# Python Orchestrator API (v1) reference document

## Introduction

This document contains a specification draft for the PaaS Orchestrator component API. The API is primarily intended to be used with its respective web dashboard (https://github.com/infn-datacloud/orchestrator-dashboard) or through the orchent (https://github.com/infn-datacloud/orchent) tool. The API definition will follow the guidelines defined in REST API Common behaviors - INFN Cloud - INFN Confluence.

For backward compatibility, this component MUST support the essential operations provided by the legacy INDIGO PaaS Orchestrator component (https://github.com/infn-datacloud/orchestrator) and used by the web dashboard and orchent. The latest documentation of the API provided by the legacy component is available at https://infn-datacloud.github.io/orchestrator/restdocs/.

The current implementation is located at

## Deprecated endpoints

All the endpoints related to a deployment schedule of the legacy REST API will not be migrated (/schedules).

# Functionalities

Set of functionalities provided by the API

| Functionality | Description |
|---|---|
| **Configuration [GET]** | GET: Retrieve micro-services endpoints |
| **Health [GET]** | GET: Endpoint to contact to inspect service health |
| **Users [GET, POST, DELETE]** | GET: Return the list of registered users or a specific user.<br>POST: Create a user instance in the DB.<br>DELETE: Remove a user from the DB. |
| **Templates [GET, POST, PATCH, DELETE]** | GET: Return the list of registered templates or a specific template.<br>POST: Create a template instance in the DB.<br>PATCH: Update the template DB entry.<br>DELETE: Remove a template from the DB. |
| **Deployments [GET, POST, PATCH, DELETE]** | POST: Create a deployment instance.<br>GET: Return the list of deployments the user can access to. Return the specific deployment (only if the user can access to).<br>PUT: Re-deploy an existing instance with changes.<br>PATCH: Update the deployment DB entry.<br>DELETE: Delete a deployment instance. |
| **Deployment's resources [GET, PATCH]** | GET: Return the list of resources of a deployment the user can access to. Return the specific resource of a deployment (only if the user can access that deployment).<br>PATCH: Start/Stop a computing node. |
| **Deployment logs [GET, DELETE]** | GET: Return the external services (IM, Rucio) logs related to the target deployment<br>POST: Create or overwrite specific deployment logs.<br>DELETE: Delete deployment logs. |

# Health

Endpoint used to monitor service health.

## Retrieve service health

### *Description*

The GET request is used to check service health.

**No authentication required.**

### *Submission*

| | |
|---|---|
| **Method** | GET |
| **Target** | /api/v1/health |
| **Content-Type** | application/json |
| **HTTP response codes** | 200, 500 |

### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/health' -i -H 'Accept:
application/json'

GET /health HTTP/1.1
Accept: application/json
Host: orchestrator-api.example.com
```

### *Response fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **status** | General service status: healthy/unhealthy. | String | No |
| **db_connection** | Return 'healthy' if the db connection is working, unhealthy' if there are communication problems | String | No |
| **vault_connection** | When vault connection is enabled, return 'healthy' if the connection is active, 'unhealthy' if there are communication problems | String | Yes |
| **opa_connection** | When OPA connection is enabled, return 'healthy' if the connection is active, 'unhealthy' if there are communication problems | String | Yes |
| **kafka_connection** | When kafka connection is enabled, return 'healthy' if the connection is active, 'unhealthy' if there are communication problems | String | Yes |
| **timestamp** | Response timestamp. | String ISO 8601 | No |
| **details** | Message with additional information | String | Yes. Present only when unhealthy. |

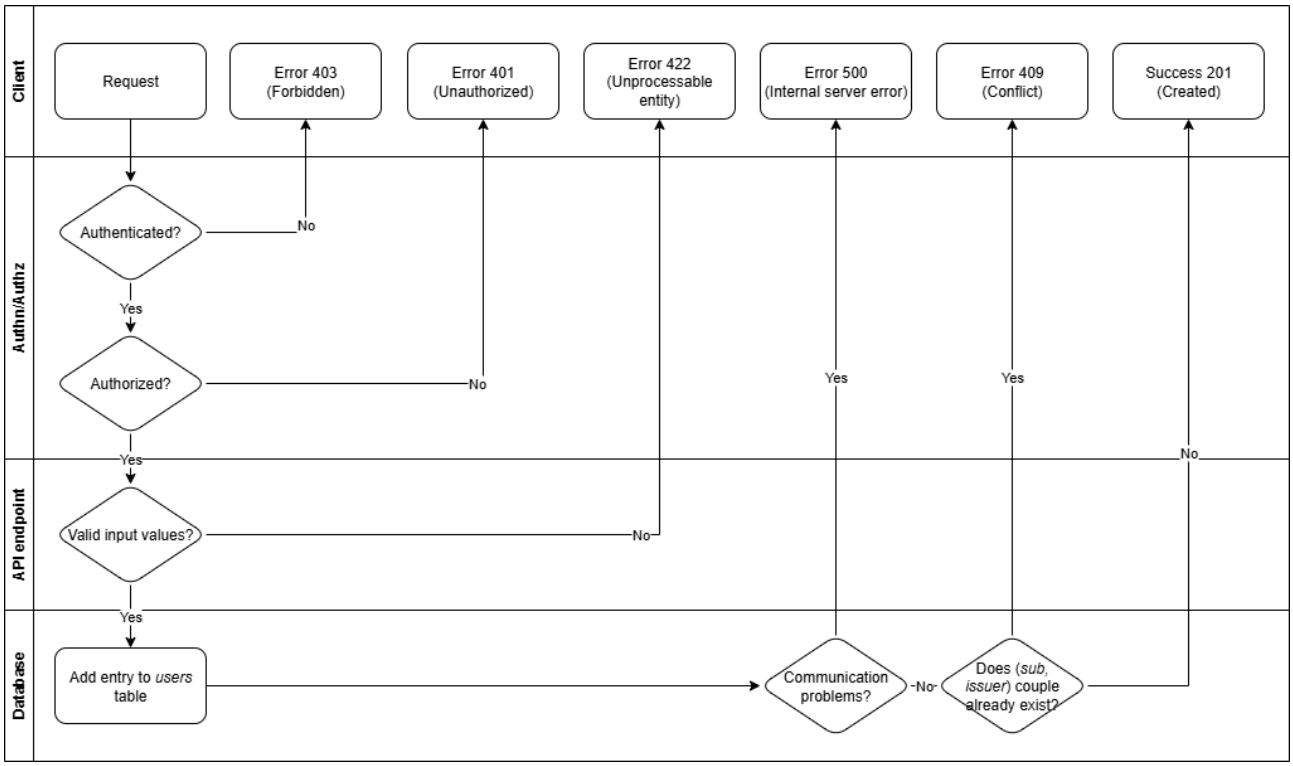
*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 342

{
  "status": "unhealthy",
  "db_connection": "healthy",
  "vault_connection": "healthy",
  "opa_connection": "healthy",
  "kafka_connection": "unhealthy",
  "timestamp": "2024-03-01T09:56+0000",
  "details": "Reason"
}
```

# Users

Users must be persisted to track their actions and to show their names and details when listing users' deployment. Users are managed as OpenID Connect users via the OAuth2 framework provided by INDIGO IAM or any trusted identity provider implementing this protocol. If the user's access token is authorized (by OPA reading the "group" claim) to login, at first access the user's sub is recorded in the user table of the database.

## Create a new User




### *Description*

The POST request allows the client to register an OpenID user (e.g., from INDIGO IAM Identity/Access Token) subject. The details to add to the DB are inferred from the *access token* and they are: *sub, name, iss, email and username*.

### *Submission*

| **Method** | **POST** |
|---|---|
| **Target** | /api/v1/users |
| **Possible response codes** | 204, 401, 403, 409, 422 |

### *Request's body fields description*

No body required.

### *Request submission example*

```
$ curl --location --request POST 'http://orchestrator-api.com/api/v1/users/' --header
'Authorization: Bearer <access-token>'

> POST /api/v1/users/ HTTP/1.1
> Host: orchestrator-api.com
> User-Agent: curl/7.81.0
```

```
> Accept: */*
> Authorization: Bearer <access-token>
```

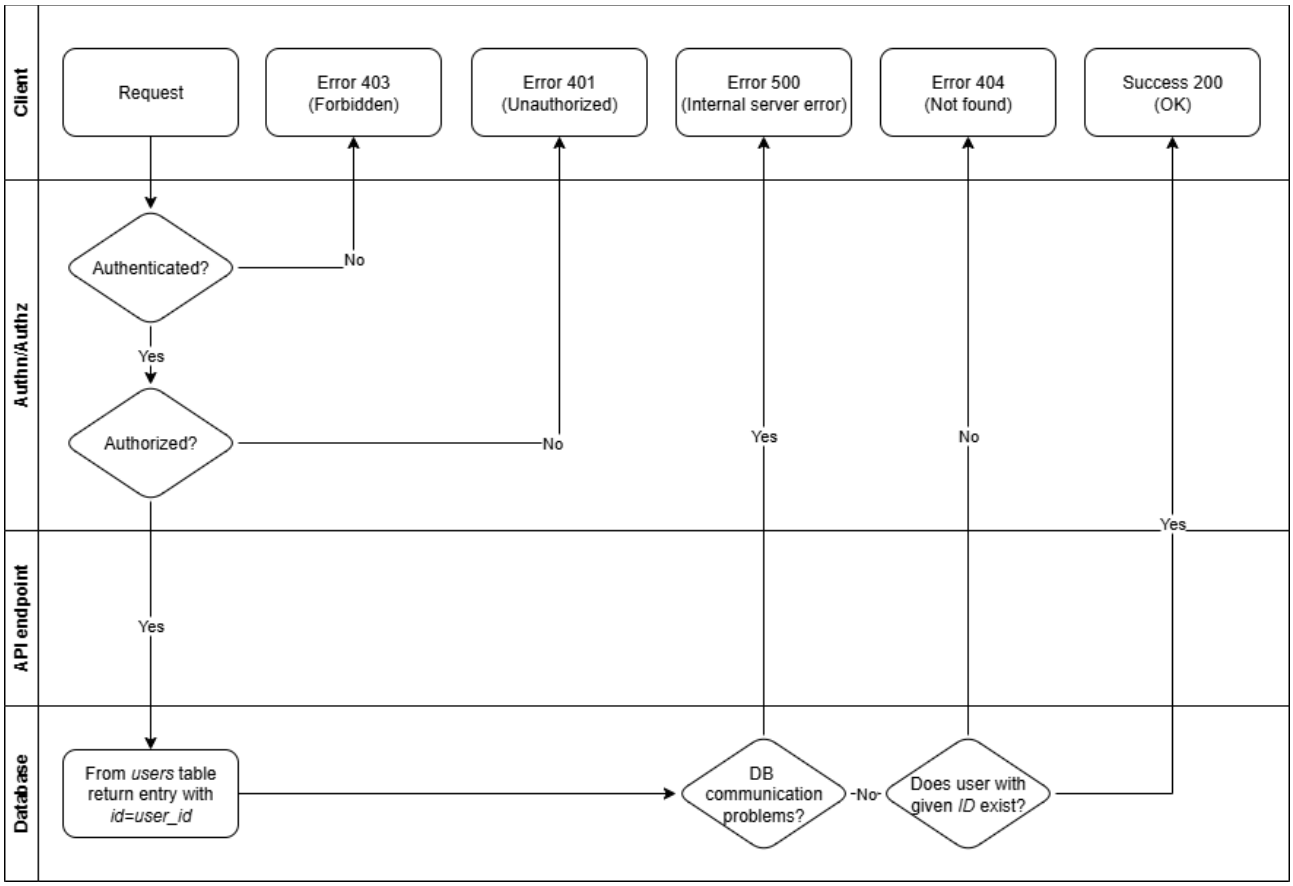
*Response fields description*

| Name | Description | Field type | Optional for the server | Remarks |
|---|---|---|---|---|
| **id** | The user id | UUID String | No | |

*Response from server example*

```
< HTTP/1.1 201 Created
< date: Fri, 01 Aug 2025 07:22:42 GMT
< server: uvicorn
< Connection: Keep-Alive
< Location: https://fed-mgr-api.example.org:1234/api/v1/users/e2b8f2b0-9952-11ec-9c53-fa163e0f6dc7
< content-length: 45
< content-type: application/json

{
  "id": "e2b8f2b0-9952-11ec-9c53-fa163e0f6dc7"
}
```

## Retrieve a User




### *Description*

The GET request allows the client to retrieve information about a registered user given the ID.

### *Submission*

| **Method** | **GET** |
|---|---|
| **Target** | /api/v1/users/{id} |
| **Content-Type** | None |
| **Possible response codes** | 200, 401, 403, 404 |

### *Request's parameters fields description*

No parameters.

