTTPS to send notifications about selected events.

Here is how esimba.ai webhooks work:

- The partner can choose events about which webhooks will be sent;
- When certain events occur, webhooks are sent to the partner;
- Data in webhooks of different events is different;
- The partner can specify the URL to which webhooks will be sent to respond to changes;
- The partner can change the URL to which webhooks will be sent;
- The partner can remove the URL to which webhooks were sent to stop receiving them;
- The partner can view the entire list of webhooks sent to them and their responses;
- A webhook is considered delivered if it receives status 200 in response;
- The undelivered webhook is not resubmitted.

WEBHOOK SIGNATURE VERIFICATION

To ensure the authenticity of incoming webhook requests, each payload includes a cryptographic signature in the X-Keepgo-Signature header. You can validate this signature to confirm the request originates from Keepgo.

Set Secret Key
At myaccount.keepgo.com, in the User Profile menu, choose Webhooks > Create a Secret Key.

Signature Header
The server sends the signature in the X-Keepgo-Signature header.

Example(Http):

```
X-Keepgo-Signature: "abc123...xyz456"
```

Signature Generation

The signature is computed as (PHP):

```
$signature = base64_encode(hash_hmac("sha256", $rawPayload, $secretKey, true));
```

Where:
$rawPayload = Raw request body (JSON string).
$secretKey = Your webhook`s secret key (provided during setup).

Verification Steps

1. Extract the X-Keepgo-Signature from the request headers.
2. Recompute the signature using the received payload and your secret key.
3. Compare the computed signature with the header value. If they match, the request is valid.

Example(Http):

```
$receivedSignature = $_SERVER["HTTP_X_KEEPGO_SIGNATURE"];
$rawPayload = file_get_contents("php://input"); // Raw JSON body
$secretKey = 'xxxxxxxxxx'; // Your secret key here, should be kept secure
$signature = base64_encode(hash_hmac("sha256", $rawPayload, $secretKey, true));
if (hash_equals($receivedSignature, $signature)) {
  // Valid request
}
```

STEPS TO RECEIVE WEBHOOKS

You can start receiving event notifications using the steps described below.

1 **Specify the URL to which webhooks will be sent**
You can enable/disable and change URL for webhooks in your Partner Account.
Please contact your Personal Manager if you need help with the webhooks setup.

2 **Select the events you want to be notified about**
The webhook events we currently support are listed in the table below

Name	Description	Example
simcard.usage.changed	Occurs when the Line's usage changes	{ "event": "simcard.usage.changed", "data": "iccid": "12345678902116678", "deactivation_date": "", "allowed_usage_kb": 1048576, "remaining_usage_kb": 1048576, "remaining_days": null, "status": "Registered", "data_bundles": [{ "id": 14, "status": "Non-active", "allowed_usage_kb": 1048576, "active_kb": 0, "remaining_usage_kb": 1048576, "validity": 7, "assigned_at": "2022-09-22 23:17:27", "activated_at": null, "terminated_at": null, "expire_at": null }] } }
stripe.paymentIntend.succeeded	Occurs when a payment is successfully completed through the Stripe payment gateway. This feature is only available if you have set up a Payment Gateway account with us.	We forward the webhook received from Stripe directly to the partner's specified URL. The payload contains the full original data from Stripe.
simcard.usage.low_data	Occurs when the Line's usage equals or falls below 100 MB.	{ "event": "simcard.usage.low_data", "timestamp": "2025-06-09T07:36:09.618973Z", "data": { "iccid": "12345678902116678", "user_email": "some-email@example.com", "remaining_data_mb": 90.23, "remaining_mb_threshold": 100 } }

Name	Description	Example

bundle.added	Occurs when a bundle is added to the platform.	{
		"event": "bundle.added",
		"timestamp": "2025-06-09T06:55:15.069936Z",
		"data": {
		"bundle": {
		"id": 352,
		"name": "Octans",
		"description": "LATAM eSIM: 17 countries, Claro”
		"bundle_type": "regional",
		"coverage": [
		"Argentina",
		"...",
		"Uruguay"
		],
		"refills": [
		{
		"title": "1 GB",
		"amount_mb": 1024,
		"price_usd": 28
		},
		"...",
		{
		"title": "50 GB",
		"amount_mb": 51200,
		"price_usd": 725
		}
		],
		"privacy_ip": "Netherlands",
		"networks": [
		{
		"title": "Argentina",
		"local_networks": [
		"Claro"
		],
		"flag": "https://myaccount.keepgo.com/img/flags/3x2/ar.
		},
		"...",
		{
		"title": "Uruguay",
		"local_networks": [
		"Claro"
		],
		"flag": "https://myaccount.keepgo.com/img/flags/3x2/uy.
		}
		]
		}
		}
		}

bundle.removed	Occurs when a bundle is removed from the platform.	The bundle.removed webhook has the same structure as bundle.added.

