

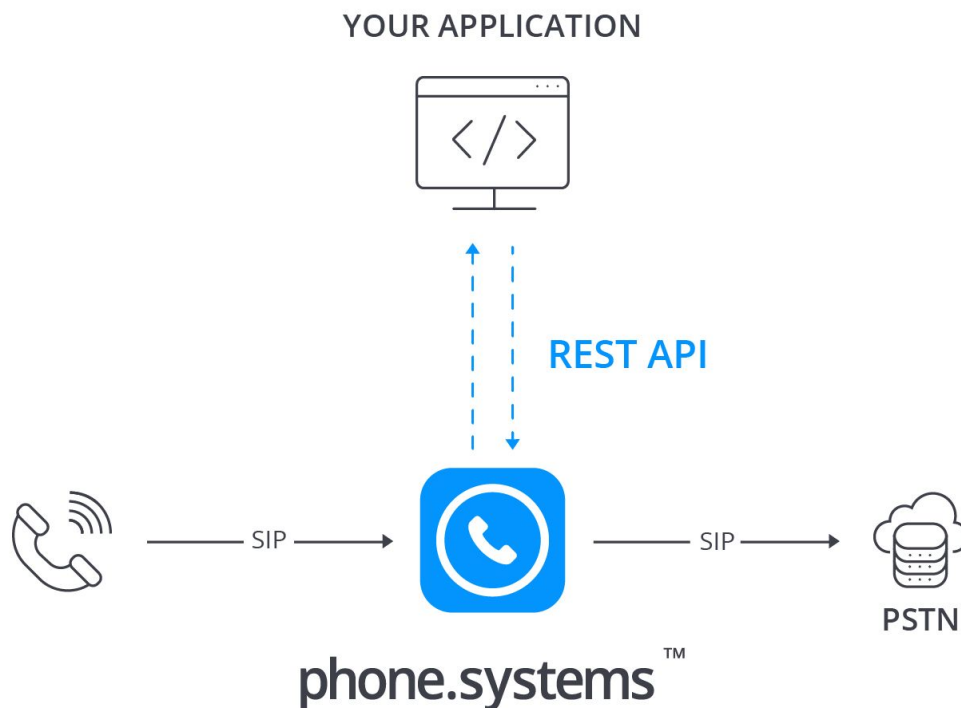
OPERATOR API

| | |
|----------------------------------|-----------|
| INTRODUCTION | 3 |
| GETTING STARTED | 3 |
| INITIAL REQUIREMENTS | 4 |
| ACCESSING THE API | 5 |
| ENVIRONMENTS | 5 |
| SUB-DOMAINS | 6 |
| SUB-DOMAIN CUSTOMIZATION [draft] | 6 |
| Co-branding | 6 |
| Termination | 6 |
| Incoming Trunks | 6 |
| Codecs | 6 |
| Mailing | 7 |
| Other | 7 |
| HEADERS | 8 |
| AUTHENTICATION | 8 |
| CONTENT NEGOTIATION | 8 |
| FETCHING DATA | 9 |
| INCLUSION OF RELATED RESOURCES | 9 |
| SPARSE FIELDSETS | 10 |
| SORTING | 11 |
| PAGINATION | 12 |
| FILTERING | 12 |
| REFERENCE | 13 |
| CUSTOMERS | 14 |
| Creating a Customer | 15 |
| Retrieving Customers | 17 |
| Updating a Customer | 19 |
| Deleting a Customer | 21 |
| CUSTOMER SESSIONS | 22 |
| Creating a Customer Session | 23 |
| ADMIN SESSIONS | 25 |
| Creating an Admin Session | 26 |
| INCOMING TRUNKS | 28 |

| | |
|--------------------------------|-----------|
| Creating an Incoming Trunk | 29 |
| Retrieving Incoming Trunks | 32 |
| Updating an Incoming Trunk | 34 |
| Deleting an Incoming Trunk | 36 |
| DID NUMBERS | 37 |
| Creating a DID Number | 38 |
| Retrieving DID Numbers | 40 |
| Updating a DID Number | 43 |
| Deleting a DID Number | 45 |
| GATEWAYS | 46 |
| Creating a Gateway | 47 |
| Retrieving Gateways | 50 |
| Updating a Gateway | 52 |
| Deleting a Gateway | 55 |
| ROUTES | 56 |
| Creating a Route | 57 |
| Retrieving Routes | 60 |
| Updating a Route | 62 |
| Deleting a Route | 64 |
| EXTERNAL ACCOUNTS | 65 |
| Creating an External Account | 66 |
| Retrieving External Accounts | 68 |
| Updating an External Account | 71 |
| Deleting an External Account | 72 |
| SIP ACCOUNTS | 73 |
| Retrieving SIP Accounts | 73 |
| Updating a SIP Account | 76 |
| TERMINATION CDRS | 78 |
| Retrieving Termination CDRs | 79 |
| INCOMING TRUNK CDRS | 83 |
| Retrieving Incoming Trunk CDRs | 83 |
| API OVERVIEW | 85 |

INTRODUCTION

Operator APIs have been built to open up the telecom.center™ software products for integration with service operators, MVNOs, resellers, enterprise customers and other types of telecommunication service providers.



The Operator API endpoints provide the ability to provision and combine third-party SIP resources and services with the cloud based PBX phone.systems™ and deliver full PBX solutions to customers via the operator's self-service portal.

GETTING STARTED

Each integration of telecom.center™ products is done on a custom sub-domain, which is created by the telecom.center™ staff in accordance with your requirements and depends on the services that your business provides. Before you begin, please review the [INITIAL REQUIREMENTS](#) and [DOMAIN CUSTOMISATION](#) sections in this documentation.

INITIAL REQUIREMENTS

phone.systems™ and call.center™ software products have been built as open platforms that may be combined with your preferred SIP trunks to enable call termination to PSTN as well as provision DID numbers to receive incoming traffic to the PBX.

It is expected that the operator is able to provide the following services that are needed for a **complete** PBX solution:

- DID numbers
- Termination gateways
- Customer billing

Exceptions are possible and may be discussed with a telecom.center™ sales representative via email at sales@telecom.center.

NOTE - phone.systems™ and call.center™ may be used in conjunction with any SIP-compatible infrastructure.

ACCESSING THE API

To get access credentials required to authenticate to the Operator API, you should contact Sales at sales@telecom.center.

You will be provided with:

- Basic authentication credentials (username/password)

Refer to the [AUTHENTICATION](#) section for more details on how to use the basic authentication credentials.

ENVIRONMENTS

- **Production**
URL <https://api.telecom.center/>
- **Sandbox**
URL <https://api.sandbox.telecom.center/>

Important - HTTP is not supported in order to ensure data privacy.

SUB-DOMAINS

For each phone.systems™ or call.center™ integration, operators are provided with a custom sub-domain where all configurations, customers and their resources will be stored. The sub-domain is preconfigured by the telecom.center™ staff according to the operator's requirements and preferences, and various UI and call routing settings are available.

Please discuss the available customization settings with telecom.center™ staff.

The settings applied to your sub-domain will be automatically applied for all newly created customers within that sub-domain.

HEADERS

AUTHENTICATION

All requests to the Operator API must be authenticated. This API uses basic authentication methods. You may find more information about the basic authentication [here](#).

Each API request should contain a header field in the form of:

```
header 'Authorization: Basic <credentials>'
```

where username and password are joined by a single colon ':' and are encoded in Base64.

Refer to the [ACCESSING THE API](#) section on how to receive your basic authentication credentials.

CONTENT NEGOTIATION

All requests to the Operator API should include the following header:

```
Content-Type: application/vnd.api+json
```

without any media type parameters.

Clients that include the JSON:API media type in their `Accept` header MUST specify the media type there at least once without any media type parameters.

Clients MUST ignore any parameters for the `application/vnd.api+json` media type received in the `Content-Type` header of response documents.

The server will respond with a 415 Unsupported Media Type status code if a request specifies the header `Content-Type: application/vnd.api+json` with any media type parameters.