### *Request submission example*

```
$ curl --location 'http://orchestrator-api.com/api/v1/users/e2b8f2b0-9952-11ec-9c53-
fa163e0f6dc7' --header 'Authorization: Bearer <access-token>'

> GET /api/v1/users/e2b8f2b0-9952-11ec-9c53-fa163e0f6dc7 HTTP/1.1
> Host: orchestrator-api.com
> Accept: */*
> Authorization: Bearer <access-token>
```

### *Response fields description*

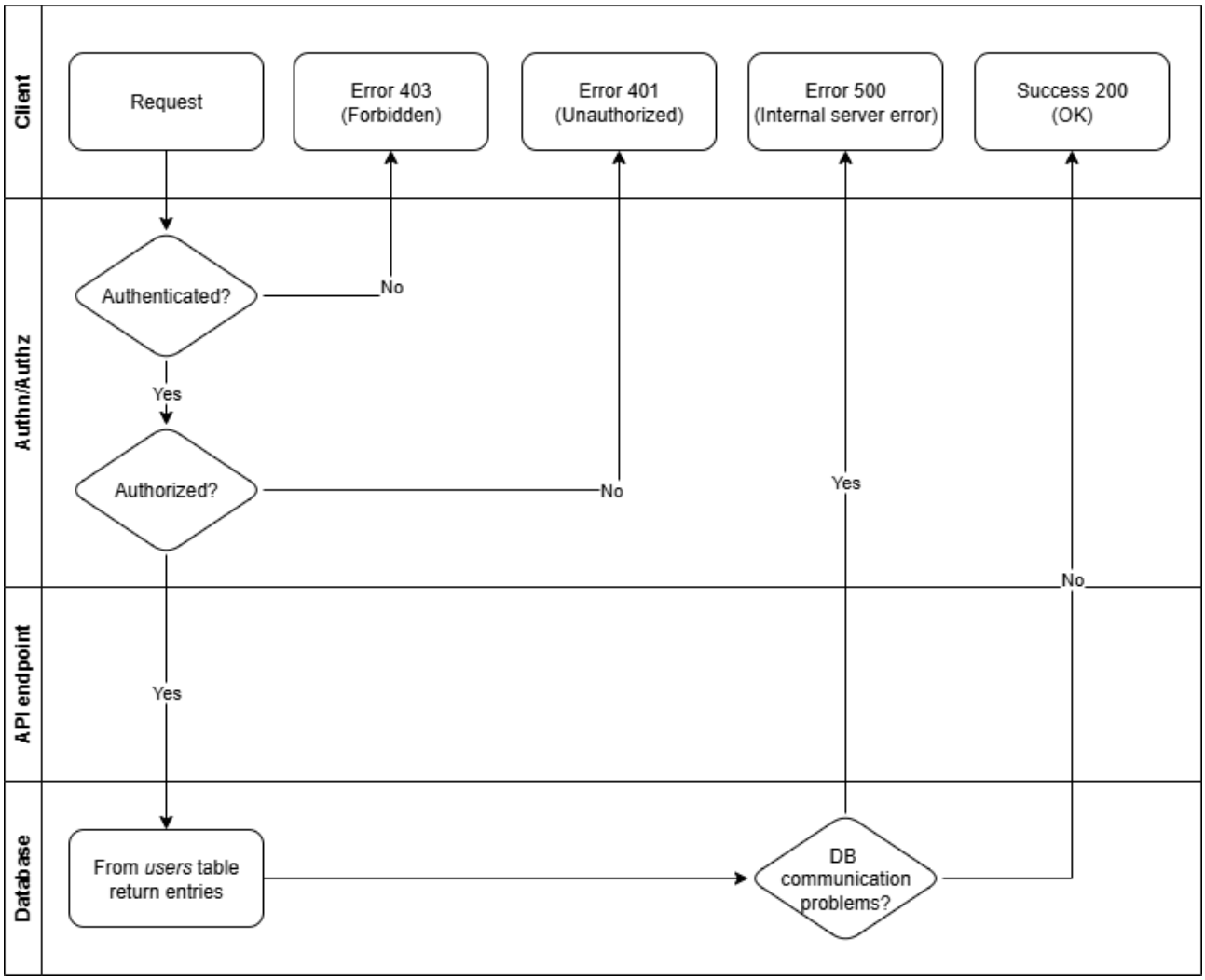
| Name | Description | Field type | Optional |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **id** | The user id | UUID String | No |
| **created_at** | The time the server received the request | String ISO 8601 | No |
| **sub** | OIDC's subject | UUID String | No |
| **issuer** | OAuth2 issuer's URL | String URI format | No |
| **name** | User's friendly name | String | No |
| **username** | User preferred username. Used by the dashboard only | String | No |
| **email** | User's registered email | String | No |
| **picture** | Avatar generated from the orchestrator on first login. Can be a link pointing to a generic image. Default None. | String URI format | Yes |
| **public_ssh_key** | Public key to install the generated VM. Default None. | String | Yes |
| **refresh_token** | Save the refresh token. Avoid user reauthentication for time consuming deployments. Default None. | String | Yes |

*Response from server example*

```
< HTTP/1.1 200 OK
< date: Fri, 01 Aug 2025 08:13:52 GMT
< server: uvicorn
< content-length: 1038
< content-type: application/json

{
  "id": "e2b8f2b0-9952-11ec-9c53-fa163e0f6dc7",
  "created_at": "2024-01-01T10:00:00.000000Z",
  "sub": "1234-abcd-56",
  "issuer": "https://iam.cloud.infn.it",
  "name": "Mario Rossi",
  "username": "mrossi",
  "email": "mario.rossi@cnaf.infn.it",
  "picture":
"https://www.gravatar.com/avatar/e4ac6dabf304314d7b3e624fd81a29d8?d=identicon&s=26",
  "public_ssh_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCYiIaQ//o55wRaKMgZGfMUMRG2xDnQGYySbhAlzv00Ye+m4m0tjXCKk
D+LNexbpnhctxG7kbBi/frswn6f4QRUkk+kOgH4BCzx/GSpwBFC48EMFdpH4cw+B6XUQacZPOuqc8shAjh9sE
+0PX/S6wGYhPc8sR/vmuFHoEtvmV9CiMebVHpzRM0yjPXD1s8tmPajHrArnD1uUpr7oWwYzpcU6hObUwxGPuT
F+9MiDSA5lccKBl1kw7RWfaiS+7HKhmb0QoTCRot5OH6DrCLclenrZau040WI/K+ufYFgHsTPxZ1ZRYj8Z7t0
A4Ofp1tdG3LwiXs+3edGxhs3Ye0edgnbZSkCFl57Gdgdzrf7sh5PdTBchI/dSKKb00Twt/LjflWjMdau07Uoz
75nv10s+39a1LtOlerXKbOtr95sez0k8E+OXIsQDGe3+uUrmgOac9hJDj2lTiTKqmL8Aju71CcIVSRtDdCLUJ
7ganmt0d5AI5V3GyhQlgoKhYyeJTTS130= mrossi",
  "refresh_token":
"eyJhbGciOiJub25lIn0.eyJqdGkiOiIzNGI3YjgyUy1lMAk3LTQ0YzQtOWRlZi1iNTI1MmNiMTYwZWQifQ."
}
```

## List of Users



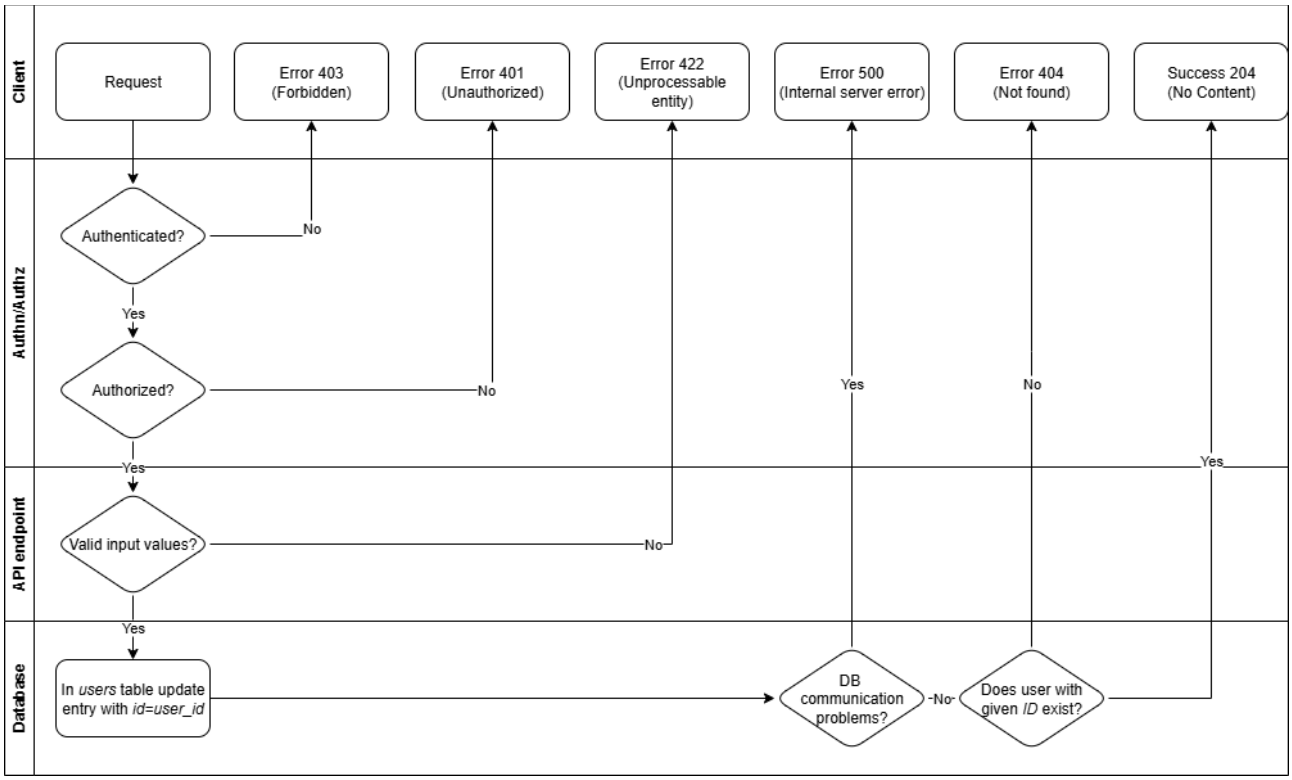

*Description*

The GET request allows the client to ask the paginated list of users (See Pagination on DataCloud Confluence page).

*Submission*

| **Method** | **GET** |
|---|---|
| **Target** | /api/v1/users |
| **Params** | Any of the user entity's attributes + page={int}&size={int}&sort={str} |
| **Possible response codes** | 200, 401, 403, 422 |

## Modify a User




### *Description*

The PATCH request allows the client to edit an existing user (useful to add details and fields).

### *Submission*

| **Method** | **PATCH** |
|---|---|
| **Target** | /api/v1/users/{id} |
| **Content-Type** | application/json |
| **Possible response codes** | 204, 401, 403, 422 |

### *Request's body fields description*

| **Name** | **Description** | **Field type** | **Optional** |
|---|---|---|---|
| **picture** | Avatar generated from the orchestrator on first login. Can be a link pointing a generic image. | String URI format | Yes |
| **public_ssh_key** | Public key to install the generated VM | String | Yes |
| **refresh_token** | Save the *refresh token*. Avoid user reauthentication for time consuming deployments | String | Yes |

### *Request submission example*

```
{
  "picture":
"https://www.gravatar.com/avatar/e4ac6dabf304314d7b3e624fd11a29d8?d=identicon&s=21",
```

```
  "public_ssh_key: "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCYiIaQ//o55wRaKMgZGfMUMRG2xDnQGYySbhAlzv00Ye+m4m0tjXCKk
D+LNexbpnhctxG7kbBi/frswn6f4QRUkk+kOgH4BCzx/GSpwBFC48EMFdpH4cw+B6XUQacZPOuqc8shAjh9sE
+0PX/S6wGYhPc8sR/vmuFHoEtvmV9CiMebVHpzRM0yjPXD1s8tmPajHrArnD1uUpr7oWwYzpcU6hObUwxGPuT
F+9MiDSA5lccKBl1kw7RWfaiS+7HKhmb0QoTCRot5OH6DrCLclenrZau040WI/K+ufYFgHsTPxZ1ZRYj8Z7t0
A4Ofp1tdG3LwiXs+3edGxhs3Ye0edgnbZSkCFl57Gdgdzrf7sh5PdTBchI/dSKKb00Twt/LjflWjMdau07Uoz
75nv10s+39a1LtOlerXKbOtr95sez0k8E+OXIsQDGe3+uUrmgOac9hJDj2lTiTKqmL8Aju71CcIVSRtDdCLUJ
7ganmt0d5AI5V3GyhQlgoKhYyeJTTS130= mrossi",
  "refresh_token:
"eyJhbGciOiJub25lIn0.eyJqdGkiOiIzNGI3YjgyUy1lMAk3LTQ0YzQtOWRlZi1iNTI1MmNiMTYwZWQifQ."
}
```

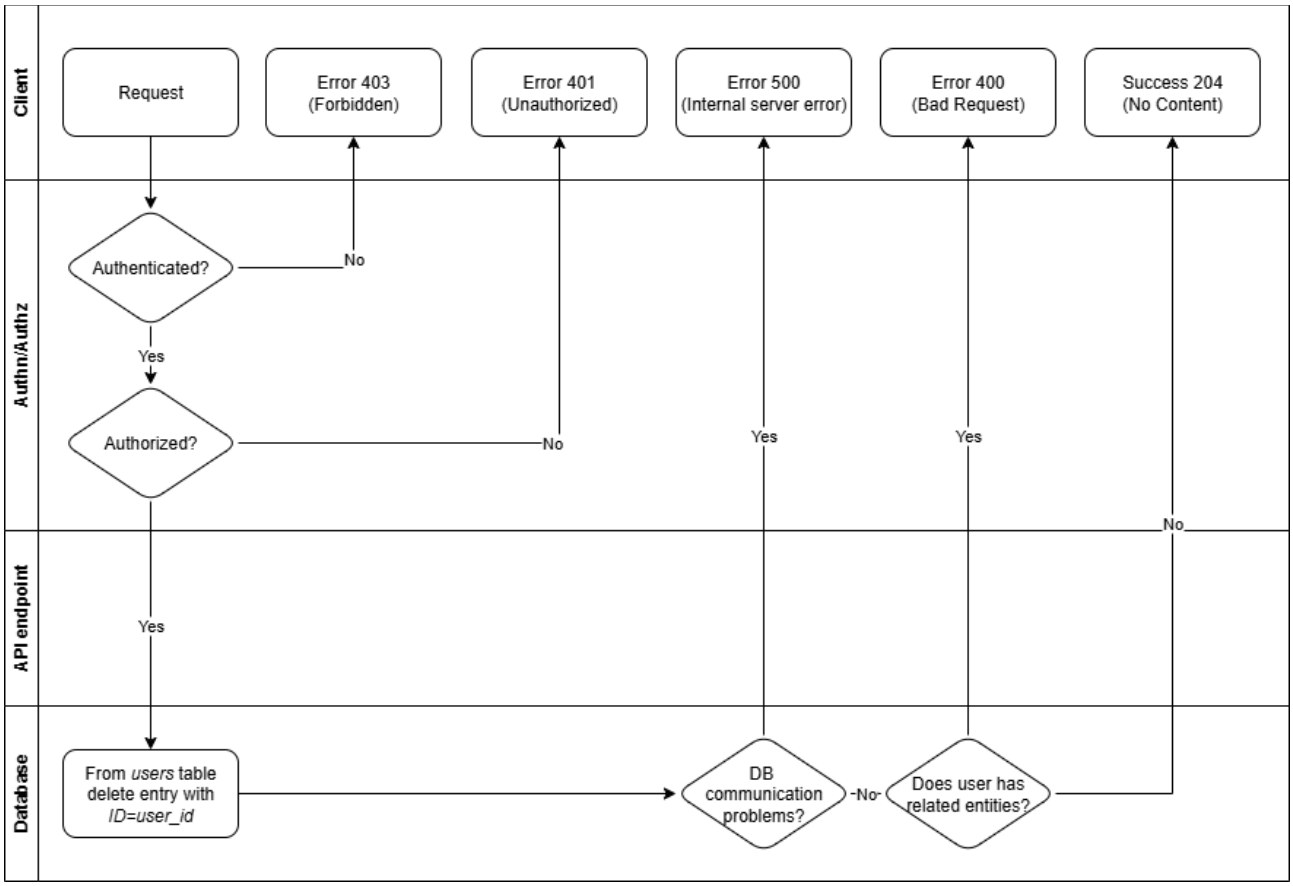
*Response fields description*

No content

*Response from server example*

```
< HTTP/1.1 204 No Content
< Connection: Keep-Alive
< Content-Length: 0
< Date: Tue, 01 Mar 2022 11:30:06 GMT
```

## Delete a User




### *Description*

Allows the client to remove a registered user. A user MUST NOT be deleted if associated with other resources (deployments and more).