3 **Receive a response object**
Your Webhook URL should return a response with HTTP status code 200. Any other HTTP response code will be treated as a failure

07. Lines Request

A line is an eSIM profile with an activation code. Each line has a bundle. Every Partner's account may have lines of any bundle type. A line
can belong only to one account at a given time. Each line has a unique ICCID of 19 to 20 characters, Privacy IP location, Countries & Networks coverage, APN settings, line status, and special notes.

Lifetime and Travel line statuses:
1. **Registered** — the line is ready for operation but has never been used (without usage)
2. **Activated** — the line's usage has started (the first usage has been recorded)
3. **Deactivated** — the line is turned off (either by force, upon reaching expired at, or when the balance is over)

Please note that the line status isn't the same as the bundle status, but it can be a function of the bundle statuses.

GET/lines

Use: This request returns a list of lines belonging to the account, their configuration, and running parameters, including data usage, rate plan, billing cycle, count of data, and any notes they may have.

Note: The API command /lines with pagination or /lines/filter_data with specific filters can list all lines with details.
The command /line/{iccid}/get_details can be used for information regarding a specific line.
Usage statistics are not directly provided through the API; only the current value of remaining usage is available. Additional usage information and billing details can be obtained through the /line/{iccid}/transactions command upon request.

Parameters

Name	Description
page integer (query)	Default value : 1
per_page integer (query)	Default value : 25

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"sim_cards": {
		"from": 1,
		"to": 2,
		"last_page": 1,
		"per_page": 25,
		"total": 2,
		"current_page": 1,
		"items": [
		{
		"iccid": "8934521179000123456",
		"deactivation_date": "",
		"expired_at": "2022-08-21 11:57:27",
		"allowed_usage_kb": 102400,
		"remaining_usage_kb": 51200,
		"remaining_days": 5,
		"status": "Activated",
		"bundle": "Aquila",
		"notes": "",
		"data_bundles": [
		{
		"id": 42,
		"status": "Active",
		"allowed_usage_kb": 102400,
		"active_kb": 51200,
		"remaining_usage_kb": 51200,
		"validity": 10,
		"assigned_at": "2022-08-11 11:57:27",
		"activated_at": "2022-08-11 11:57:27",
		"terminated_at": null,
		"expire_at": "2022-08-21 11:57:27"
		}
		]
		}
		]
		}
		}

401	Unauthenticated	
403	Forbidden	

GET/lines/filter_data

Use: Receives a list of possible Line statuses (Registered, Activated, Deactivated) and auto refill status IDs for filtering purposes in other API requests.

Responses

Code	Description	Example

200	Successful operation	<pre>{
		  "ack": "success",
		  "filters": {
		  "status_id": {
		  "1": "Registered",
		  "2": "Activated",
		  "3": "Deactivated"
		  },
		  "auto_refills_status_id": {
		  "0": "Off",
		  "1": "On"
		  }
		  }
		}</pre>

GET/line/{iccid}/available_refills

Use: This request receives a list of available refills for a Line. It displays such values as the amount of data, price (in USD and EUR), and the number of days for all available refills for the chosen iccid.

FAQ

Is it better to display available refills based on /line/{iccid}/available_refills or /bundles/{id} refills[*]?
It's better to display refills to the user based on available refills for a specific line. The request is faster completed using the /line/{iccid}/available_refills command. Both /line/{iccid}/available_refills and /bundles/{id} have the same refills. It is better and faster to find the available top-ups for a specific line using /line/{iccid}/available_refills.

Parameters

Name	Description
iccid *** required** string **(path)**	iccid

Responses

Code	Description	Example

200	Successful operation	<pre>{
		  "ack": "success",
		  "available_refills": [
		  {
		  "title": "1GB",
		  "amount_mb": 1024,
		  "amount_days": 10,
		  "price_usd": 7,
		  "price_eur": 6.5
		  }
		  ]
		}</pre>

GET/line/{iccid}/get_details

Use: This request receives details for the specific Line, including Line's usage, remaining data/days, auto refill info, currency, etc.