The server will respond with a 406 Not Acceptable status code if a request's `Accept` header contains the JSON:API media type, and all instances of that media type are modified with media type parameters.

FETCHING DATA

The Operator API has been built according to JSON:API specifications and supports common features such as data sorting, filtering and sparse fieldsets. The following section contains general information taken from JSON:API specifications and is intended to be used as a quick reference on how certain Operator API features work and should be used.

If you are not familiar with JSON specifications, please review them [here](#) before building your first application using the Operator API.

INCLUSION OF RELATED RESOURCES

Some endpoints support an `include` request parameter to allow the client to customize which related resources should be returned.

If an endpoint does not support the `include` parameter, it will respond with 400 Bad Request to any requests that include that endpoint.

If an endpoint supports the `include` parameter and a client supplies it, the server will not include unrequested resource objects in the included section of the compound document.

NOTE - Some endpoints return resources related to the primary data by default.

The value of the included parameter must be a comma-separated (U+002C COMMA, “,”) list of relationship paths. Multiple related resources can be requested in the comma-separated list.

Below is an example of retrieving a list of routes, with gateway and customer relationships included:

GET

`/api/rest/public/operator/termination_routes?include=gateway,customer`

NOTE - This section applies to any endpoint that responds with primary data, regardless of the request type. For instance, an endpoint may support the inclusion of related resources along with a POST request to create a resource or relationship.

SPARSE FIELDSETS

A client may request that an endpoint return only specific fields in the response on a per-type basis, by including a `fields[TYPE]` parameter in the request URL.

The value of the 'fields' parameter must be a comma-separated (U+002C COMMA, ",") list that refers to the name(s) of the fields to be returned.

If a client requests a restricted set of fields for a given resource type, an endpoint will not include additional fields in the resource objects of that type in its response.

An example of retrieving a list of gateways specifying 'host' and 'name' attributes is shown below:

GET

`/api/rest/public/operator/termination_gateways?fields[termination_gateways]=host,name`

NOTE - In this example, the URI is shown in unencoded [and] characters for simpler readability. In practice, these characters must be percent-encoded, as per the requirements in RFC 3986.

NOTE - This section applies to any endpoint that responds with resources as primary or included data, regardless of the request type. For instance, a server may support sparse fieldsets along with a POST request to create a resource.

SORTING

Most Operator API endpoints support requests to sort the primary data with a `sort` query parameter. The value for 'sort' must represent sort fields.

An example of sorting customers by a capacity limit attribute is show below:

GET /api/rest/public/operator/customers?sort=capacity_limit

Multiple sort fields can be included by allowing comma-separated (U+002C COMMA, ",") sort fields. Sort fields should be applied in the order specified.

An example of sorting customers by the capacity limit and SIP account limit is show below:

**GET
/api/rest/public/operator/customers?sort=capacity_limit,sip_account_limit**

The sort order for each sort field will be ascending unless it is prefixed with a minus (U+002D HYPHEN-MINUS, "-"). In such a case, results will be sorted in descending order.

An example of sorting Customer by capacity limit in descending order is show below:

GET /api/rest/public/operator/customers?sort=-capacity_limit

This example will return customers with the highest capacity limit values first.

If the endpoint does not support sorting as specified in the query parameter `sort`, it will return 400 Bad Request.

The server may apply default sorting rules to top-level data if the request parameter sort is not specified.

PAGINATION

Some of the Operator API endpoints limit the number of resources returned in a response to a subset (“page”) of the whole set available. The API uses a page-based pagination strategy and supports the following query parameters:

- `page[number]`
- `page[size]`

Also, responses in some of the Operator API endpoints provide links to traverse a paginated data set (“pagination links”). Pagination links appear in the `links` object that corresponds to a collection.

The following keys are used for pagination links:

- `first`: the first page of data
- `last`: the last page of data
- `prev`: the previous page of data
- `next`: the next page of data

Keys will either be omitted or have a `null` value to indicate that a particular link is unavailable.

Concepts of order, as expressed in the naming of pagination links, remain consistent with JSON:API's [sorting rules](#).

NOTE - This section applies to any endpoint that responds with a resource collection as primary data, regardless of the request type.

FILTERING

The `filter` query parameter is reserved for filtering data. Clients should use this key for filtering operations. Each Operator API endpoint has a different set of supported filters.

An example of filtering Termination Gateways by host “hello.com” is shown below:

GET

`/api/rest/public/operator/termination_gateways?filter[host]=hello.com`

REFERENCE

The following category explains all available Operator API endpoints with request/response examples for each HTTP method.

Please note that all endpoints can be tested using the Operator UI that has been built on top of the Operator API.

The Operator UI can be accessed at:

<https://console.phone.systems/>

<https://console.sandbox.phone.systems/>

Your Basic Auth credentials may be used to login to the Operator UI. This tool can be used as a reference on how each endpoint may be implemented, or to better understand the system's resource structure and relations.

CUSTOMERS

This endpoint provides the ability to manage customers and their available limitations on your domain. You can create a customer for each registered user account on your hosted service and delete this entry when such an account is closed.

Each customer contains their own set of resources and configuration that is isolated and independent from other customers.

The following HTTP method(s) are available:

- GET /customers - For [Retrieving Customers](#)
- POST /customers - For [Creating a new Customer](#)
- DELETE /customers/{id} - For [Deleting a Customer](#)
- PATCH /customers/{id} - For [Updating a Customer](#)

Creating a Customer

Use the HTTP POST method to create a new customer.

POST /api/rest/public/operator/customers

This method supports [Sparse Fieldsets](#)

Attributes:

| Attribute | Value | Comment | Optional | Controlled via UI |
|-------------------|---------------------------------|---|--------------------------------------|-------------------|
| name | Any string | Customer name | No | No |
| language | "EN", "RU", "LT", "LV" | Customer phone.systems™ User Interface language (English, Russian, Lithuanian or Latvian) | Yes, default is EN | Yes |
| capacity_limit | Any integer | Capacity limitation | Yes, default is no limitation (null) | No |
| sip_account_limit | Any integer | SIP Account limitation | Yes, default is no limitation (null) | No |

Request body:

```
{
  "data": {
    "id": "123",
    "type": "customers",
    "attributes": {
      "name": "John Johnson",
      "language": "EN",
      "incoming_capacity": 10,
      "sip_account_limit": 5
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "123",
    "type": "customers",
    "links": {
      "self":
"https://api.telecom.center/api/rest/public/operator/customers/2"
    },
    "attributes": {
      "name": "John Johnson",
      "language": "EN",
      "incoming_capacity": 10,
      "sip_account_limit": 5
    }
  }
}
```

Retrieving Customers

Use the HTTP GET method to retrieve a list of customers:

GET /api/rest/public/operator/customers

This method supports [Sorting](#), [Pagination](#) and [Sparse Fieldsets](#)

Other supported parameters:

| Parameter | Value | Description |
|------------|---------|---|
| filter[id] | integer | Used for filtering results by Customer ID |

Request Body:

```
-
```

Response Body:

```
{
  "data": [
    {
      "id": "123",
      "type": "customers",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/customers/1"
      },
      "attributes": {
        "name": "John Johnson",
        "language": "EN",
        "capacity_limit": 10,
        "sip_account_limit": 5
      }
    },
    {
      "id": "124",
      "type": "customers",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/customers/34"
      },
```



```
    "attributes": {
      "name": "Smith Johnson",
      "language": "EN",
      "capacity_limit": null,
      "sip_account_limit": null
    }
  ],
  "meta": {
    "record_count": 2
  },
  "links": {
    "first":
    "https://api.telecom.center/api/rest/public/operator/customers?page%5Bnumbe
r%5D=1&page%5Bsize%5D=50",
    "last":
    "https://api.telecom.center/api/rest/public/operator/customers?page%5Bnumbe
r%5D=1&page%5Bsize%5D=50"
  }
}
```

The response body contains an array of all customers.