### *Submission*

| **Method** | **DELETE** |
|---|---|
| **Target** | /api/v1/users/{id} |
| **Possible response codes** | 204, 401, 403, 409 |

### *Request's parameters fields description*

No parameters.

### *Request submission example*

```
$ curl --location --request DELETE 'http://orchestrator-
api.com/api/v1/users/91ae0dbe-c1fe-4546-97bc-60af7c16d6ae' --header 'Authorization:
Bearer <access-token>'

> DELETE /api/v1/users/91ae0dbe-c1fe-4546-97bc-60af7c16d6ae HTTP/1.1
> Host: localhost:8000
> Accept: */*
> Authorization: Bearer <access-token>
```

### *Response fields description*

No content

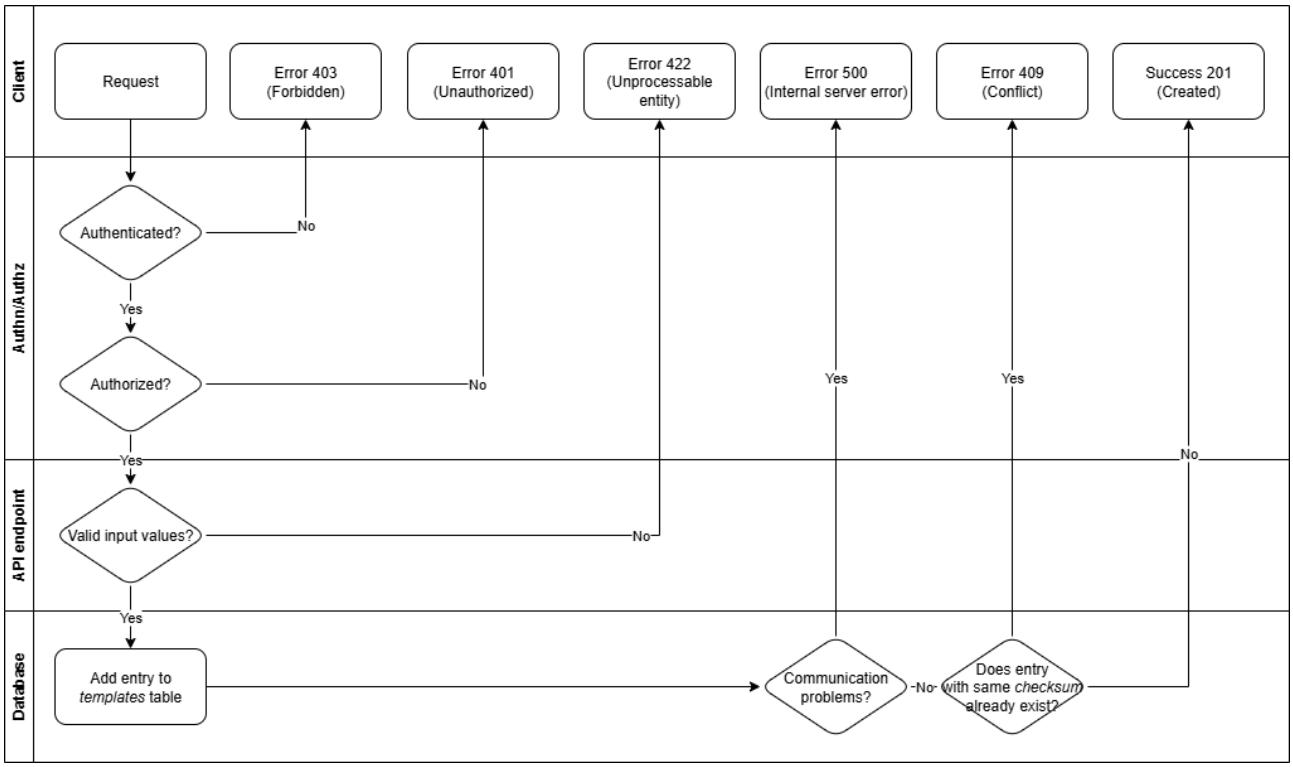
*Response from the server example*

```
< HTTP/1.1 204 No Content
< date: Fri, 01 Aug 2025 07:53:09 GMT
< server: uvicorn
< content-type: application/json
```

# Templates

A TOSCA template is a file with the resources definition that will be used to generate deployments.

## Add a Template




### *Description*

The POST request creates a TOSCA template. A TOSCA template has a unique checksum value calculated when adding a new TOSCA template or when updating the content of an existing one. This is used to compare tosca contents and avoid duplications.

If the *template_name* in the metadata field does not exist and the *name* attribute in the request's body is undefined, reject the request. The same happens when both *name* in the request and *template_name* in the metadata are defined.

### *Submission*

| **Method** | **POST** |
|---|---|
| **Target** | /api/v1/templates |
| **Content-Type** | application/json |
| **Possible response codes** | 202, 400, 401, 403, 409, 422, 500 |

### *Request's body fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **content** | A string containing a TOSCA YAML-formatted template | String | No |
| **name** | Template name. Useful for those templates without the template_name metadata. | String | Yes |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/templates' -i -X POST -H 'Content-
Type: text/x-yaml' -H 'Authorization: Bearer <access token>' -d '{"content":
"Template in YAML format"}'

POST /templates HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: example.com
Content-Length: 256

{
  "content": "Template in YAML format",
}
```

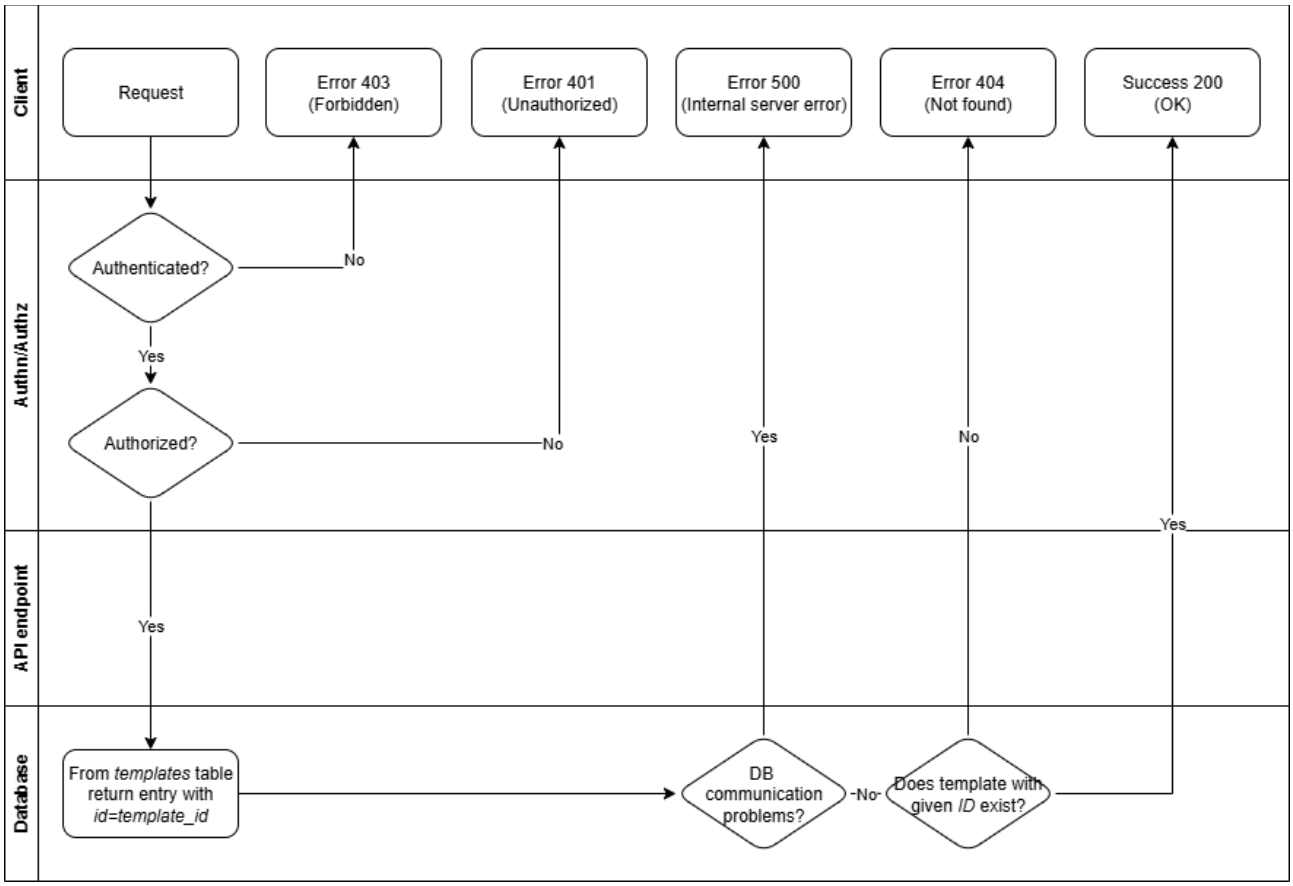
*Response fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **id** | Entity unique identifier | UUID String | No |

*Response from server example*

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 51

{
  "id": "3a1cbf3e-0d70-4036-95b4-46b3f5c15cec"
}
```

## Retrieve Template




### *Description*

The GET request retrieves a TOSCA template, by the template's id.

### *Submission*

| **Method** | **GET** |
|---|---|
| **Target** | /api/v1/templates/{id} |
| **Content-Type** | application/json |
| **Possible response codes** | 200, 401, 403, 404, 422, 500 |

### *Request fields description*

Nothing

### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/templates/34483-d937-4578-bfdb-
ebe196bf82dd' -i -H 'Authorization: Bearer <access token>'

GET /templates/34483-d937-4578-bfdb-ebe196bf82dd HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

### *Response fields description*

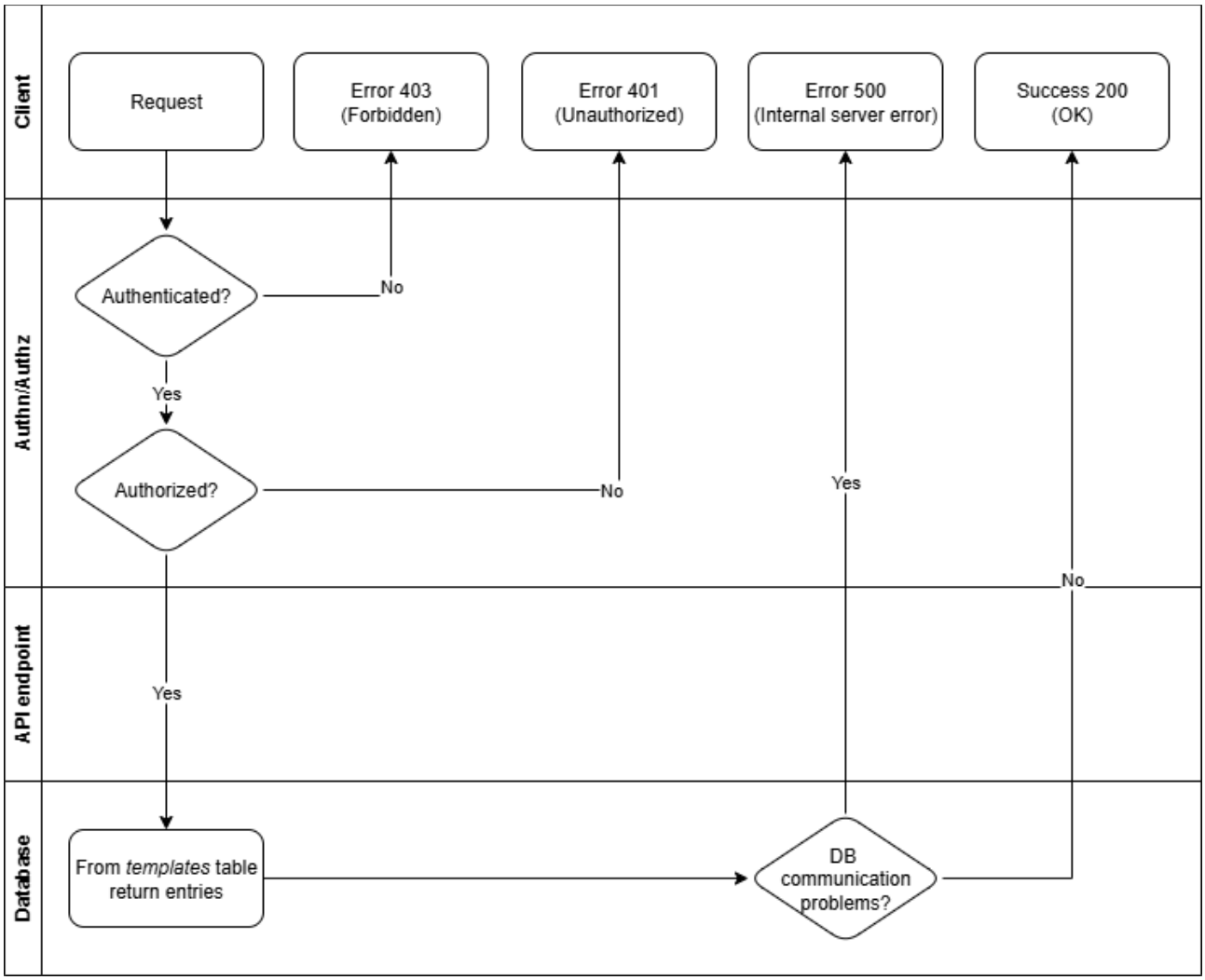
| Name | Description | Type | Optional |
|---|---|---|---|
| **id** | Unique identifier | UUID String | No |
| **created_at** | Creation timestamp. | String ISO 8601 | No |