FAQ

What is a "deactivation_date"?
The "deactivation_date" is a date when the line will be automatically deactivated in the future.
There is also an option to set the past date. In this case, the line will be immediately deactivated.

Parameters

Name	Description
iccid * required string **(path)**	iccid

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"sim_card": {
		"iccid": "8965972579000123456",
		"msisdn": "34590100123456",
		"lpa_code": "LPA:1$consumer.ppp.local$TN20210805541629D9E05F37",
		"deactivation_date": "",
		"allowed_usage_kb": 102400,
		"remaining_usage_kb": 51200,
		"remaining_days": 5,
		"status": "Activated",
		"bundle_id": 96,
		"bundle": "Aquila",
		"auto_refill_turned_on": false,
		"auto_refill_amount_mb": 0,
		"auto_refill_price": 0,
		"auto_refill_currency": "usd",
		"auto_refill_list": [
		{
		"title": "100MB/10DAYS",
		"amount_mb": 100,
		"amount_days": 10,
		"price_usd": 3,
		"price_eur": 3
		},
		{
		"title": "1GB/20DAYS",
		"amount_mb": 1024,
		"amount_days": 20,
		"price_usd": 7,
		"price_eur": 7
		}
		],
		"notes": "",
		"data_bundles": [
		{
		"id": 42,
		"status": "Active",
		"allowed_usage_kb": 102400,
		"active_kb": 51200,
		"remaining_usage_kb": 51200,
		"validity": 10,
		"assigned_at": "2022-08-11 11:57:27",
		"activated_at": "2022-08-11 12:07:00",
		"terminated_at": null,
		"expire_at": "2022-08-21 11:57:27"
		},
		{
		"id": 43,
		"status": "Non-active",
		"allowed_usage_kb": 102400,
		"active_kb": 0,
		"remaining_usage_kb": 102400,
		"validity": 10,
		"assigned_at": "2022-08-11 11:57:37",
		"activated_at": null,
		"terminated_at": null,
		"expire_at": null
		},

		{
		"id": 44,
		"status": "Finished",
		"allowed_usage_kb": 102400,
		"active_kb": 102400,
		"remaining_usage_kb": 0,
		"validity": 10,
		"assigned_at": "2021-01-01 10:00:00",
		"activated_at": "2021-01-01 10:05:00",
		"terminated_at": "2021-01-03 17:00:00",
		"expire_at": "2021-11-01 10:05:00"
		}
		]
		}
		}

GET/line/{iccid}/transactions

Use: This request receives a list of cash transactions by iccid, including the date of the transaction, refill amount, payment method
(balance or card), transaction ID, status, and the reason for the transaction.

Parameters

Name	Description
iccid * required string **(path)**	iccid

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"transactions": {
		"data": [
		{
		"created_at": "2021-09-24 09:15:49",
		"status": "Completed",
		"amount": "277.00",
		"currency": "EUR",
		"type": "ideal",
		"invoice_hash": "f391a7537acf5775989d20772cc01736",
		"refill_amount_mb": 51200,
		"reason": "Account Mr. Smith (Refill)",
		"transaction_id": "AB_49406_58"
		}
		]
		}
		}

GET/line/{iccid}/audit_logs

Use: This request receives an audit log and system actions for the specific Line, including action performed, prior value, current value,
and username of a person or entity acting.

Parameters

Name	Description
iccid * required integer **(path)**	iccid

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"audit_logs": {
		"data": [
		{
		"date": "2024-09-24 09:15:59",
		"field": "Status",
		"prior_value": "Deactivated",
		"value": "Activated",
		"user_name": "System",
		"description": ""
		}
		]
		}
		}

POST/line/create

Use: This request is used to create a Line. The request body is required to contain such information as:

- **refill_mb** – required
- Either **bundle_id** or **filters** – at least one is required
- **refill_days** – required only for Travel bundles
- **count** – optional (default: 1, max: 1000)

Parameters

- **refill_mb** (required): Specifies the size of the refill in megabytes. Must match one of the available refills for a bundle, as defined in the /bundles or /bundles/{id} endpoints.
- **refill_days:** Required only for Travel bundles. If provided with a Lifetime bundle, it will be ignored.
- **bundle_id:** Directly identifies the bundle. If provided, filters will be ignored.
- **filters:**
  Used when bundle_id is not provided. Filters include:
  - country_code (e.g., "US", "FR")
  - type ("travel" or "lifetime")
  - preference (optional: "fastest", "cheapest", "default")
- If no preference is specified:
  - The default bundle will be used if available.
  - If not, the fastest bundle will be selected.
- **count:**
  Optional. Number of lines to create. Default is 1. Max is 1000.