Additionally, a Customer ID may be included in the request URL to retrieve a specific Customer as shown below:

GET /api/rest/public/operator/customers/{id}

The response body will contain a single customer if the ID matches.

Updating a Customer

Use the HTTP PATCH method to update an existing customer:

PATCH /api/rest/public/operator/customers/{id}

This method allows you to update a customer's available limitations when needed, or when any specific event has occurred on your hosted service that will have an impact on a customer's services in phone.systems™.

Attributes:

| Attribute | Value | Comment | Optional | Controlled via UI |
|-------------------|-------------|------------------------|----------|-------------------|
| capacity_limit | Any integer | Capacity limitation | Yes | No |
| sip_account_limit | Any integer | SIP Account limitation | Yes | No |

Other supported parameters:

| Parameter | Value | Description |
|-------------------|--------------------------|--|
| fields[customers] | Any Customer's attribute | Used for selecting specific attributes that will be returned |

Request body:

```
{
  "data": {
    "id": "123",
    "type": "customers",
    "attributes": {
      "incoming_capacity": 3,
      "sip_account_limit": 8
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "123",
    "type": "customers",
    "attributes": {
      "name": "John Johnson",
      "language": "string",
      "incoming_capacity": 3,
      "sip_account_limit": 8
    }
  }
}
```

Deleting a Customer

Use the HTTP DELETE method to delete a customer.

DELETE /api/rest/public/operator/customers/{id}

This method will delete the customer, its related resources and call routing configuration. IMPORTANT - this is a permanent action, and there is no ability to restore the customer data. The customer will no longer be able to access the UI. The deleted Customer ID may be reused to create a new customer.

NOTE - Deleted customers will no longer appear when retrieving a list of customers.

Request body:

-

Response body:

-

The HTTP status code 204 will be the response after a successful deletion of a customer with an empty response body.

CUSTOMER SESSIONS

This endpoint is used to provide access for the customer (end-user) to the phone.systems™ User Interface. The endpoint responds with a URL and token that should be used when redirecting and authenticating the customer to the UI on your custom domain.

A single customer may have only one active session at the same time. A newly generated session will cancel the previous one.

The following HTTP method is available:

- POST /sessions - For [creating a customer session](#) (token) to access the UI

Creating a Customer Session

Use the HTTP POST method to create a new customer session.

POST /api/rest/public/operator/sessions

Request body:

```
{
  "data": {
    "type": "sessions",
    "relationships": {
      "customer": {
        "data": {
          "id": "123",
          "type": "customers"
        }
      }
    }
  }
}
```

Response Body:

```
{
  "data": {
    "id": "123",
    "type": "sessions",
    "attributes": {
      "uri":
        "https://telecom.center/api/rest/public/operator/auth?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQ5MzQsImV4cCI6MTU4OTUyNDcxOH0.bjnTSBCgUJUAsUsBK_p09GJ08y476TdeIfqoknbHJuc"
    }
  }
}
```

Redirect the customer to a "uri" that is included in the response body. Afterwards, the user will be automatically redirected to a URL that contains your custom sub-domain name.

URL example:

<https://custom.phone.systems?token=YiJwhkmdnMSwqGCHxaUtRXLGdkyqPDpf>

As a result, the customer is automatically logged in to the phone.systems™ User Interface and can make adjustments to their PBX account.

ADMIN SESSIONS

This endpoint is used to access the Customers User Interface as an Administrator (staff). The created admin session will not interrupt active customer sessions.

NOTE - In case when both the Customer and Admin are accessing the UI at the same time, some changes made by one of the parties can not be automatically reflected for the other party. It is therefore recommended not to initiate both sessions simultaneously in order to avoid an unsatisfactory user experience for the customer.

The following HTTP method is available:

- POST /customer_admin_sessions - For [creating a new admin session](#) to access the UI

Creating an Admin Session

Use the HTTP POST method to create a new admin session:

POST /customer_admin_sessions

Request Body:

```
{
  "data": {
    "type": "customer_admin_sessions",
    "relationships": {
      "customer": {
        "data": {
          "id": "999",
          "type": "customers"
        }
      }
    }
  }
}
```

Response Body:

```
{
  "data": {
    "id": "999",
    "type": "customer_admin_sessions",
    "links": {
      "self":
      "https://api.telecom.center/api/rest/public/operator/customer_admin_sessions/999"
    },
    "attributes": {
      "uri":
      "https://phone.systems/ui/index.html?x-admin-token=trZdERpmNSJWTTrnpOzeXDqTaWwDRJSnp"
    }
  }
}
```

The **uri** attribute in the response body contains a full URL with an included token for redirecting to the UI.

Example URL:

<https://phone.systems/ui/index.html?x-admin-token=trZdERpmNSJWTTrnpOzeXDqTaWwDRJSnp>

Redirect the admin to the provided URL. As a result, the admin is automatically logged in to the phone.systems™ User Interface and can make modifications to the customer's PBX account.

INCOMING TRUNKS

This endpoint provides the ability to manage Incoming Trunks (Inbound Trunks in UI) and their settings in customer accounts.

Incoming trunks are used to receive incoming traffic from DID numbers. A single customer may include an unlimited amount of incoming trunks, however, a single trunk for each customer account is sufficient in most cases.

Each incoming trunk has a unique domain that is generated automatically during the creation process.

NOTE - Your sub-domain may be configured to create an incoming trunk by default for every newly created customer. See the [SUB-DOMAIN CUSTOMIZATION](#) section regarding supported sub-domain settings.

All calls from the DID provider to the PBX are reflected in the Incoming Trunk CDRs.

The following HTTP method(s) are available:

- GET /incoming_trunks - For [Retrieving Incoming Trunks](#)
- POST /incoming_trunks - For [Creating an Incoming Trunk](#)
- DELETE /incoming_trunks/{id} - For [Deleting an Incoming Trunk](#)
- PATCH /incoming_trunks/{id} - For [Updating an Incoming Trunk](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------|---------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer's ID that will include this incoming trunk |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Creating an Incoming Trunk

Use the HTTP POST method to create a new Incoming Trunk:

POST /api/rest/public/operator/incoming_trunks

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|--------------------|--|---|-----------------------------------|-------------------|
| name | Any string | Trunk name. May be used as a secondary identifier or displayed in the UI. | Yes, default is null | Yes |
| transport_protocol | "UDP", "TCP" | Preferred transport protocol for this incoming trunk | Yes, default is UDP | Yes |
| codecs | "g729", "G722", "GSM", "PCMA", "PCMU", "OPUS", "telephone-event" | Allowed codecs for this incoming trunk | Yes, defaults may vary per domain | Yes |
| destination_field | "RURI_USERPART", "TO_USERPART" | Incoming Trunk DID number lookup mode | Yes, default is RURI_USERPART | Yes |

Request Body:

```
{
  "data": {
    "type": "incoming_trunks",
    "attributes": {
      "name": "DIDWW",
      "transport_protocol": "UDP",
      "destination_field": "RURI_USERPART",
      "codecs": [
        "PCMU"
      ]
    },
    "relationships": {
      "customer": {
        "data": {
          "id": "123",
          "type": "customers"
        }
      }
    }
  }
}
```

Response Body:

```
{
  "data": {
    "id": "9edf20a5-48d1-485e-895c-dc35a466d614",
    "type": "incoming_trunks",
    "links": {
      "self":
        "https://api.telecom.center/api/rest/public/operator/incoming_trunks/9edf20a5-48d1-485e-895c-dc35a466d614"
    },
    "attributes": {
      "name": "DIDWW",
      "codecs": [
        "PCMU"
      ],
      "transport_protocol": "UDP",
      "domain": "3391031169570.in.phone.systems",
      "destination_field": "RURI_USERPART"
    },
    "relationships": {
      "customer": {
        "links": {
          "self":
            "https://api.telecom.center/api/rest/public/operator/incoming_trunks/9edf20a5-48d1-485e-895c-dc35a466d614/relationships/customer",
          "related":
            "https://api.telecom.center/api/rest/public/operator/incoming_trunks/9edf20a5-48d1-485e-895c-dc35a466d614/customer"
        }
      }
    }
  }
}
```

The response body contains a “domain” attribute that should be used as the Host part of the SIP URI when forwarding calls to this trunk.