| | | | |
|---|---|---|---|
| **created_by** | ID of the user which submitted the request | String | No |
| **updated_at** | Update timestamp. If no changes have been made it matches the *created_at* field. | String ISO 8601 | No |
| **updated_by** | ID of the user which submitted the latest update to this deployment. If no changes have been made it matches the *created_by* field. | String | No |
| **content** | TOSCA template YAML content | String | No |
| **checksum** | Checksum calculated from the TOSCA template content | String | No |
| **name** | Name assigned to this TOSCA template. | String | No |
| **version** | TOSCA template specific version. For example, multiple versions of the 'single VM' TOSCA templates can exist. Default is None. | String | Yes |
| **target_provider_type** | IaaS type this template has been written for. Default is *openstack* | String | Yes |
| **tosca_definition_version** | Version of the TOSCA standard used. Default is *tosca_simple_yaml_1_0* | String | Yes |

*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 722

{
  "id": "34483-d937-4578-bfdb-ebe196bf82dd",
  "created_at": "2024-01-01T10:00:00.000000Z",
  "created_by": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "updated_at": "2024-01-01T10:00:00.000000Z",
  "updated_by": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "content": "Template in YAML format",
  "checksum": "e092eea67daf2649d8d2b18578f09249bae61236213d83d05bff30d55e1869ab",
  "name": "single_vm/vm_with_volume",
  "version": "v1.2.3",
  "target_provider_type": "openstack",
  "tosca_definition_version": "tosca_simple_yaml_1_0",
}
```

## Retrieve Templates




### *Description*

The GET request retrieves all templates.

### *Submission*

| **Method** | **GET** |
|---|---|
| **Target** | /api/v1/templates |
| **Content-Type** | application/json |
| **Params** | Any of the template entity's attributes +<br>page={int}&size={int}&sort={str} |
| **Possible response codes** | 200, 400, 401, 403, 422, 500 |

### *Request fields description*

Clients can filter requests for any of the read attributes.

### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/templates' -i -H 'Accept:
application/json' -H 'Authorization: Bearer <access token>'

GET /deployments HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
```

```
Host: orchestrator-api.example.com
```

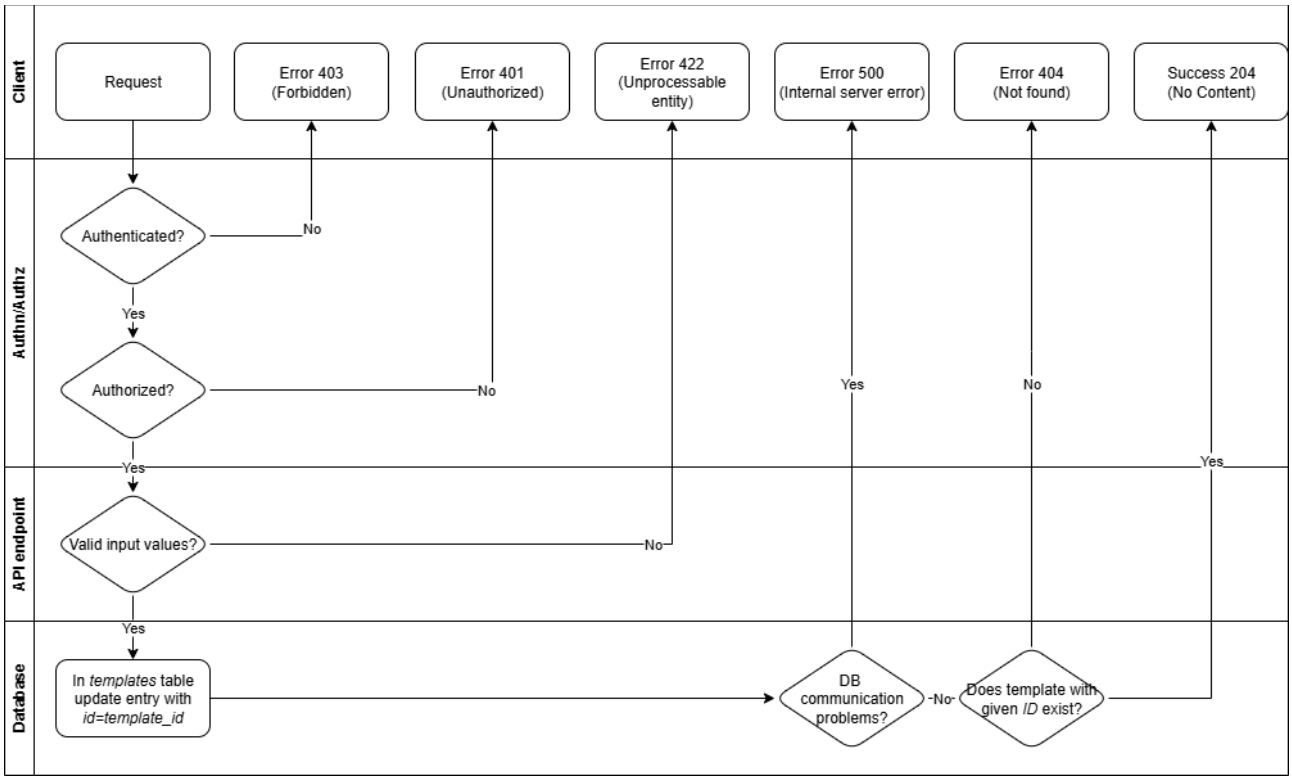
*Response fields description*

See Pagination for main fields. The *data* field contains the list of deployments. Each item has the shape of the response returned by Retrieve template

*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2059

{
  "data": [List of Templates],
  "links": {
    "first": "http://example.com/api/v1/templates?page=1&size=5",
    "prev": "http://example.com/api/v1/templates?&page=1&size=5",
    "next": "http://example.com/api/v1/templates?page=3&size=5",
    "last": "http://example.com/api/v1/templates?page=10&size=5",
  }
  "page": {
    "size": 5,
    "total_elements": 50,
    "total_pages": 10
    "number": 3
  }
}
```

## Modify a Template




### *Description*

Allows the client to edit an existing template metadata (useful to patch values).

### *Submission*

| **Method** | **PUT** |
|---|---|
| **Target** | /api/v1/templates/{id} |
| **Content-Type** | application/json |
| **Possible response codes** | 204, 400, 401, 403, 409, 422, 500 |

### *Request's body fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **name** | Name assigned to this TOSCA template. | String | No |
| **version** | TOSCA template specific version. For example, multiple versions of the 'single VM' TOSCA templates can exist. | String | Yes |
| **target_provider_type** | IaaS type this template has been written for. | String | Yes |
| **tosca_definition_version** | Version of the TOSCA standard used. | String | Yes |

### *Request submission example*

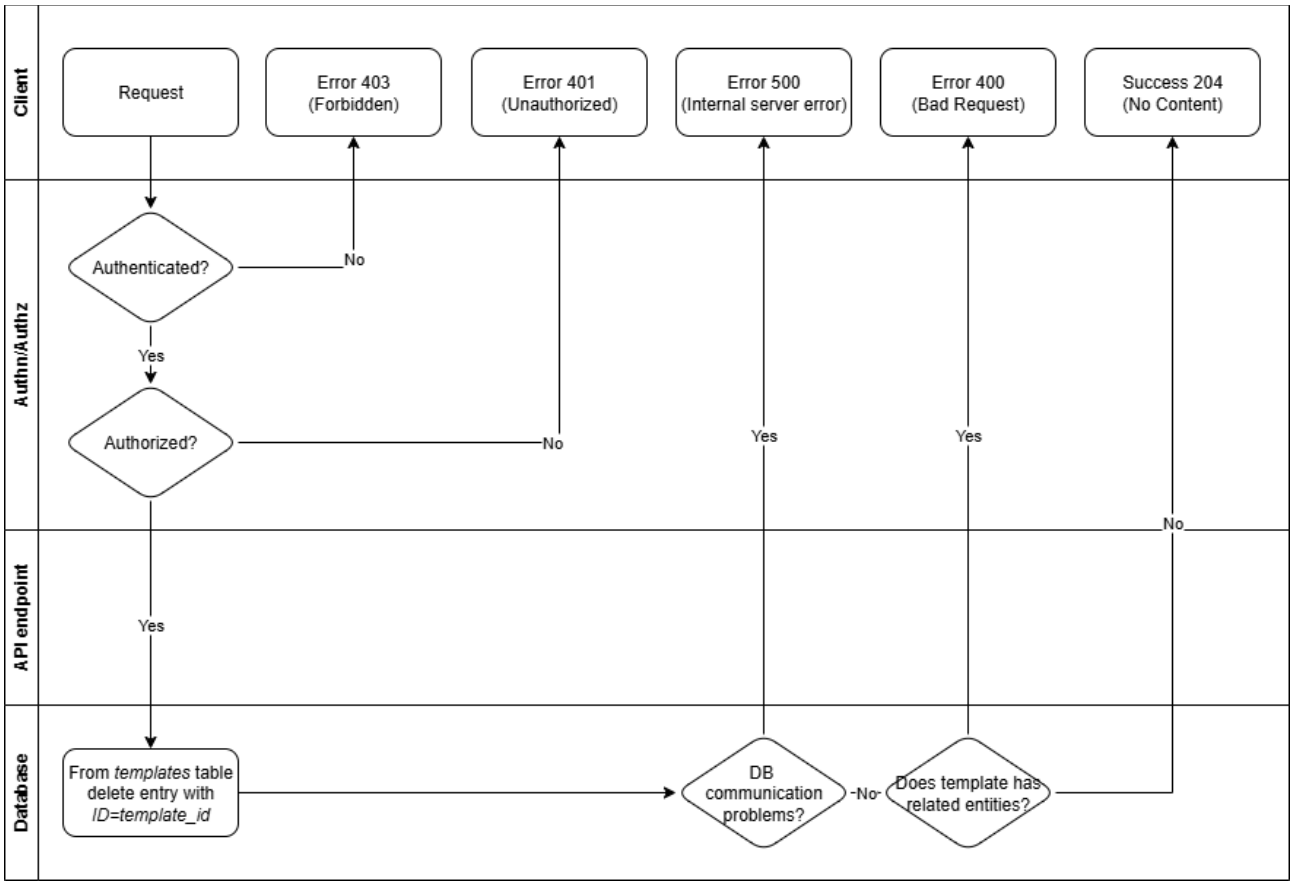
```
$ curl 'http://orchestrator-api.example.com/api/v1/templates/' -i -X POST -H
'Content-Type: text/x-yaml' -H 'Authorization: Bearer <access token>' -d 'Template in
YAML format'

POST /templates HTTP/1.1
Content-Type: text/x-yaml
```

```
Authorization: Bearer <access token>
Host: example.com
Content-Length: 256

{
  "name": "single_vm/vm_with_volume",
  "version": "v1.2.3",
  "target_provider_type": "openstack",
  "tosca_definition_version": "tosca_simple_yaml_1_0"
}
```

*Response from server example*