FAQ

Is the "refill_mb" a required parameter?
Yes, refill_mb is required. It defines the data amount of the selected refill and must match one of the available refill options in the selected bundle.

Where can the "refill_mb" parameter be found?
You can find valid refill_mb values by using the GET/bundles or GET/bundles/{id} endpoints. Each bundle includes a list of available refills.

Is the "count" a required parameter?
No, «count» is an optional parameter. It represents the number of lines to be created.
The parameter is set to 1 by default. The maximum value is 1000.

What is the "refill_mb" needed for?
The refill_mb parameter is required to determine which lpa_code (eSIM profile) to issue. When creating a line, you must include:

- refill_mb: the desired data amount
- bundle_id: to select a bundle directly, or
- filters: to find a suitable bundle based on country/type/preference

Without specifying refill_mb, it is not possible to issue a line.

What are the maximum lengths of "qr_code" and "lpa_code"?
Both qr_code and lpa_code can be up to 191 characters in length. Typically, the actual codes are shorter, but this is the maximum supported limit.

FAQ

How to connect the products/bundles I have in my account to the client management system we are building?

Our API is designed to help you activate and refill eSIMs and integrate available bundles from your account into your client management system. To achieve this, use:

- POST /line/create – to create a line
- POST /line/{iccid}/refill – to top up an existing line
- GET /bundles and GET /bundles/{id} – to browse available bundles and refills
- GET /lines – to retrieve a list of created lines

These endpoints support full integration with external systems or CMS platforms.

As a developer integrating purchased bundles and a payment gateway into a website, how can the API help?

Our API supports integration with both standard and custom workflows, allowing purchased bundles to be activated via POST /line/create. Integration steps typically include:

- Handling payment confirmation on your side
- Triggering POST /line/create based on purchase logic
- Using GET /lines and GET /lines/line/{iccid}/get_details to manage and display user-specific eSIMs

Even if your CMS does not support direct API requests, you can use middleware platforms (like Zapier) to bridge the connection. Crypto payment gateways can also be integrated, just trigger the API after confirming the payment on-chain or via webhook.

Request body examples

Travel bundle using bundle_id	{ "refill_mb": 1024, "refill_days": 7, "bundle_id": 2, "count": 3 }
Lifetime bundle using bundle_id	{ "refill_mb": 1024, "bundle_id": 20, "count": 3 }
Travel bundle with filters and default preference	{ "refill_mb": 1024, "refill_days": 30, "count": 1 "filters": { "country_code": "US", "type": "travel" } }
Travel bundle with filters and fastest preference	{ "refill_mb": 1024, "refill_days": 30, "count": 1 "filters": { "country_code": "US", "type": "travel" "preference": "fastest" } }
Travel bundle with filters and cheapest preference	{ "refill_mb": 1024, "refill_days": 30, "count": 1 "filters": { "country_code": "US", "type": "travel" "preference": "cheapest" } }
Lifetime bundle with filters and fastest preference	{ "refill_mb": 1024, "count": 1 "filters": { "country_code": "US", "type": "lifetime" "preference": "fastest" } }

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"sim_card": {
		"iccid": "8910300001003044041",
		"lpa_code": "LPA:1$consumer.ppp.local$TN20210805541629D9E05F37"
		},
		"data_bundle_id": 356
		}

POST/line/{iccid}/refill

Use: This request is used to refill the Line with a certain amount of data for a specific period (number of days).

FAQ

Do the 'amount_mb' and 'amount_days' parameters in the >/line/{iccid}/refill command need to match values retrieved from the /bundles command?
Yes, the 'amount_mb' and 'amount_days' parameters in the >/line/{iccid}/refill command must match the corresponding values obtained from the /bundles command. Ensuring consistency between these parameters is crucial for accurate and effective eSIM refills.

Parameters

Name	Description
iccid * required **string** **(path)**	iccid

Request body required
Amount in MB example
	{
	"amount_mb": 0,
	"amount_days": 0
	}

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"data_bundle_id": 367,
		"transaction_id": "AB_32321_42"
		}

PUT/line/{iccid}/update_notes

Use: This request is used to update the notes of the Line. The request body is the required parameter. Using notes, you can bring more order to how lines are displayed on your platform and easily find the required line based on the notes attached.