Retrieving Incoming Trunks

Use the HTTP GET method to retrieve a list of Incoming Trunks:

GET /api/rest/public/operator/incoming_trunks

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|---------------------|----------------------------------|---|
| filter[customer.id] | Customer ID | Used for filtering Incoming Trunks by Customer ID |
| filter[domain] | The unique incoming trunk domain | Used for filtering Incoming Trunks by Domain |
| filter[id] | Incoming Trunk ID | Used for filtering results by Incoming Trunk ID |

Request Body:

-

Response Body:

```
{
  "data": [
    {
      "id": "25d2ea46-f684-4e72-fake-5fbd26983cb0",
      "type": "incoming_trunks",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/incoming_trunks/25d2ea
46-f684-4e72-b397-5fbd26983cb0"
      },
      "attributes": {
        "name": "Default",
        "codecs": [
          "PCMU",
          "PCMA",
          "telephone-event/8000"
        ],
        "transport_protocol": "UDP",
        "domain": "3391161930393.in.phone.systems",
        "destination_field": "RURI_USERPART"
      }
    }
  ],
  "meta": {
    "record_count": 1
  },
  "links": {
    "first":
"https://api.telecom.center/api/rest/public/operator/incoming_trunks?page%5
Bnumber%5D=1&page%5Bsize%5D=50",
    "last":
"https://api.telecom.center/api/rest/public/operator/incoming_trunks?page%5
Bnumber%5D=1&page%5Bsize%5D=50"
  }
}
```

The response body contains an array of all Incoming Trunks.

Additionally, an Incoming Trunk ID may be included to the request URL in order to retrieve a single Incoming Trunk:

GET /api/rest/public/operator/incoming_trunks/{id}

Updating an Incoming Trunk

Use the HTTP PATCH method to update an Incoming Trunk:

PATCH /api/rest/public/operator/incoming_trunks/{id}

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|--------------------|---|--|-----------------------------------|-------------------|
| name | Any string | Trunk name that is displayed in UI | Yes, default is null | Yes |
| transport_protocol | "UDP", "TCP" | Preferred transport protocol for this incoming trunk | Yes, default is UDP | Yes |
| codecs | "g729", "G722", "GSM", "PCMA", "PCMU", "OPUS" | Allowed codecs for this incoming trunk | Yes, defaults may vary per domain | Yes |
| destination_field | "RURI_USERPART", "TO_USERPART" | Incoming Trunk DID number lookup mode | Yes, default is RURI_USERPART | Yes |

Request body:

```
{
  "data": {
    "id": "25d2ea46-f684-4e72-b397-5fbd26983cb0",
    "type": "incoming_trunks",
    "attributes": {
      "name": "Updated name",
      "transport_protocol": "TCP",
      "destination_field": "TO_USERPART",
      "codecs": [
        "PCMA"
      ]
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "25d2ea46-f684-4e72-b397-5fbd26983cb0",
    "type": "incoming_trunks",
    "links": {
      "self":
      "https://api.telecom.center/api/rest/public/operator/incoming_trunks/25d2ea46-f684-4e72-b397-5fbd26983cb0"
    },
    "attributes": {
      "name": "Updated name",
      "codecs": [
        "PCMU",
        "PCMA",
        "telephone-event/8000"
      ],
      "transport_protocol": "TCP",
      "domain": "3391161930393.in.phone.systems",
      "destination_field": "TO_USERPART"
    }
  }
}
```

Deleting an Incoming Trunk

Use the HTTP DELETE method to delete an Incoming Trunk:

DELETE /api/rest/public/operator/incoming_trunks/{id}

This method will delete the Incoming Trunk with no ability to recover this trunk.

NOTE - The unique domain that was assigned to the deleted trunk will also be lost. New incoming trunks will not reuse the deleted unique incoming trunk domains.

Request body:

-

Response body:

-

The HTTP status code 204 will be the response after a successful deletion of an Incoming Trunk with an empty response body.

DID NUMBERS

This endpoint provides the ability to manage DID numbers (referred to as Phone Numbers in the UI) in your customer accounts. Using this endpoint, you can automate the process of provisioning DID numbers from your DID number pool and allocating them to phone.systems™ customer accounts.

NOTE - DID numbers are unique per Customer. The same number cannot be provisioned twice to the same Customer or to different customers.

The provisioned DIDs may be disabled when the number has to be temporarily suspended. Disabled numbers remain in the customer accounts and are visible to the end-user. The UI displays the disabled numbers in red. Also, the system rejects inbound calls made to the disabled numbers.

NOTE - Your sub-domain may be pre-configured to disallow Customers to add new DID numbers manually. See the [SUB-DOMAIN CUSTOMIZATION](#) section regarding supported sub-domain settings.

The following HTTP method(s) are available:

- GET /available_did_numbers - For [Retrieving DID Numbers](#)
- POST /available_did_numbers - For [Creating a DID Number](#)
- DELETE /available_did_numbers/{id} - For [Deleting a DID Number](#)
- PATCH /available_did_numbers/{id} - For [Updating a DID Number](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------|---------------|-------------------|-----------|---|
| customer | customers | To-one | No | Parent customer that this available DID number belongs to |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Creating a DID Number

Use the HTTP POST method to create a new available DID number:

POST /api/rest/public/operator/available_did_numbers

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|-----------|--------------------|--|------------------------|--------------------------------|
| number | Any integer | The number itself. Used for call routing | No | Depends on SUB-DOMAIN settings |
| enabled | "true", "false" | DID number status | Yes, default is "true" | No |

Request body:

```
{
  "data": {
    "type": "available_did_numbers",
    "attributes": {
      "number": "12120000001"
    },
    "relationships": {
      "customer": {
        "data": {
          "id": "123",
          "type": "customers"
        }
      }
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "8c58d9cb-54af-462d-be40-1cca246a5ffd",
    "type": "available_did_numbers",
    "attributes": {
      "number": "12120000001",
      "enabled": true
    },
    "relationships": {
      "customer": {
        "data": {
          "id": "123",
          "type": "customers"
        }
      }
    }
  }
}
```

NOTE - Added DID numbers are not yet ready to receive calls. Customers must add these numbers to the call flow configuration in the UI before they form part of the PBX logic

Retrieving DID Numbers

Use the HTTP GET method to retrieve a list of existing available DID numbers:

GET /api/rest/public/operator/available_did_numbers

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|---------------------|-------------------------|---|
| filter[id] | Available DID Number ID | Used for filtering results by resource ID |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[number] | A valid DID number | Used for filtering results by number |
| filter[enabled] | true, false | Used for filtering results by status |

Request body:

-

Response body:

```
{
  "data": [
    {
      "id": "55580532-d88b-46d9-ab9d-646373af90ad",
      "type": "available_did_numbers",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/available\_did\_numbers/55580532-d88b-46d9-ab9d-646373af90ad"
      },
      "attributes": {
        "number": "121200001",
        "enabled": true
      }
    },
    {
      "id": "5efdb730-703e-4f86-bc4e-e9b724154fc7",
      "type": "available_did_numbers",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/available\_did\_numbers/5efdb730-703e-4f86-bc4e-e9b724154fc7"
      },
      "attributes": {
        "number": "121200002",
        "enabled": true
      }
    }
  ],
  "meta": {
    "record_count": 2
  },
  "links": {
    "first":
"https://api.telecom.center/api/rest/public/operator/available\_did\_numbers?page%5Bnumber%5D=1&page%5Bsize%5D=50",
    "last":
"https://api.telecom.center/api/rest/public/operator/available\_did\_numbers?page%5Bnumber%5D=1&page%5Bsize%5D=50"
  }
}
```


The response body contains an array of all available DID numbers.