```
< HTTP/1.1 204 No Content
< Connection: Keep-Alive
< Content-Length: 0
< Date: Tue, 01 Mar 2022 11:30:06 GMT
```

## Delete Template




### *Description*

Allows the client to remove a TOSCA template. If at least one deployment is using this template, the TOSCA template can't be deleted.

### *Submission*

| **Method** | **DELETE** |
|---|---|
| **Target** | /api/v1/templates/{id} |
| **Content-Type** | application/json |
| **Possible response codes** | 204, 401, 403, 409 |

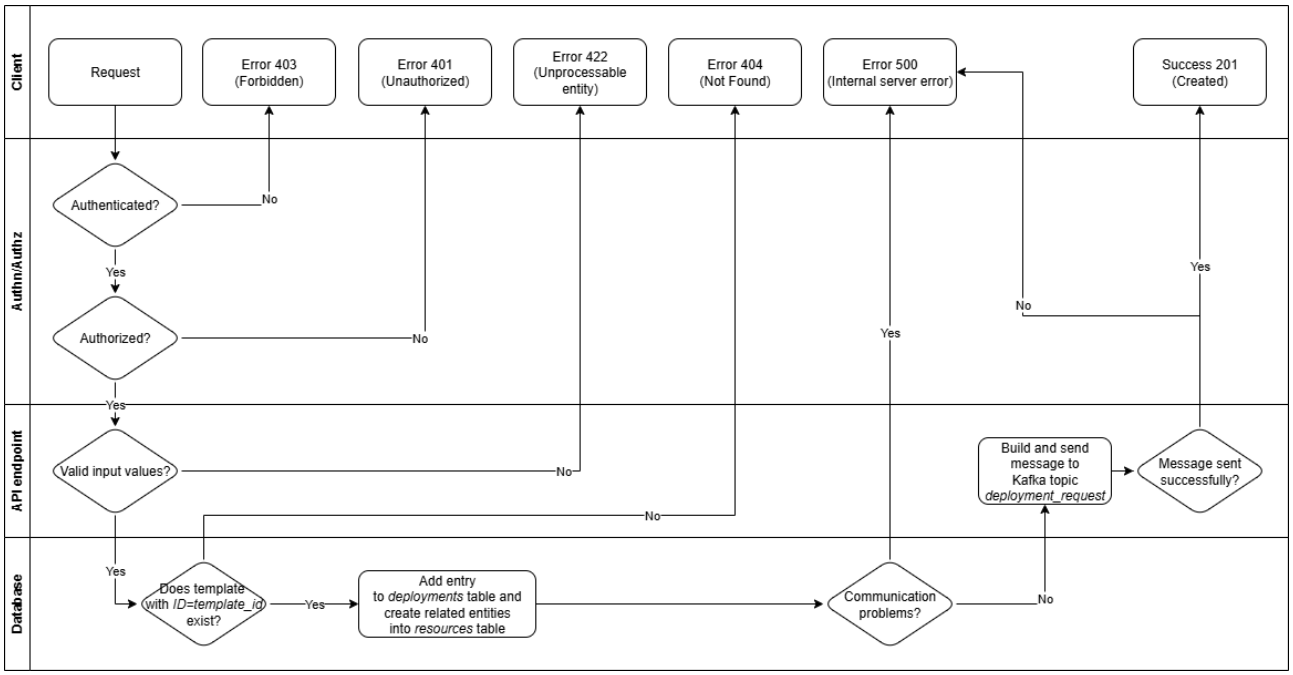
### *Response from the server example*

```
< HTTP/1.1 204 No Content
< Connection: Keep-Alive
< Content-Length: 0
< Date: Tue, 01 Mar 2022 11:30:06 GMT
```

# Deployments

A Deployment represents a user request to use resources on an available cloud provider. A deployment is issued by authenticated user. A (TOSCA) template defines the resources topology required by a specific deployment. A deployment requires a minimal set of resources specified by the user (or though the template default values) and these belong to one of the issuer's available resource providers.

## New deployment submission




### *Description*

The POST request creates a deployment. A deployment is defined by a (TOSCA) template. It specifies the minimum required resources.

If not specified, the Orchestrator will choose the resource provider where to deploy the request. The user can specify the resource provider and the region to use.

A deployment MUST be issued by a user belonging to an authorized user group. The user will be the creator of that deployment request. Since the new orchestrator will support multiple identity providers, the server retrieves the identity providers from the access token.

When a new request is sent to this endpoint a new message, containing the request's body, is sent to kafka.

### *Submission*

| | |
|---|---|
| **Method** | POST |
| **Target** | /api/v1/deployments |
| **Content-Type** | application/json |
| **Possible response codes** | 202, 400, 401, 403, 422, 500 |

### *Request's body fields description*

| Name | Description | Type | Optional |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **template_id** | ID of the deployment | String | No |
| **template_inputs** | The minimum required resources for the deployment. Values changes based on the deployment type | JSON object | No |
| **user_group** | The user group for which the deployment will be created | String | No |
| **per_provider_max_retries** | Maximum number of retries for each provider. In range [1,10]. | Integer | Yes. Default is 3. |
| **max_providers** | The maximum number of cloud providers on which attempt to create the deployment. In range [1, +inf). | Integer | Yes. Default is None (=No limits). |
| **tot_timeout_mins** | Overall timeout value in minutes. If provided, it must be greater than 0. In range [1,14400] | Integer | Yes. Default is 14400 mins. |
| **per_provider_timeout_mins** | Timeout value for a single provider (it does not apply to single tries but it is the overall timeout for a provider). If provided, it must be greater than 0 and equal or lower than *tot_timeout_mins* | Integer | Yes. Default is 1440 mins. |
| **keep_last_attempt** | Whether the IM, in case of failure, will keep the resources of the last attempted deployment or not. | Boolean | Yes. Default is false. |
| **target_provider** | Name of the target provider to use. | String | Yes. Default is None. |
| **target_region** | Name of the region to use. | String | Yes. Default is None. |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments' -i -X POST -H
'Content-Type: application/json' -H 'Authorization: Bearer <access token>' -d '{
  "template": "template in YAML format",
  "template_inputs": {
    "cpus": 1,
    ...
  },
  "user_group": "beta-testers",
  "per_provider_max_retries": 2,
  "max_providers": 3,
  "per_provider_timeout_mins": 10,
  "tot_timeout_mins": 25,
  "keep_last_attempt": false,
  "target_provider”: "backbone",
  "target_region”: "cnaf"
}'

POST /deployments HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: example.com
Content-Length: 256

{
  "template": "template in YAML format",
  "template_inputs": {
```

```
      "cpus": 1,
        ...
    },
    "user_group": "beta-testers",
    "per_provider_max_retries": 2,
    "max_providers": 3,
    "per_provider_timeout_mins": 10,
    "tot_timeout_mins": 25,
    "keep_last_attempt": false,
    "target_provider”: "backbone",
    "target_region”: "cnaf"
}
```

*Response fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **id** | Entity unique identifier | UUID String | No |

*Response from server example*

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 51

{
  "id": "3a1cbf3e-0d70-4036-95b4-46b3f5c15cec",
}
```

*Breaking changes*

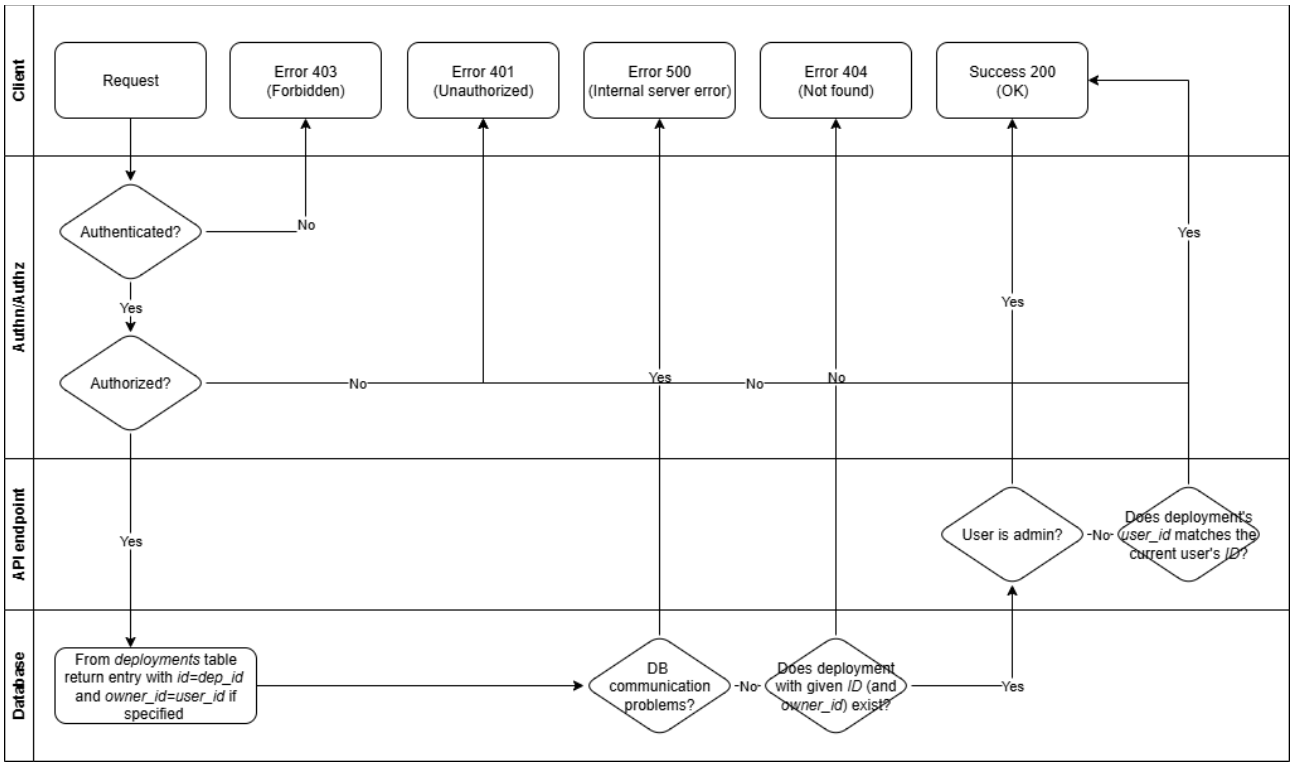
Submission

| Java Orchestrator field | Python Orchestrator field | Notes |
|---|---|---|
| **template** | template_id | Instead of the complete TOSCA template content, point to the ID of the TOSCA template stored in the DB |
| **userGroup** | user_group | |
| **maxProvidersRetry** | max_providers | |
| **timeoutMins** | tot_timeout_mins | |
| **providerTimoutMins** | per_provider_timeout_mins | |
| **keepLastAttempt** | keep_last_attempt | |
| **parameters** | template_inputs | |
| **callback** | - | Removed because the REST API does know nothing about interfaces or who is using the service. The clients will poll for updates. |

The response returns only the deployment uuid. The deployment key is no more *uuid* but *id*.

With Orchent depcreate must execute the POST on templates to create the template if it does not already exist, otherwise retrieves the template ID and does the POST on deployments with the correct template ID.

## Retrieve a Deployment




### *Description*

The GET request retrieves a deployment, by the deployment's id.

Administrators can retrieve any deployment; standard users can retrieve only their own deployments.

### *Submission*

| | |
|---|---|
| **Method** | GET |
| **Target** | /api/v1/deployments/{id} |
| **Content-Type** | application/json |
| **Possible response codes** | 200, 401, 403, 404, 422, 500 |

### *Request fields description*

Nothing

### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-ebe196bf82dd' -i -H 'Authorization: Bearer <access token>'

GET /deployments/34483-d937-4578-bfdb-ebe196bf82dd HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

### *Response fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **id** | Unique identifier | UUID String | No |
| **created_at** | Creation timestamp. | String ISO 8601 | No |

| | | | |
|---|---|---|---|
| **created_by** | ID of the user which submitted the request | String | No |
| **updated_at** | Update timestamp. If no changes have been made it matches the *created_at* field. | String ISO 8601 | No |
| **updated_by** | ID of the user which submitted the latest update to this deployment. If no changes have been made it matches the *created_by* field. | String | No |
| **owned_by** | List of users (IDs) which ownes the request. By default, it matches the *created_by* field. Each owner will have its public key installed on the VM and its home. | List of String | No |
| **template_id** | ID of the used TOSCA template | UUID String | No |
| **template_inputs** | The minimum required resources for the deployment. Values changes based on the deployment type | JSON object | No |
| **template_outputs** | The outputs of the template. Until the procedure ends, it is an empty dict. | JSON object | No |
| **user_group** | The user group the deployment has been created for. | String | No |
| **per_provider_max_retries** | Maximum number of retries for each provider | Integer | No |
| **max_providers** | The maximum number of cloud providers on which attempt to create the deployment. Can be None (No imposed limits) | Integer | No |
| **tot_timeout_mins** | Overall timeout value in minutes. If provided, it must be greater than 0 | Integer | No |
| **per_provider_timeout_mins** | Timeout value for a single provider. If provided. | Integer | No |
| **keep_last_attempt** | Whether the Orchestrator, in case of failure, will keep the resources of the last attempted deployment or not | Boolean | No |
| **target_provider** | Name of the IaaS to use. Can be None if the user does not specify it. Then the orchestrator will populate it with the chosen provider. | String | No |
| **target_region** | Name of the region to use. Can be None if the user does not specify it. Then the | String | No |

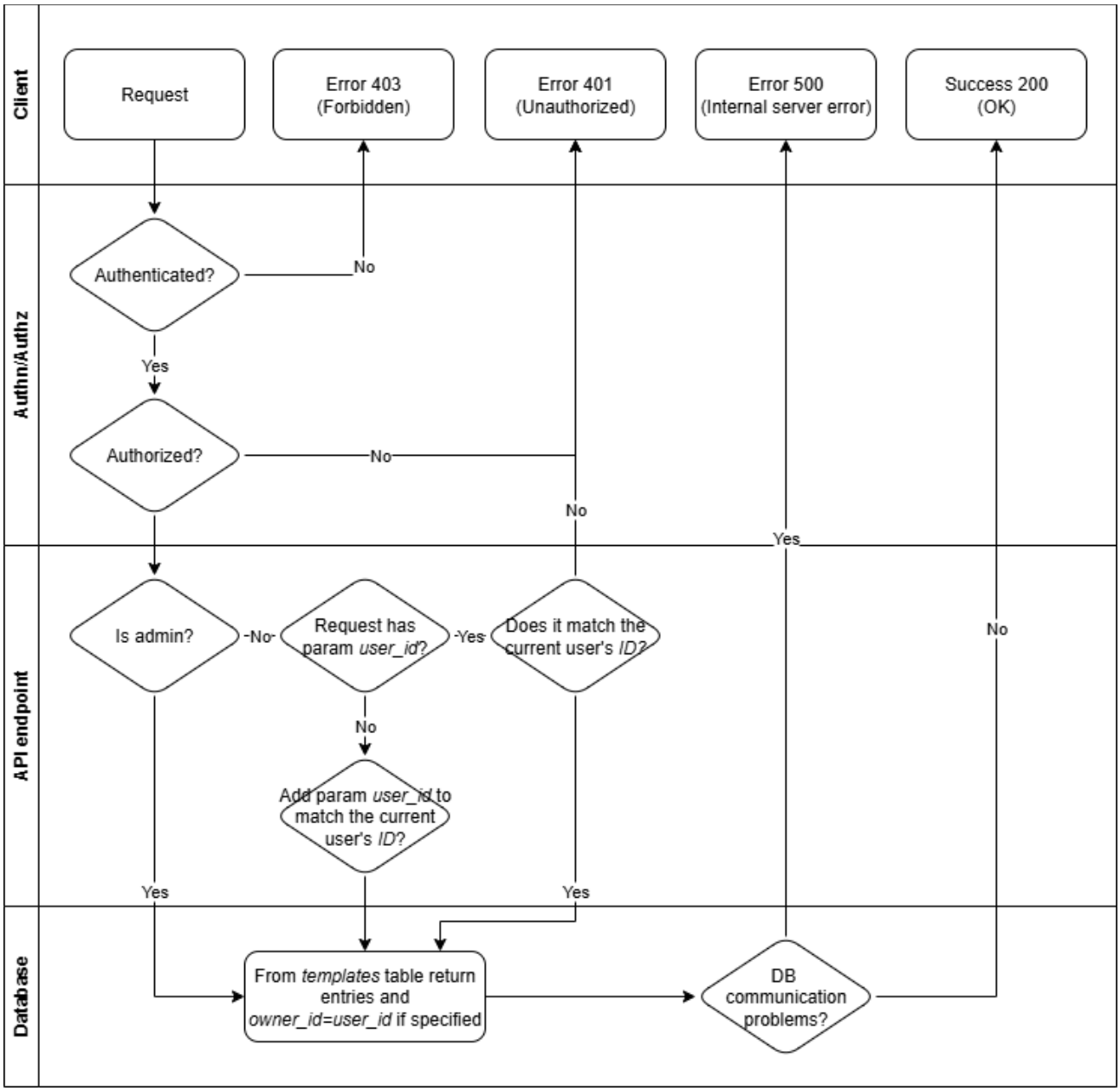
| | | | |
|---|---|---|---|
| | orchestrator will populate it with the chosen provider. | | |
| **target_iaas_project** | IaaS tenant/namespace chosen by the orchestrator | String | No |
| **im_infra_id** | ID of the corresponding infrastructure in the IM DB. Initially it is None. | UUID String | No |
| **status** | The status (code) of the deployment:<br>0. CREATE_COMPLETE: Creation procedure ended successfully.<br>1. CREATE_FAILED: Creation procedure ended with an error.<br>2. CREATE_IN_PROGRESS: Creation procedure on going.<br>3. DELETE_COMPLETE: Deletion procedure ended successfully.<br>4. DELETE_FAILED: Deletion procedure ended with an error.<br>5. DELETE_IN_PROGRESS: Deletion procedure on going.<br>6. UNKNOWN: Unknown state<br>7. UPDATE_COMPLETE: Update procedure ended successfully.<br>8. UPDATE_FAILED: Update procedure ended with an error.<br>9. UPDATE_IN_PROGRESS: Update procedure on going. | Integer | No |
| **status_name** | Human readable status string | String | No |
| **status_reason** | Verbose explanation of reason that led to the deployment status. Can be None. It is not None only if the deployment is in an erroneous status. | String | No |
| **task** | The current step (code) of the deployment process. | String | No |
| **task_name** | Human readable step string | String | No |
| **links** | JSON dict with the links pointing to its sub-resources | JSON object | No |
| **links.templates** | A string containing a TOSCA YAML-formatted template. Value returned only when | String | Yes. |

| | | | |
|---|---|---|---|
| | request parameters *template* is **true**. | | |
| **links.resources** | Endpoint to contact to retrieve the deployment generated resources | String URI format | No |
| **links.logs** | Deployment logs retrieved from the IM. Can be None. | String | No |

*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 722

{
  "uuid": "34483-d937-4578-bfdb-ebe196bf82dd",
  "created_at": "2024-01-01T10:00:00.000000Z",
  "created_by": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "updated_at": "2024-01-01T10:00:00.000000Z",
  "updated_by": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "owned_by": [
    "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  ],
  "template_id": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "template_inputs": {
    "cpus": 1,
    ...
  },
  "template_outputs": {
    "server_ip": "10.0.0.1"
    ...
  },
  "user_group": "beta-testers"
  "per_provider_max_retries": 2,
  "max_providers": 3,
  "per_provider_timeout_mins": 10,
  "tot_timeout_mins": 25,
  "keep_last_attempt": false,
  "target_provider": "backbone",
  "target_region": "cnaf",
  "target_iaas_project": "test-project",
  "status": 2,
  "status_name": "CREATE_IN_PROGRESS",
  "status_reason": null,
  "task": 0,
  "task_name": "submitted",
  "links": {
    "template": "http://example.com/deployments/3a1cbf3e-0d70-4036-95b4-
46b3f5c15cec/templates",
    "resources": "http://example.com/deployments/3a1cbf3e-0d70-4036-95b4-
46b3f5c15cec/resources",
    "logs": "http://orchestrator-api/deployments/3a1cbf3e-0d70-4036-95b4-
46b3f5c15cec/logs",
  }
}
```