Parameters

Name	Description
iccid * required string **(path)**	iccid

Request body required
Notes example
	{
	"notes": "My ESIM #1"
	}

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success"
		}

PUT/line/{iccid}/deactivation_date

Use: This request is used to update the deactivation date of the Line. The deactivation_date is the date when the line will be automatically deactivated in the future. If deactivation_date is set as a past date, the line will be deactivated.

Parameters

Name	Description
iccid * required string **(path)**	iccid

Request body required
format Y-m-d example
	{
	"deactivation_date": "2022-12-25"
	}

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success"
		}

POST/line/{iccid}/turn_on_auto_refill

Use: This request is used to turn on Auto-Refill for the Line.

For Travel lines, the **'amount_mb'** and **'amount_days'** parameters must correspond to the number of megabytes and days from the available refills for your line (use /line/{iccid}/available_refills to select the appropriate refill option for Auto-Refill). There is no need to transfer the number of days for Lifetime lines.

The **'bill_protect_amount'** parameter contains the maximum amount that will be spent using the Auto-Refill.
The maximum spending limit will not be set if this parameter is not passed or the value is 0.

To set a new billing protection amount — send a request to enable the Auto-Refill with **'amount_mb'**, **'amount_days'** and a new billing protection amount. To reset the amount spent as part of the billing protection — send a command to turn off the Auto-Refill, and then a command to turn it on.

Parameters

Name	Description
iccid * required string (path)	iccid

Request body required
Amount in MB. Bill protection in USD
	{
	"amount_mb": 0,
	"amount_days": 0,
	"bill_protect_amount": 0
	}

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"message": "Auto-refill was turned ON"
		}

PUT/line/{iccid}/turn_off_auto_refill

Use: This request is used to turn off the Auto-Refill for the Line.

Parameters

Name	Description
iccid * required string **(path)**	iccid

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"message": "Auto-refill was turned OFF"
		}

08. Product Types Request

Product type is a property of a bundle. esimba.ai currently provides two product types — Lifetime and Travel — with different operation logics.
You can add lines on both Lifetime and Travel product types to one account. Each product type comes with a set of bundles. One Line can belong to one product type.

- **Lifetime product type** offers data bundles with no expiration. When you check the remaining data validity, it will always display '365,' but the data itself has no expiration.
- **Travel product type** offers short-term and long-term bundles, mostly used in the travel industry. These bundles have a limited number of megabytes and a validity period. They are activated on the first usage.

GET/product_types

Use: This request receives a list of all product types from the esimba.ai console. Currently, it includes Lifetime and Travel product types.

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"product_types": [
		"Lifetime",
		"Pay per megabyte",
		"Data expires in X days",
		"Unlimited with slow down",
		"Flex Unlimited"
		]
		}

09. Account balance

GET/account_balance

Use: This request returns the financial sum available within a partner's account in the platform. It represents the current amount of funds that a partner has at their disposal for conducting transactions.

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"account_balance": "8.63"
		}

401	Unauthenticated	
403	Forbidden	

10. Bundles Request

The bundles list is provided in response to the /line/{iccid}/get_details request.
Only those bundles that belong to a specific line may have the following statuses:

Name	Description
NON-ACTIVE	the bundle is not active until the customer makes the first usage of the bundle's data
ACTIVE	when the customer is using the bundle's data
FINISHED	if customer used all data on the bundle (before expiration)
EXPIRED	if days were expired (before used all data)
CONFIGURING	if the purchased data bundle is in the configuring process

GET/bundles

Use: This request reveals **a list of Global/Regional/Country data bundles** from the Console that you can use in your account. It reveals the bundle's names, descriptions, coverage, and available refills. Each bundle is defined by the locations where the line can be used, the data allowance, the period this data is valid, APN settings, data breakout, etc. You can have different bundles in one account.

Responses

Code	Description	Example
200	Successful operation	

GET/bundles/{id}

Use: This request reveals **a specific bundle** from esimba.ai console that you can use in your Partner Account. It reveals the bundle's name, description, coverage, and available refills.

FAQ

Can a bundle change after it's already created?
Once a bundle is created, coverage may change (countries and operators may be added or removed, the notification will be sent in advance) or refill prices. In this case, the existing bundle will be changed.