Additionally, an available DID number ID may be included in the request URL to retrieve a single available DID number:

GET /api/rest/public/operator/available_did_numbers/{id}

Updating a DID Number

Use the HTTP PATCH method to update an existing available DID number:

PATCH /api/rest/public/operator/available_did_numbers/{id}

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|-----------|--------------------|---|------------------------|--------------------------------|
| number | Any integer | The number itself. Used for call routing. | No | Depends on SUB-DOMAIN settings |
| enabled | "true", "false" | DID number status | Yes, default is "true" | No |

Other supported parameters:

| Parameter | Value | Description |
|-----------|------------|--|
| include | "customer" | Used for including linked object details to the response |

Request body:

```
{
  "data": {
    "id": "8c58d9cb-54af-462d-be40-1cca246a5ffd",
    "type": "available_did_numbers",
    "attributes": {
      "number": "12129999999",
      "enabled": false
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "8c58d9cb-54af-462d-be40-1cca246a5ffd",
    "type": "available_did_numbers",
    "attributes": {
      "number": "12129999999",
      "enabled": false
    },
    "relationships": {
      "customer": {
        "data": {
          "id": "123",
          "type": "customers"
        }
      }
    }
  }
}
```

Deleting a DID Number

Use the HTTP DELETE method to delete an available DID number from a customer's account:

DELETE /api/rest/public/operator/available_did_numbers/{id}

NOTE - the linked object (if such object exists) in the call flow configuration will also be deleted.

Request body:

-

Response body:

-

The HTTP status code 204 will be the response after a successful deletion of an available DID number with an empty response body.

GATEWAYS

This endpoint provides the ability to manage gateways on your Domain and Customers.

Gateways are used to enable outbound calls and call forwarding to PSTN on customer accounts. A single customer may include several gateways, however, a single gateway per customer account is sufficient in most cases.

NOTE - Your sub-domain may be pre-configured with a single gateway that will be used by default on all customer accounts. See the [SUB-DOMAIN CUSTOMIZATION](#) section regarding supported sub-domain settings.

All calls from the PBX to gateways are reflected in Termination CDRs.

The following HTTP method(s) are available:

- GET /termination_gateways - For [Retrieving Gateways](#)
- POST /termination_gateways - For [Creating a Gateway](#)
- DELETE /termination_gateways/{id} - For [Deleting a Gateway](#)
- PATCH /termination_gateways/{id} - For [Updating a Gateway](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------|---------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer's ID that will include this incoming trunk |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Creating a Gateway

Use the HTTP POST method to create a new gateway:

POST /api/rest/public/operator/termination_gateways

This method supports [Sparse Fieldsets](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|------------------------|---|---|--|-------------------------------|
| codecs | "g729", "G722", "GSM", "PCMA", "PCMU", "OPUS", "telephone-event/8000" | Allowed codecs for the gateway | Yes, defaults depend on SUB-DOMAIN configuration | Yes, if allowed on SUB-DOMAIN |
| port | integer | Port used on this gateway | Yes, default is 5060 | Yes, if allowed on SUB-DOMAIN |
| name | string | Gateway's name | No | Yes, if allowed on SUB-DOMAIN |
| host | string | Gateway's host address | No | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_rule | string | Regex-based rewrite rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_result | string | Regex-based result rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_rule | string | Regex-based rewrite rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_result | string | Regex-based result rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| authorization_name | string | Authorization username | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| authorization_password | string | Authorization password | Yes, default is null | Yes, if allowed on SUB-DOMAIN |

| | | | | |
|----------------------|---|--|-----------------------------------|----------------------------------|
| network_p rotocol | "IPv4_only", "IPv6_only", "IPv4_pref", "IPv6_pref", "Dualstack" | Network protocol used on this gateway | Yes, default is "IPv4_only" | Yes, if allowed on SUB-DOMAIN |
|----------------------|---|--|-----------------------------------|----------------------------------|

To-one relationship(s):

| Type | Description |
|-----------|--|
| customers | Parent customer's ID that will include this Gateway |

Other supported parameters:

| Parameter | Value | Description |
|-----------|------------|---|
| include | "customer" | Used for including linked object details to the response |

Request body:

```
{
  "data": {
    "type": "termination_gateways",
    "attributes": {
      "name": "Test Gateway",
      "host": "testing.api.com"
    },
  },
  "relationships": {
    "customer": {
      "data": {
        "id": "1",
        "type": "customers"
      }
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "0c18063b-ef04-4072-b19b-06484708e84a",
    "type": "termination_gateways",
    "links": {
      "self":
        "https://api.telecom.center/api/rest/public/operator/termination_gateways/0c18063b-ef04-4072-b19b-06484708e84a"
    },
    "attributes": {
      "name": "Test Gateway",
      "host": "testing.api.com",
      "port": null,
      "src_rewrite_rule": null,
      "src_rewrite_result": null,
      "dst_rewrite_rule": null,
      "dst_rewrite_result": null,
      "authorization_name": null,
      "authorization_password": null,
      "codecs": [
        "PCMU",
        "PCMA",
        "telephone-event/8000"
      ],
      "network_protocol": "IPv4_only"
    }
  }
}
```

NOTE - In order for a gateway to receive calls from the PBX, an appropriate route pointing to this gateway should be created. Refer to the [ROUTES](#) section for more details on how to create a route.

Retrieving Gateways

Use the HTTP GET method to retrieve gateways:

GET /api/rest/public/operator/termination_gateways

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|---------------------|----------------------------|---|
| filter[id] | Termination gateway ID | Used for filtering results by resource ID |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[name] | Termination gateway's name | Used for filtering results by name |
| filter[host] | Termination gateway's host | Used for filtering results by host |

Request body:

-

Response body:

```
{
  "data": [
    {
      "id": "b4f04436-b378-403f-a6c5-8040ae1fb824",
      "type": "termination_gateways",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/termination_gateways/b4f04436-b378-403f-a6c5-8040ae1fb824"
      },
      "attributes": {
        "name": "Test GW",
        "host": "sup.com",
        "port": null,
        "src_rewrite_rule": null,

```

```

    "src_rewrite_result": null,
    "dst_rewrite_rule": null,
    "dst_rewrite_result": null,
    "authorization_name": null,
    "authorization_password": null,
    "codecs": [
      "PCMU"
    ],
    "network_protocol": "IPv4_only"
  }
},
"meta": {
  "record_count": 1
},
"links": {
  "first":
  "https://api.telecom.center/api/rest/public/operator/termination_gateways?page%5Bnumber%5D=1&page%5Bsize%5D=50",
  "last":
  "https://api.telecom.center/api/rest/public/operator/termination_gateways?page%5Bnumber%5D=1&page%5Bsize%5D=50"
}
}

```

The response body will contain an array of all gateways.

Additionally, a Customer ID may be included in the request URL in order to retrieve a specific customer:

GET /api/rest/public/operator/termination_gateways{id}

Updating a Gateway

Use the HTTP PATCH method to retrieve gateways:

PATCH /api/rest/public/operator/termination_gateways/{id}

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|------------------------|---|---|--|-------------------------------|
| codecs | "g729", "G722", "GSM", "PCMA", "PCMU", "OPUS", "telephone-event/8000" | Allowed codecs for the gateway | Yes, defaults depend on SUB-DOMAIN configuration | Yes, if allowed on SUB-DOMAIN |
| port | integer | Port used on this gateway | Yes, default is 5060 | Yes, if allowed on SUB-DOMAIN |
| name | string | Gateway's name | No | Yes, if allowed on SUB-DOMAIN |
| host | string | Gateway's host address | No | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_rule | string | Regex-based rewrite rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_result | string | Regex-based result rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_rule | string | Regex-based rewrite rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_result | string | Regex-based result rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| authorization_name | string | Authorization username | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| authorization_password | string | Authorization password | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| network_protocol | "IPv4_only", "IPv6_only", "IPv4_pref", | Network protocol used on this gateway | Yes, default is "IPv4_only" | Yes, if allowed on SUB-DOMAIN |

| | | | | |
|--|-----------------------------|--|--|--|
| | "IPv6_pref", "Dualstack" | | | |
|--|-----------------------------|--|--|--|