## Retrieve accessible deployments




### *Description*

The GET request retrieves all deployments and their associated resources.

Clients can filter requests for any of the read attributes.

Administrators can retrieve all deployments; standard users can retrieve only their own deployments. If the user does not fill the *owned_by* parameter, the application automatically returns only its deployment. If a user tries to retrieve deployments belonging to another user, it receives an error message.

Parameters to filter deployments based on users can accept the **me** value. It is shortcut to ask for the deployments created by the user making the request.

### *Submission*

| **Method** | GET |
|---|---|

| | |
|---|---|
| **Target** | /api/v1/deployments |
| **Content-Type** | application/json |
| **Params** | Any of the deployment entity's attributes + page={int}&size={int}&sort={str} |
| **Possible response codes** | 200, 400, 401, 403, 422, 500 |

*Request fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **status** | List of status. Only deployments with status in this list will be returned. | List of string | Yes |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments' -i -H 'Accept:
application/json' -H 'Authorization: Bearer <access token>'

GET /deployments HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

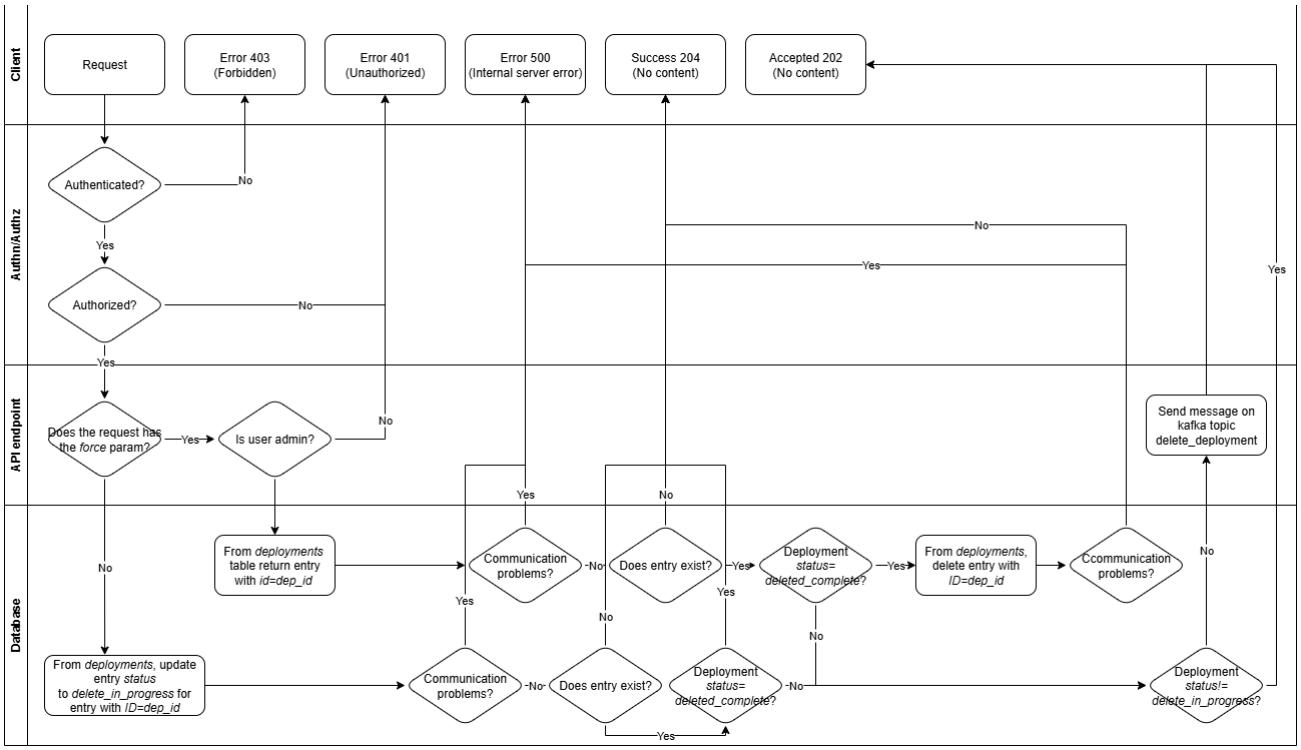
*Response fields description*

See Pagination for main fields. The *data* field contains the list of deployments. Each item has the shape of the response returned by Retrieve accessible deployments.

*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2059

{
  "data": [List of Deployments],
  "links": {
    "first": "http://example.com/api/v1/deployments?page=1&size=5",
    "prev": "http://example.com/api/v1/deployments?&page=1&size=5",
    "next": "http://example.com/api/v1/deployments?page=3&size=5",
    "last": "http://example.com/api/v1/deployments?page=10&size=5",
  }
  "page": {
    "size": 5,
    "total_elements": 50,
    "total_pages": 10
    "number": 3
  }
}
```

## Delete a deployment




### *Description*

The DELETE request deletes deployment by the deployment's id. In this case, the deletion frees resources from the resource provider but does not delete entry from the local database.

Administrators can delete any deployment; standard users can delete only owned deployments.

Administrators can pass the attribute *force* to forcibly delete the entry from the database.

Administrators can pass the attribute *unsecure* to ignore errors or failures during the deletion of the resources related to a specific deployment.

### *Submission*

| | |
|---|---|
| **Method** | DELETE |
| **Target** | /api/v1/deployments/{id} |
| **Content-Type** | application/json |
| **Params** | force={bool}&unsecure={bool} |
| **Possible response codes** | 202, 204, 400, 401, 403, 404, 500 |

### *Request fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **force** | Parameter to forcibly delete from the DB the entry. | Bool | Yes. Default is false. |
| **unsecure** | Parameter used to forward the force command to the IM and used to mark as DELETED also resources which have not been successfully deleted | Bool | Yes. Default is false. |

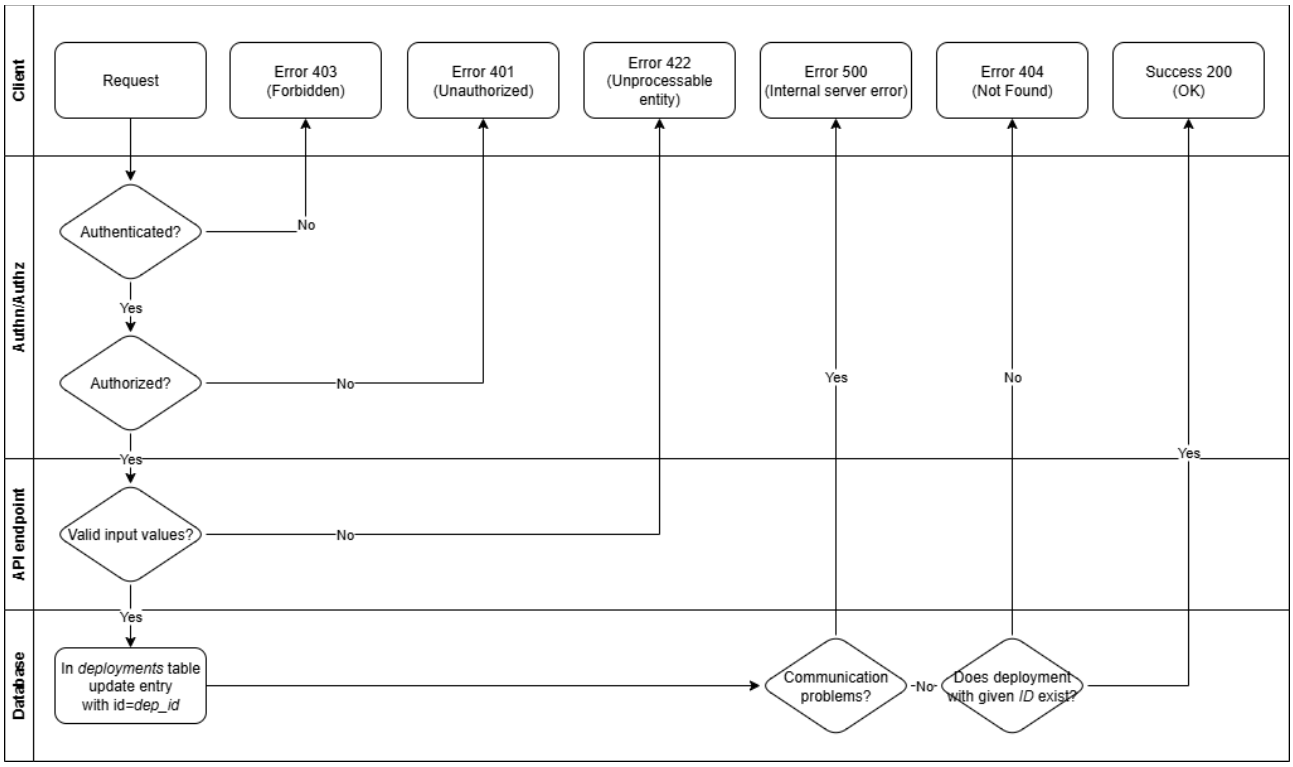
*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd' -i -X DELETE -H 'Authorization: Bearer <access token>'

DELETE /deployments/34483-d937-4578-bfdb-ebe196bf82dd HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

*Response from server example*