Parameters

Name	Description
id * required integer **(path)**	Bundle ID

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"bundle": [
		{
		"id": 1,
		"name": "Orion",
		"description": "Affordable data bundle for Europe and the USA",
		"coverage": [
		"Austria",
		"Belgium",
		"Bulgaria"
		],
		"refills": [
		{
		"title": "100MB",
		"amount_mb": 100,
		"amount_days": null,
		"price_usd": 3,
		"price_eur": 2.5
		}
		],
		"privacy_i": "Netherlands if usage is un Europe. Virginia if usage is in the USA",
		"networks": [
		{
		"title": "Austria",
		"local_networks": [
		"3",
		"A1",
		"T-Mobile"
		],
		"flag": "https://{your_link}.local/img/flags/1*1/at.svg"
		}
		]
		}

404	Not found	{
		"ack": "fail",
		"message": "Bundle not found"
		}

11. Transactions Request

Every line that belongs to the Partner Account comes with a list of cash transactions, offering the entire history of refills, including the refill amount, date and time, payment method, transaction ID, and status.

GET/transactions

Use: This request reveals a list of transactions referred to a specific Line belonging to your account and integrates them into your platform.

Parameters

Name	Description
sort_field string (query)	sort_field=/transactions?common_filter=02765888
sort_order string (query)	sort_order=/transactions?common_filter=00614433&sort_field=refill_amount_mb
common_filter string (query)	common_filter=asc or desc

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"transactions": {
		"data": [
		{
		"created_at": "2022-03-30 07:42:35",
		"status": "Completed",
		"amount": "6.50",
		"currency": "EUR",
		"type": "bancontact",
		"invoice_hash": "8a7a8bf8c1e55ffdb872a54afb197640",
		"refill_amount_mb": 1024,
		"reason": "Customer (Refill)",
		"transaction_id": "AB_49406_58"
		}
		]
		}
		}

12. Network Providers Request

Every bundle works on one or several network providers in the countries/regions where it provides coverage. The list of network providers is an attribute of a bundle. Contact your Personal Manager if you require a specific network that is not provided by default.

GET/network_providers

Use: The /network_providers request reveals a list of all operators we have in bundles.

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"network_providers": {
		"1970": "3 (Austria)",
		"2018": "3 (Indonesia)"
		}
		}

13. Countries / Regions Request

The lists of countries and regions are an attribute of a bundle. Every bundle provides coverage services that are only available in certain countries.
Using the /countries command, you get all countries issued in general, without reference to bundles.

GET/countries

Use: The request reveals a list of all countries where esimba.ai products have coverage.

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"countries": {
		"1": "Afghanistan",
		"2": "Albania",
		"3": "Algeria",
		"4": "American Samoa",
		"5": "...",
		"6": "..."
		}
		}

GET/regions

Use: The request reveals a list of all regions where esimba.ai products have coverage.

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"regions": {
		"1": "Africa",
		"2": "Asia",
		"3": "Europe",
		"4": "North America",
		"5": "Pacific",
		"6": "EMEA",
		"7": "Middle East"
		}
		}

14. eSIM devices Request

eSIMs don't work on devices that do not support the technology. To make it easier for your customers to determine if the eSIM is compatible with their devices, we provide a regularly updated list of eSIM devices where eSIMba bundles can be installed.

GET/esim_enabled_devices

Use: This request receives a list of eSIM-enabled devices compatible with esimba.ai eSIM bundles

Responses

Code	Description	Example

200	Successful operation	{
		"ack": "success",
		"data": [
		{
		"type": "Tablets & laptops",
		"brands": [
		{
		"title": "ASUS",
		"models": [
		"ASUS VivoBook Flip 14 TP401NA",
		"ASUS NovaGo TP370QL"
		]
		},
		{
		"title": "Acer",
		"models": [
		"Acer Spin 7",
		"Acer TravelMate P6",
		"Acer TravelMate Spin P4"
		]
		}
		]
		},
		{
		"type": "Smartphones",
		"brands": [
		{
		"title": "Apple",
		"models": [
		"iPhone 13 Pro",
		"iPhone 13 Pro Max",
		"iPhone 13 mini",
		"iPhone 13"
		]
		},
		{
		"title": "Google",
		"models": [
		"Google Pixel 3a",
		"Google Pixel 4"
		]
		}
		]
		}
		]
		}