Request body:

```
{
  "data": {
    "id": "0c18063b-ef04-4072-b19b-06484708e84a",
    "type": "termination_gateways",
    "attributes": {
      "port": 5060,
      "codecs": [
        "PCMU"
      ],
      "name": "GatewayTest",
      "host": "testing.com"
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "0c18063b-ef04-4072-b19b-06484708e84a",
    "type": "termination_gateways",
    "links": {
      "self":
        "https://api.telecom.center/api/rest/public/operator/termination_gateways/0c18063b-ef04-4072-b19b-06484708e84a"
    },
    "attributes": {
      "name": "GatewayTest",
      "host": "testing.com",
      "port": 5060,
      "src_rewrite_rule": null,
      "src_rewrite_result": null,
      "dst_rewrite_rule": null,
      "dst_rewrite_result": null,
      "authorization_name": null,
      "authorization_password": null,
      "codecs": [
        "PCMU"
      ],
      "network_protocol": "IPv4_only"
    }
  }
}
```

Deleting a Gateway

Use the HTTP DELETE method to delete a gateway:

DELETE /api/rest/public/operator/termination_gateways/{id}

This method will delete the gateway and all [Routes](#) that are linked to this deleted gateway.

Request body:

-

Response body:

-

The HTTP status code 204 will be the response after a successful deletion of a gateway with an empty response body.

ROUTES

This endpoint provides the ability to manage Routes on your Domain and Customers.

Routes are used to point call traffic to a specific Termination Gateway. They support RegEx rewrite and simple SRC/DST number rules that can be configured during the creation process or added later if your Customers or Domain uses multiple gateways.

NOTE - Your sub-domain may be pre-configured with a termination route that will be used by default on all customer accounts. See the [SUB-DOMAIN CUSTOMIZATION](#) section regarding supported sub-domain settings.

The following HTTP method(s) are available:

- GET /termination_routes - For [Retrieving Routes](#)
- POST /termination_routes - For [Creating a Route](#)
- DELETE /termination_routes/{id} - For [Deleting a Route](#)
- PATCH /termination_routes/{id} - For [Updating a Route](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------|---------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer's ID that will include this incoming trunk |
| gateway | gateways | To-one | Yes | Gateway that this route will point the call traffic to |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Creating a Route

Use the HTTP POST method to create a new Route

POST /api/rest/public/operator/termination_routes

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|--------------------|---------|--|----------------------|-------------------------------|
| name | string | Route name | No | Yes, if allowed on SUB-DOMAIN |
| src_prefix | integer | Source prefix that will take this Route | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_prefix | integer | Destination prefix that will take this Route | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_rule | string | Regex-based rewrite rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_result | string | Regex-based result rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_rule | string | Regex-based rewrite rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_result | string | Regex-based result rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |

Request body:

```
{
  "data": {
    "type": "termination_routes",
    "attributes": {
      "name": "Test Route"
    },
    "relationships": {
      "customer": {
        "data": {
          "id": "1",
          "type": "customers"
        }
      },
      "gateway": {
        "data": {
          "id": "b4f04436-b378-403f-a6c5-8040ae1fb824",
          "type": "termination_gateways"
        }
      }
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "a29e4de3-63d0-47ff-b098-8aaa9b4e34e1",
    "type": "termination_routes",
    "links": {
      "self":
"https://api.telecom.center/api/rest/public/operator/termination\_routes/a29e4de3-63d0-47ff-b098-8aaa9b4e34e1"
    },
    "attributes": {
      "name": "Test Route",
      "src_prefix": "",
      "dst_prefix": "",
      "src_rewrite_rule": null,
      "src_rewrite_result": null,
      "dst_rewrite_rule": null,
      "dst_rewrite_result": null
    }
  }
}
```

Retrieving Routes

Use the HTTP GET method to retrieve a list of Routes

GET /api/rest/public/operator/termination_routes

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|---------------------|--------------------------|---|
| filter[id] | Route ID | Used for filtering results by resource ID |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[name] | Termination route's name | Used for filtering results by name |

Request body:

-

Response body:

```
{
  "data": [
    {
      "id": "a29e4de3-63d0-47ff-b098-8aaa9b4e34e1",
      "type": "termination_routes",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/termination_routes/a29e4de3-63d0-47ff-b098-8aaa9b4e34e1"
      },
      "attributes": {
        "name": "Test Route",
        "src_prefix": "",
        "dst_prefix": "",
        "src_rewrite_rule": null,
        "src_rewrite_result": null,
        "dst_rewrite_rule": null,
        "dst_rewrite_result": null
      }
    }
  ],
  "meta": {
    "record_count": 1
  },
  "links": {
    "first":
"https://api.telecom.center/api/rest/public/operator/termination_routes?page%5Bnumber%5D=1&page%5Bsize%5D=50",
    "last":
"https://api.telecom.center/api/rest/public/operator/termination_routes?page%5Bnumber%5D=1&page%5Bsize%5D=50"
  }
}
```

The response body will contain an array of all Routes.

Additionally, a Route ID may be included in the request URL to retrieve the details of a specific Route:

GET /api/rest/public/operator/termination_routes/{id}

Updating a Route

Use the HTTP PATCH method to update a Route

PATCH /api/rest/public/operator/termination_routes/{id}

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|--------------------|---------|--|----------------------|-------------------------------|
| name | string | Route name | No | Yes, if allowed on SUB-DOMAIN |
| src_prefix | integer | Source prefix that will take this Route | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_prefix | integer | Destination prefix that will take this Route | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_rule | string | Regex-based rewrite rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| src_rewrite_result | string | Regex-based result rule for SRC number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_rule | string | Regex-based rewrite rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |
| dst_rewrite_result | string | Regex-based result rule for DST number | Yes, default is null | Yes, if allowed on SUB-DOMAIN |

Request body:

```
{
  "data": {
    "id": "a29e4de3-63d0-47ff-b098-8aaa9b4e34e1",
    "type": "termination_routes",
    "attributes": {
      "name": "Updated Route"
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "a29e4de3-63d0-47ff-b098-8aaa9b4e34e1",
    "type": "termination_routes",
    "links": {
      "self":
"https://api.telecom.center/api/rest/public/operator/termination\_routes/a29e4de3-63d0-47ff-b098-8aaa9b4e34e1"
    },
    "attributes": {
      "name": "Updated Route",
      "src_prefix": "",
      "dst_prefix": "",
      "src_rewrite_rule": null,
      "src_rewrite_result": null,
      "dst_rewrite_rule": null,
      "dst_rewrite_result": null
    }
  }
}
```

Deleting a Route

Use the HTTP DELETE method to delete a route

DELETE /api/rest/public/operator/termination_routes/{id}

Request body:

-

Response body:

-

The HTTP status code 204 will be the response after a successful deletion of a route with an empty response body.

EXTERNAL ACCOUNTS

This endpoint provides the ability to manage External Accounts on your Customer accounts.

External accounts represent objects that are outside the telecom.center™ servers and are controlled by the operator. Each sub-domain may have a unique External Account integration that allows the seamless combination of phone.systems™ PBX services with third-party operator services. See the [SUB-DOMAIN CUSTOMIZATION](#) section regarding supported sub-domain settings.

NOTE - External Accounts cannot be created or deleted by customers. A customer may only update allowed settings for an External Account that was provisioned to their account.

The following HTTP method(s) are available:

- GET /external_account_routes - For [Retrieving External Accounts](#)
- POST /external_account_routes - For [Creating an External Account](#)
- DELETE /external_account_routes/{id} - For [Deleting an External Account](#)
- PATCH /external_account_routes/{id} - For [Updating an External Account](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------------------|-----------------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer's ID that will include this incoming trunk |
| available_did_number | available_did_numbers | To-one | Yes | DID number that will be used as external caller ID |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Creating an External Account

Use the HTTP POST method to create an External Account:

POST /api/rest/public/operator/external_account_routes

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

NOTE - If a valid available DID number relationship is included in this request, `"allow_external_calls"` will be set to `true` for this external account.