```
HTTP/1.1 202 Accepted

HTTP/1.1 204 No Content
```

## Update without re-deploying a deployment by ID (Admin only)




### *Description*

Update the user_group name in case or iam renaming. Administrators only.

### *Submission*

| | |
|---|---|
| **Method** | PATCH |
| **Target** | /api/v1/deployments/{id} |
| **Content-Type** | application/json |
| **Possible response codes** | 202, 204, 400, 401, 403, 404, 500 |

### *Request's body fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **user_group** | The user group the deployment has been created for. | String | No |

### *Request submission example*

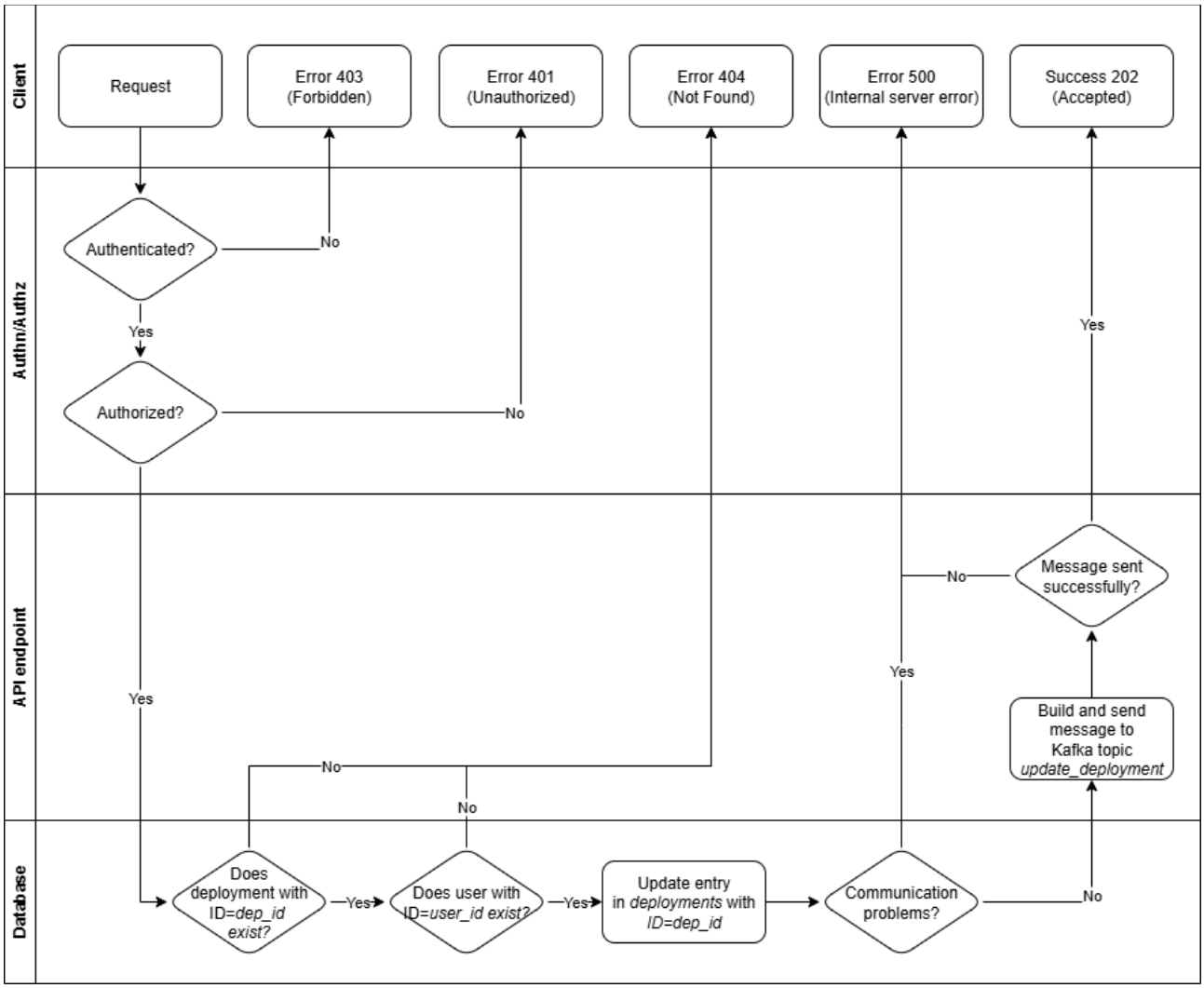
```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd' -i -X PATCH -H 'Content-Type: application/json' -H 'Authorization:
Bearer <access token>' -d '{
  "user_group": "admins/catch-all"
}'

PATCH /deployments/34483-d937-4578-bfdb-ebe196bf82dd HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
Content-Length: 224

{
```

```
  "user_group": "admins/catch-all"
}
```

*Response from server example*

```
HTTP/1.1 204 No content
```

## Add a deployment owner




### *Description*

The POST request can be used by administrators to add the ownership of a deployment request to another user changing the *owned_by* field. If the user already owns a deployment request, the endpoint does nothing, otherwise the orchestrator adds the new owner public key to the target resources. Consequently, the deployment status changes from CREATE_COMPLETE to UPDATE_IN_PROGRESS and returns to CREATE_COMPLETE once the operation ends.

### *Submission*

| | |
|---|---|
| **Method** | POST |
| **Target** | /api/v1/deployments/{dep-id}/owners |
| **Content-Type** | application/json |
| **Possible response codes** | 202, 400, 401, 403, 404, 422, 500 |

### *Request's body fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **id** | ID of the user to add to the deployment owners | String | Yes |

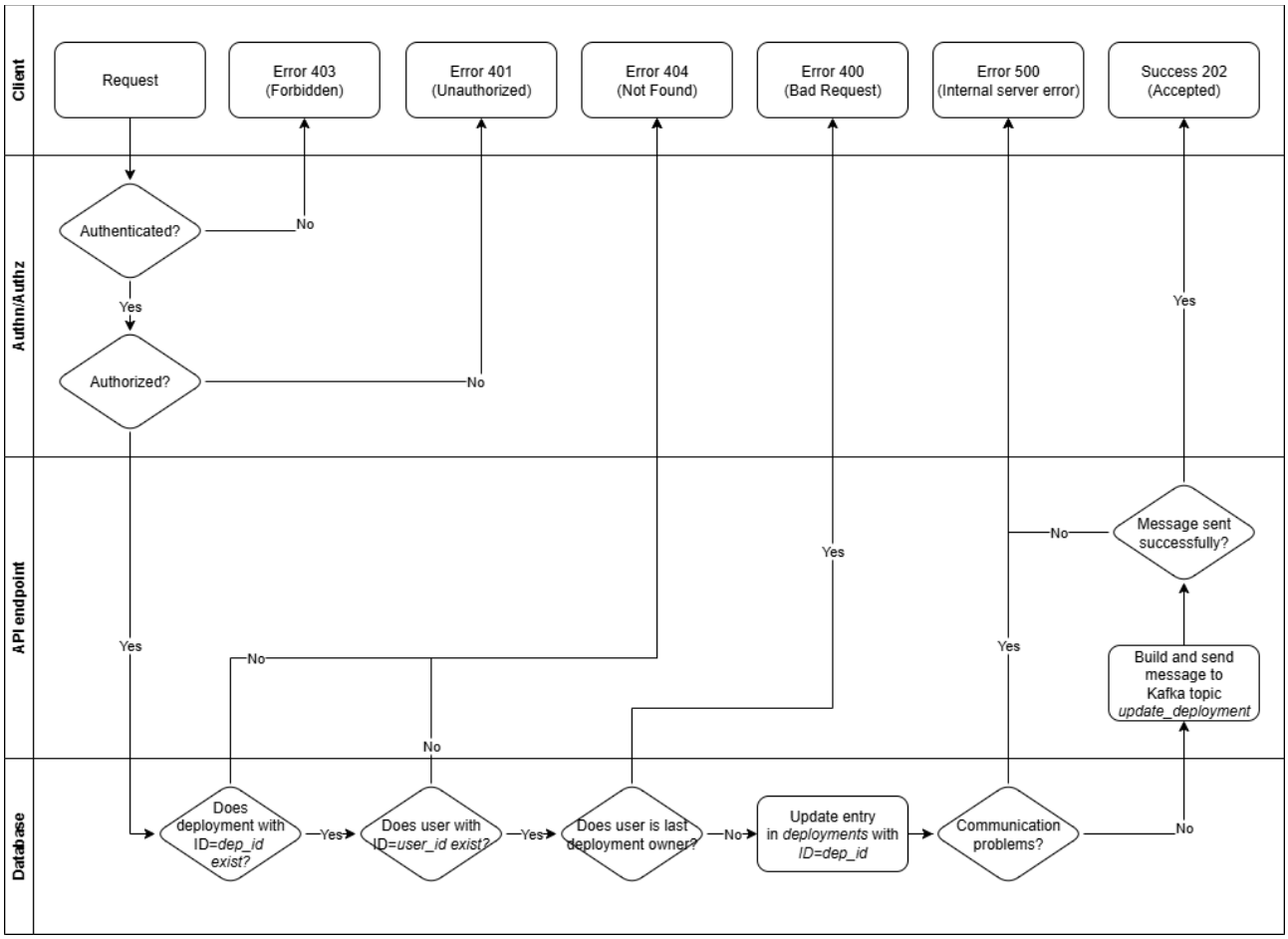
*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/owners' -i -X POST -H 'Content-Type: application/json' -H
'Authorization: Bearer <access token>' -d '{
  "id": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd"
}'

POST /deployments/34483-d937-4578-bfdb-ebe196bf82dd/owners HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
Content-Length: 224

{
  "id": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd"
}
```

*Response from server example*

```
HTTP/1.1 202 Accepted
```

## Remove a deployment owner




### *Description*

The DELETE request can be used by administrators to remove the ownership of a deployment request from a user changing the *owned_by* field. If the user is not an owner of the deployment request, the endpoint does nothing, otherwise the orchestrator removes the owner public key from the target resources. Consequently, the deployment status changes from CREATE_COMPLETE to UPDATE_IN_PROGRESS and returns to CREATE_COMPLETE once the operation ends.

If the user to remove is the unique owner of the deployment, the operation fails.

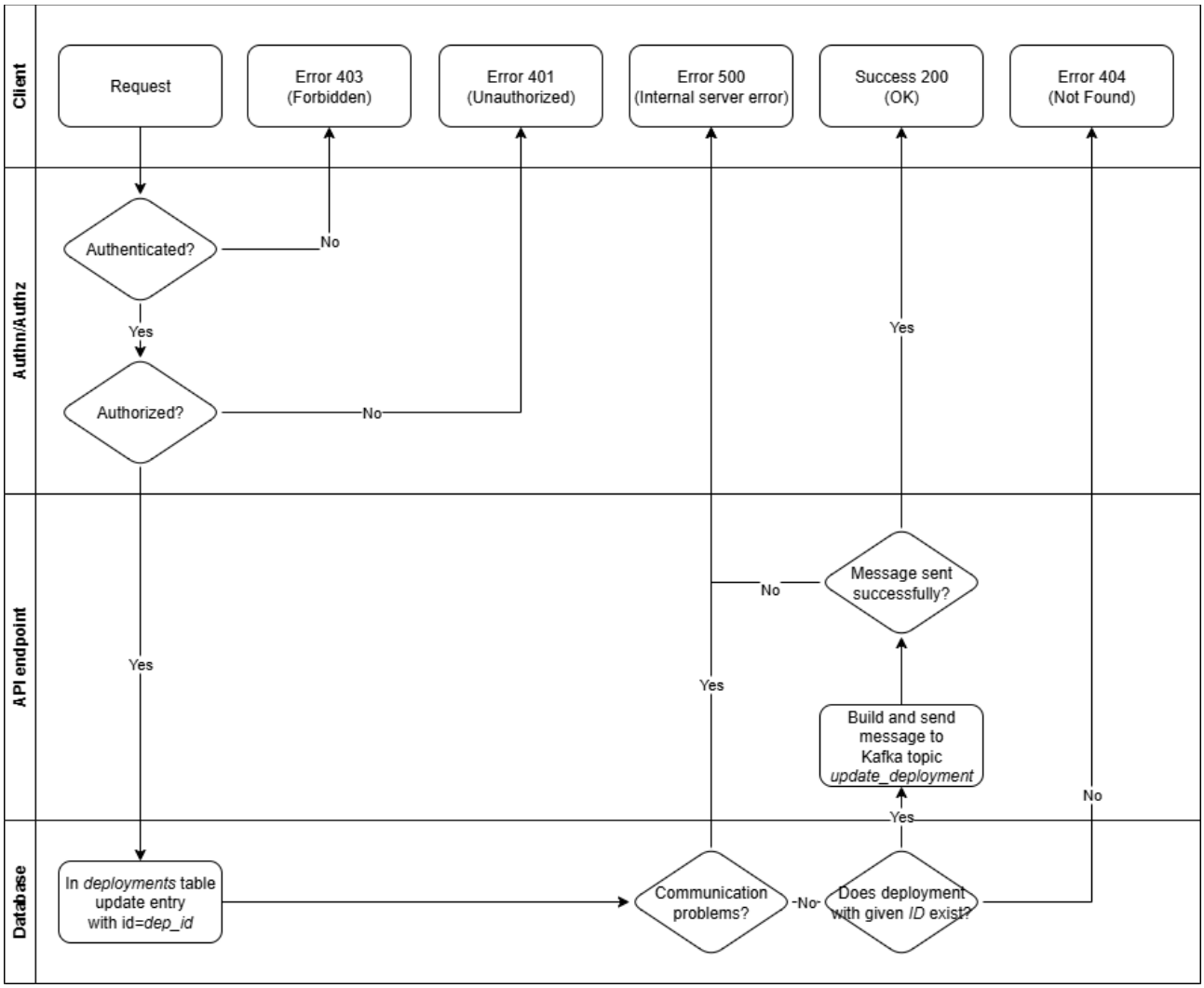
### *Submission*

| | |
|---|---|
| **Method** | DELETE |
| **Target** | /api/v1/deployments/{dep-id}/owners/{user-id} |
| **Content-Type** | application/json |
| **Possible response codes** | 202, 400, 401, 403, 404, 409, 422, 500 |

### *Response from server example*

```
HTTP/1.1 202 Accepted
```

## Reset deployment status




### *Description*

The POST request can be used to reset the deployment state given the deployment id. (e.g. If a deployment remains stuck in *DELETE_IN_PROGRESS*, clients can use this endpoint to manually reset the state of the deployment to the *DELETE_FAILED* state. You can then try to delete the deployment again). Similarly, from CREATE_IN_PROGRESS reset to CREATE_FAILED and from UPDATE_IN_PROGRESS reset to UPDATE_FAILED.