Request body:

```
{
  "data": {
    "id": "123",
    "type": "external_account_routes",
    "relationships": {
      "customer": {
        "data": {
          "id": "123",
          "type": "customers"
        }
      },
      "available_did_number": {
        "data": {
          "id": "32d039d5-f45c-44a5-ad01-a1ce39eafbe0",
          "type": "available_did_numbers"
        }
      }
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "123",
    "type": "external_account_routes",
    "links": {
      "self":
"https://api.telecom.center/api/rest/public/operator/external\_account\_routes/123"
    },
    "attributes": {
      "incoming_internal": false,
      "incoming_external": false,
      "outgoing_internal": false,
      "outgoing_external": false,
      "on_demand": false,
      "caller_id_name": null,
      "internal_caller_id": null,
      "external_caller_id": "987654345678",
      "allow_external_calls": true
    }
  }
}
```

Retrieving External Accounts

Use the HTTP GET method to retrieve a list of External Accounts:

GET /api/rest/public/operator/external_account_routes

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|------------------------------|---------------------|--|
| filter[id] | External Account ID | Used for filtering results by resource ID |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[external_caller_id] | integer | Used for filtering results by external caller ID |
| filter[allow_external_calls] | "true", "false" | Used for filtering results by allowed external calls |

Request body:

-

Response body:

```
{
  "data": [
    {
      "id": "123",
      "type": "external_account_routes",
      "links": {
        "self":
"https://api.telecom.center/api/rest/public/operator/external_account_routes/123"
      },
      "attributes": {
        "incoming_internal": false,
        "incoming_external": false,
        "outgoing_internal": false,
        "outgoing_external": false,
        "on_demand": false,
        "caller_id_name": null,
        "internal_caller_id": null,
        "external_caller_id": "987654345678",
        "allow_external_calls": true
      }
    }
  ],
  "meta": {
    "record_count": 1
  },
  "links": {
    "first":
"https://api.telecom.center/api/rest/public/operator/external_account_routes?page%5Bnumber%5D=1&page%5Bsize%5D=50",
    "last":
"https://api.telecom.center/api/rest/public/operator/external_account_routes?page%5Bnumber%5D=1&page%5Bsize%5D=50"
  }
}
```

Additionally, an External Account ID may be included in the request URL in order to retrieve the details of a specific External Account:

GET /api/rest/public/operator/external_account_routes/{id}

Updating an External Account

Use the HTTP PATCH method to update an External Account:

PATCH /api/rest/public/operator/external_account_routes/{id}

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Request body:

```
{
  "data": {
    "id": "123",
    "type": "external_account_routes",
    "relationships": {
      "available_did_number": {
        "data": {
          "id": "123",
          "type": "available_did_numbers"
        }
      }
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "1243",
    "type": "external_account_routes",
    "links": {
      "self":
"https://pbx2-staging.in.didww.com/api/rest/public/operator/external\_account\_routes/1243"
    },
    "attributes": {
      "incoming_internal": false,
      "incoming_external": false,
      "outgoing_internal": false,
      "outgoing_external": false,
      "on_demand": false,
      "caller_id_name": null,
      "internal_caller_id": null,
      "external_caller_id": "121200001",
      "allow_external_calls": true
    }
  }
}
```

Deleting an External Account

Use the HTTP DELETE method to delete an External Account:

DELETE /api/rest/public/operator/external_account_routes/{id}

Request body:

-

Response body:

-

The HTTP status code 204 will be the response after a successful deletion of an External Account with an empty response body.

SIP ACCOUNTS

This endpoint provides the ability to manage SIP Accounts on your customer accounts.

The customer can create, delete and modify the configuration manually. However, depending on your pricing model, the number of allowed SIP Accounts may be limited when using the [“Customers”](#) endpoint.

When the SIP Account limitation is reached, the ability to “create” a new SIP Account will be locked on the UI, while existing SIP Accounts will remain functional.

The following HTTP method(s) are available on this endpoint:

- GET /sip_device_routes - For Retrieving a Route
- PATCH /sip_device_routes/{id} - For Updating a Route

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------|---------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer’s ID that will include this incoming trunk |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Retrieving SIP Accounts

Use the HTTP GET method to retrieve a list of SIP Accounts

GET /api/rest/public/operator/sip_device_routes

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|------------------------------|-----------------|--|
| filter[id] | SIP Account ID | Used for filtering results by resource ID |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[username] | string | Used for filtering results by SIP Account username |
| filter[caller_id_name] | string | Used for filtering results by Caller ID name |
| filter[allow_external_calls] | "true", "false" | Used for filtering results by allowed outbound calls |
| filter[internal_caller_id] | integer | Used for filtering results by Internal Caller ID |
| filter[external_caller_id] | integer | Used for filtering results by External Caller ID |

Request body:

```
-
```

Response body:

```
{
  "data": [
    {
      "id": "c1c798dc-ee42-462e-8fca-0fa96d6fe3c0",
      "type": "sip_device_routes",
      "links": {
        "self":
          "https://api.telecom.center/api/rest/public/operator/sip_device_routes/c1c7
```

```

98dc-ee42-462e-8fca-0fa96d6fe3c0"
  },
  "attributes": {
    "username": "z4wnypvker",
    "password": "08hxobdp5z",
    "incoming_internal": false,
    "incoming_external": false,
    "outgoing_internal": false,
    "outgoing_external": false,
    "on_demand": false,
    "caller_id_name": "Helo bye",
    "internal_caller_id": null,
    "external_caller_id": null,
    "allow_external_calls": false,
    "is_online": false,
    "user_agent": null,
    "use_allowed_ips": false,
    "allowed_ips":
"0.0.0.0/0,0000:0000:0000:0000:0000:0000:0000:0000/0"
  }
}
],
"meta": {
  "record_count": 1
},
"links": {
  "first":
"https://api.telecom.center/api/rest/public/operator/sip_device_routes?page
%5Bnumber%5D=1&page%5Bsize%5D=50",
  "last":
"https://api.telecom.center/api/rest/public/operator/sip_device_routes?page
%5Bnumber%5D=1&page%5Bsize%5D=50"
}
}
}

```

Additionally, a SIP Account ID may be included in the request URL to retrieve a specific SIP Account:

GET /api/rest/public/operator/sip_device_routes/{id}

Updating a SIP Account

Use the HTTP POST method to update an existing SIP Account:

PATCH /api/rest/public/operator/sip_device_routes/{id}

This method supports [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Supported attributes:

| Attribute | Value | Description | Optional | Controlled via UI |
|-----------------|-----------------|---|----------|-------------------|
| use_allowed_ips | "true", "false" | Used to enable IP-based verification for SIP Account registration | Yes | Yes |
| allowed_ips | string | Used to specify allowed IP addresses for SIP Account registration | Yes | Yes |

Request body:

```
{
  "data": {
    "id": "c1c798dc-ee42-462e-8fca-0fa96d6fe3c0",
    "type": "sip_device_routes",
    "attributes": {
      "use_allowed_ips": "true",
      "allowed_ips": "127.0.0.1"
    }
  }
}
```

Response body:

```
{
  "data": {
    "id": "c1c798dc-ee42-462e-8fca-0fa96d6fe3c0",
    "type": "sip_device_routes",
    "links": {
      "self":
"https://api.telecom.center/api/rest/public/operator/sip\_device\_routes/c1c798dc-ee42-462e-8fca-0fa96d6fe3c0"
    },
    "attributes": {
      "username": "z4wnypvker",
      "password": "08hxobdp5z",
      "incoming_internal": false,
      "incoming_external": false,
      "outgoing_internal": false,
      "outgoing_external": false,
      "on_demand": false,
      "caller_id_name": "Helo bye",
      "internal_caller_id": null,
      "external_caller_id": null,
      "allow_external_calls": false,
      "is_online": false,
      "user_agent": null,
      "use_allowed_ips": true,
      "allowed_ips": "127.0.0.1/32"
    }
  }
}
```

TERMINATION CDRS

This endpoint provides the ability to retrieve Termination CDR history.

Termination CDRs reflect all calls that originated from the PBX to Termination Gateways (PSTN), and these records include outbound calls as well as call forwarding to PSTN. In general, these calls are billed by the termination provider.

The following HTTP method is available for this endpoint:

- GET /termination_cdrs - For [Retrieving Termination CDRs](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|---------------------|---------------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer's ID that will include this incoming trunk |
| termination_route | termination_routes | To-one | No | Route that was used to terminate a particular call |
| termination_gateway | termination_gateway | To-one | No | Gateway that was used to terminate a particular call |

NOTE - Relationships can be used with `include` and `fields` parameters if the method supports it.

Retrieving Termination CDRs

Use the HTTP GET method to retrieve Termination CDRs

GET /api/rest/public/operator/termination_cdrs

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Other supported parameters:

| Parameter | Value | Description |
|-------------------------|------------------------------|---|
| filter[id] | | Used for filtering results by Termination CDR ID |
| filter[aleg_code] | Leg A disconnect code | Used for filtering results by Leg A disconnect code |
| filter[aleg_reason] | Leg A disconnect reason | Used for filtering results by Leg A disconnect reason |
| filter[src_name_in] | Source Name | Used for filtering results by Source Name |
| filter[src_number_in] | Source Number | Used for filtering results by Source Number |
| filter[dst_number_in] | Destination Number | Used for filtering results by Destination Number |
| filter[orig_call_id] | Originating call ID | Used for filtering results by Originating call ID |
| filter[local_tag] | Local tag | Used for filtering results by Local tag |
| filter[global_tag] | Global tag | Used for filtering results by Global tag |
| filter[time_start] | Call time start | Used for filtering results by Call time start |
| filter[time_start_gteq] | Call time start greater than | Used for filtering results by Call start time. Returns all results that exceed the specified time |
| filter[time_start_lted] | Call time start less than | Used for filtering results by Call start time. Returns all |

| | | |
|--------------------------------|------------------------|--|
| | | results that are less than the specified time |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[termination_gateway.id] | Termination Gateway ID | Used for filtering results by Termination Gateway ID |
| filter[termination_route.id] | Termination Route ID | Used for filtering results by Termination Route ID |

Request body:

```
-
```

Response body:

```
{
  "data": [
    {
      "id": "123",
      "type": "termination cdrs",
      "attributes": {
        "aleg_reason": "string",
        "src_name_in": "string",
        "src_number_in": "string",
        "dst_number_in": "string",
        "orig_call_id": "string",
        "local_tag": "string",
        "global_tag": "string",
        "duration": 0,
        "aleg_code": 0,
        "time_start": "1970-01-01 00:00",
        "time_connect": "1970-01-01 00:00",
        "time_end": "1970-01-01 00:00"
      },
      "relationships": {
        "customer": {
          "data": {
            "id": "123",
            "type": "customers"
          }
        }
      }
    }
  ]
}
```

The response body will contain an array of Termination CDRs

In addition, a Termination CDR ID may be included in the request URL to retrieve a single CDR:

GET /api/rest/public/operator/termination_cdrs/{id}

INCOMING TRUNK CDRS

This endpoint provides the ability to retrieve Incoming Trunk CDR history.

Incoming Trunk CDRs reflect all calls that were received on the PBX via Incoming Trunks (DID providers). Internal calls between Internal Numbers (extensions) are not included.

The following HTTP method is available:

- GET /incoming_trunk_cdcs - For [Retrieving Incoming Trunk CDRs](#)

Relationship to other resources:

| Name | Resource Type | Relationship Type | Updatable | Description |
|----------------|-----------------|-------------------|-----------|--|
| customer | customers | To-one | No | Parent customer's ID that will include this incoming trunk |
| incoming_trunk | incoming_trunks | To-one | No | Incoming Trunk that was used to receive a particular call |

Retrieving Incoming Trunk CDRs

Use the HTTP GET method to retrieve Incoming Trunk CDRs

GET /api/rest/public/operator/incoming_trunk cdrs

This method supports [Sorting](#), [Pagination](#), [Sparse Fieldsets](#) and [Inclusion of Related Resources](#)

Supported parameters:

| Parameter | Value | Description |
|---------------------|-----------------------|---|
| filter[id] | Incoming Trunk ID | Used for filtering results by Incoming Trunk ID |
| filter[customer.id] | Customer ID | Used for filtering results by Customer ID |
| filter[domain] | Incoming Trunk domain | Used for filtering results by Incoming Trunk domain |

Request body:

```
-
```

Response body:

```
{
  "data": [
    {
      "id": "123",
      "type": "Component cdrs",
      "attributes": {
        "duration": "string",
        "src_name": "string",
        "src_number": "string",
        "dst_number": "string",
        "did_number": "string",
        "lega_disconnect_code": "string",
        "lega_disconnect_reason": "string",
        "lega_call_id": "string",
        "success": "string",
        "time_start": "1970-01-01 00:00",
        "time_end": "1970-01-01 00:00",
        "time_connect": "1970-01-01 00:00"
      },
      "relationships": {
        "customer": {
          "data": {
            "id": "123",
            "type": "customers"
          }
        }
      }
    }
  ]
}
```

The response body will contain an array of Incoming Trunk CDRs

In addition, an Incoming Trunk CDR ID may be included in the request URL to retrieve a single CDR:

GET /api/rest/public/operator/termination_cdrs/{id}

API OVERVIEW

CUSTOMERS

- **GET /customers** - returns a list of customers
- **POST /customers** - creates a customer
- **GET /customers/{id}** - returns a single customer
- **PATCH /customers/{id}** - updates a customer

CUSTOMER SESSIONS

- **POST /sessions** - creates a session for customer to access the UI

ADMIN SESSIONS

- **POST /customer_admin_sessions** - creates a session for admin to access the customer UI

INCOMING TRUNKS

- **GET /incoming_trunks** - returns a list of incoming trunks
- **POST /incoming_trunks** - creates an incoming trunk
- **DELETE /incoming_trunks/{id}** - deletes an incoming trunk
- **GET /incoming_trunks/{id}** - returns a single incoming trunk
- **PATCH /incoming_trunks/{id}** - updates an incoming trunk

DID NUMBERS

- **GET /available_did_numbers** - returns a list of available DID numbers
- **POST /available_did_numbers** - creates an available DID number
- **DELETE /available_did_numbers/{id}** - deletes an available DID number
- **GET /available_did_numbers/{id}** - returns an available DID number
- **PATCH /available_did_numbers/{id}** - updates an available DID number

GATEWAYS

- **DELETE /termination_gateways/{id}** - deletes a termination gateway
- **GET /termination_gateways/{id}** - returns a single termination gateway
- **PATCH /termination_gateways/{id}** - updates a termination gateway
- **GET /termination_gateways** - returns a list of termination gateways

- **POST /termination_gateways** - creates a termination gateway

EXTERNAL ACCOUNTS

- **GET /external_account_routes** - returns a list of external account routes (SIM Cards)
- **POST /external_account_routes** - creates an external account route (SIM Card)
- **GET /external_account_routes/{id}** - returns a single external account route (SIM Card)
- **PATCH /external_account_routes/{id}** - updates an external account route (SIM Card)

SIP ACCOUNTS

- **GET /sip_device_routes** - returns list of SIP device routes
- **GET /sip_device_routes/{id}** - return single SIP device route
- **PATCH /sip_device_routes/{id}** - update SIP device route

TERMINATION CDRs

- **GET /termination cdrs** - returns a list of termination CDRs
- **GET /termination cdrs/{id}** - returns a single termination CDR

INCOMING TRUNK CDRS

- **GET /incoming_trunk cdrs** - returns a list of incoming trunk CDRs
- **GET /incoming_trunk cdrs/{id}** - returns a single incoming trunk CDR