### *Submission*

| | |
|---|---|
| **Method** | POST |
| **Target** | /api/v1/deployments/{dep-id}/reset-status |
| **Content-Type** | application/json |
| **Possible response codes** | 202, 400, 401, 403, 404, 422, 500 |

### *Request's body fields description*

No body.

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/owners' -i -X POST -H 'Content-Type: application/json' -H
'Authorization: Bearer <access token>' -d '{
  "id": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd"
}'

POST /deployments/34483-d937-4578-bfdb-ebe196bf82dd/owners HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
Content-Length: 224
```

*Response from server example*

HTTP/1.1 202 Accepted

## Update and re-deploy a deployment by ID

*Description*

The PATCH request can re-deploy an already submitted and completed deployment request.

The updating of the input values with the specified ones triggers the re-deployment of the request on the same provider. It is usually used when there is the need to update the resources allocated to the resource provider (e.g. increase/decrease the number of nodes in a k8s cluster) or the deployment configuration (e.g. change security groups).

This operation should not be used to re-deploy a deployment on another resource provider. Clients must create a completely new instance instead of updating the existing one.

If the deployment status is *CREATE_IN_PROGRESS* or in *UPDATE_IN_PROGRESS* or in *DELETE_IN_PROGRESS* or *DELETE_COMPLETED*, the server discards the update request.

*Submission*

| | |
|---|---|
| **Method** | PATCH |
| **Target** | /api/v1/deployments/{id} |
| **Content-Type** | application/json |
| **Possbile response codes** | 202, 204, 400, 401, 403, 404, 500 |

*Request's body fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **template_inputs** | The new value for the template inputs. Only the ones to be overwritten. Not defined ones will keep the previous value. | JSON object | Yes |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/inputs' -i -X PATCH -H 'Content-Type: application/json' -H
'Authorization: Bearer <access token>' -d '{
  "cpus": 1,
  ...
}'

PATCH /deployments/34483-d937-4578-bfdb-ebe196bf82dd HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
Content-Length: 224

{
  "cpus": 1,
  ...
}
```

*Response from server example*

```
HTTP/1.1 202 Accepted
```

or

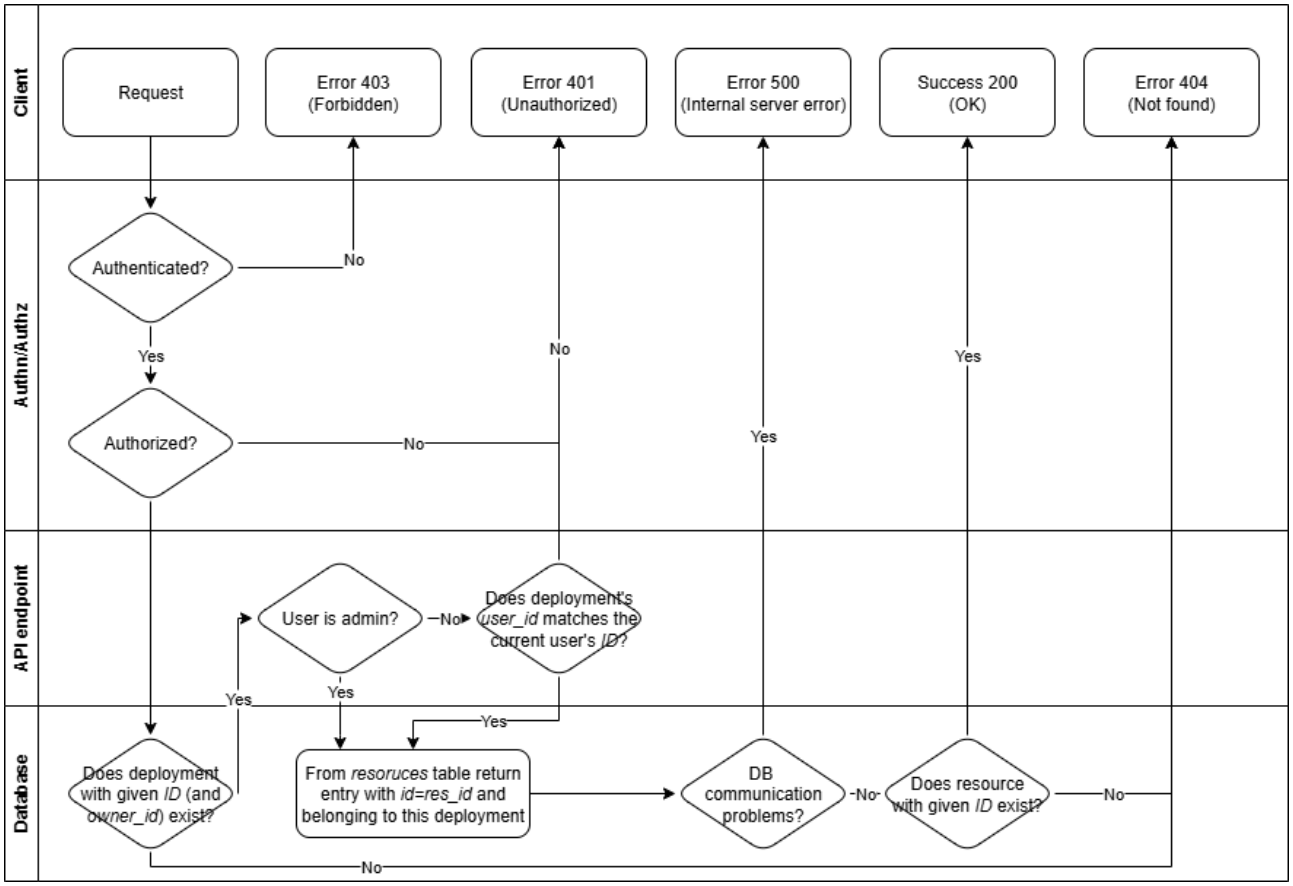
```
HTTP/1.1 204 No Content
```

# Deployments/Resources

A Resource represents a (TOSCA) node of the template. A resource always belongs to a deployment instance.

Non esiste la POST perchè i nodi vengono creati internamente dalla POST /deployments quando il deployment viene creato facendo il parsing del TOSCA template.

## Retrieve a resource




### *Description*

The GET request is used to retrieve a specific resource (TOSCA node) allocated for a specific deployment.

A standard user can retrieve only resources belonging to owned deployments. Administrators can see any resource.

### *Submission*

| | |
|---|---|
| **Method** | GET |
| **Target** | /api/v1/deployments/{dep-id}/resources/{id} |
| **Content-Type** | application/json |
| **Possbile response codes** | 200, 400, 401, 403, 404, 500 |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/173aaf8b-0cbd-4309-
b30c-fb160024f5cc/resources/9c77b87f-18f7-4cfb-8782-f7052932b69f' -i -H
'Authorization: Bearer <access token>'

GET /deployments/173aaf8b-0cbd-4309-b30c-fb160024f5cc/resources/9c77b87f-18f7-4cfb-
8782-f7052932b69f HTTP/1.1
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

*Response fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **uuid** | Unique identifier of the resource | String | No |
| **created_at** | Creation timestamp | String ISO 8601 | No |
| **created_by** | ID of the user which submitted the request | String | No |
| **updated_at** | Last update timestamp | String ISO 8601 | No |
| **updated_by** | ID of the user which submitted the latest update to this deployment. If no changes have been made it matches the *created_by* field. | String | No |
| **im_vm_idx** | The index of the VM created by the IM. Corresponds to the *im_id* field in the *data.vms_list* of the deployment corresponding infrastructure entry in the IM DB. It is not None only for Compute TOSCA node types. Can be None if the resource has not been assigned yet or a corresponding item does not exist in the IM DB (for example Network, Port, ...). | String | No |
| **status** | Resource status (code):<br>0. CONFIGURED: Node has been configured prior to being started.<br>1. CONFIGURING: Node is transitioning from *CREATED* state to *CONFIGURED* state.<br>2. CREATED: Node software has been installed.<br>3. CREATING: Node is transitioning from *INITIAL* state to *CREATED* state.<br>4. DELETED: Node is deleted.<br>5. DELETING: Node is transitioning from its current state to *DELETED*.<br>6. ERROR: Node is in an error state.<br>7. INITIAL: Node is not yet created.<br>8. STARTED: Node is started.<br>9. STARTING: Node is transitioning from *CONFIGURED* or *STOPPED* state to *STARTED* state. | String | No |

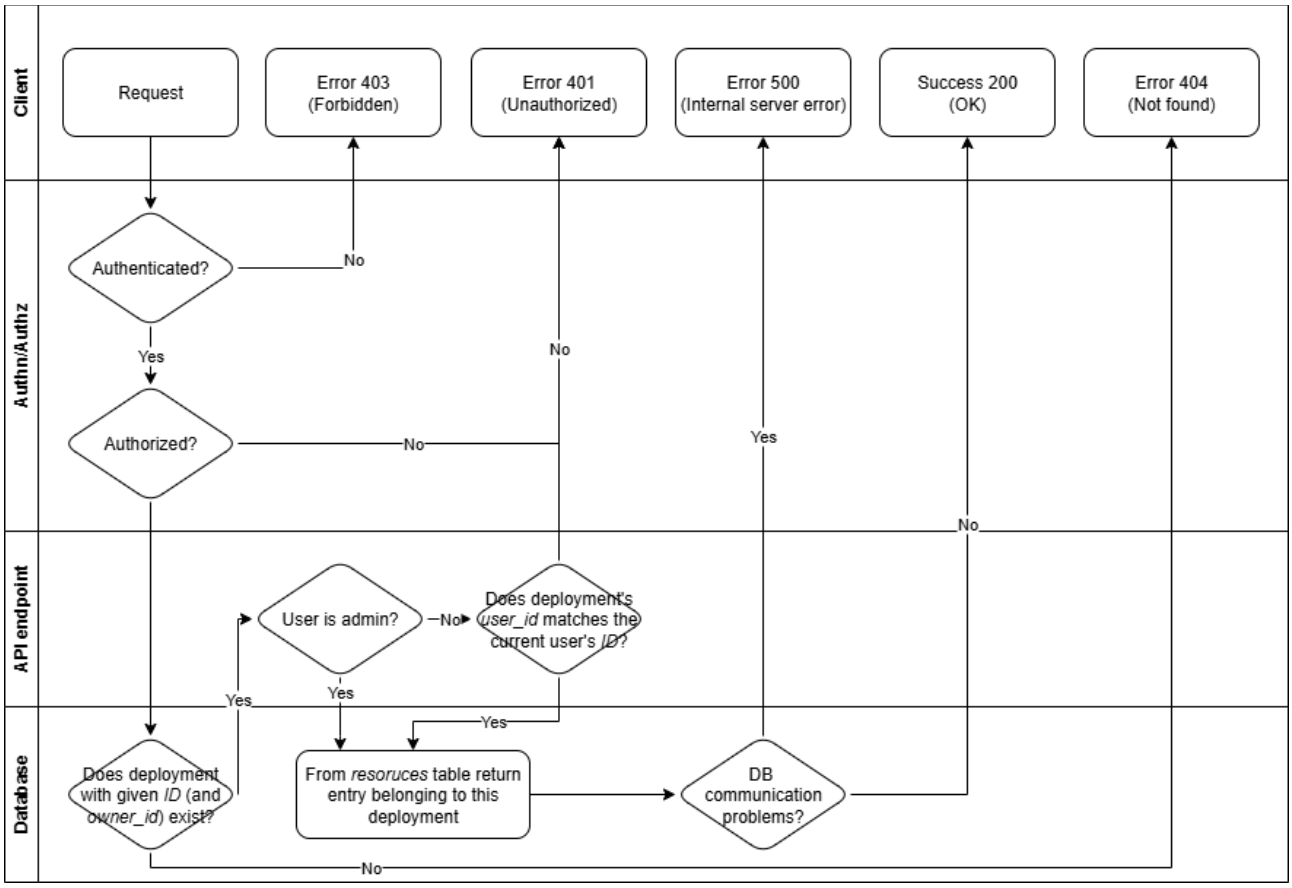
| | | | |
|---|---|---|---|
| | 10. STOPPED: Node is stopped.<br>11. STOPPING: Node is transitioning from its current state to a stopped state. | | |
| **status_name** | Human readable status string | String | No |
| **tosca_node_type** | The type of the represented TOSCA node. | String | No |
| **tosca_node_name** | The name of the TOSCA node. | String | No |
| **required_by** | List of nodes that require this resource. Can be an empty list. | List of strings | No |
| **metadata** | Additional information. They are the content field *info* of the corresponding item in the *data.vms_list* field. It is None for entries not linked to an infrastructure or not infrastructures not yet configured. They are also the source for the template output generation. | JSON object | No |

*Response from server example*


```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 665

{
  "uuid": "9c77b87f-18f7-4cfb-8782-f7052932b69f",
  "created_at": "2024-01-01T10:00:00.000000Z",
  "created_by": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "updated_at": "2024-01-01T10:00:00.000000Z",
  "updated_by": "1cb7eaef-6eb7-454a-ab38-c785fd6d08bd",
  "im_vm_idx": "0",
  "status": 0,
  "status_name": "CREATING",
  "type": "tosca.nodes.Compute",
  "name": "node_e336a04a-8f90-417a-acaf-baaec15c531b",
  "required_by": [ ],
  "metadata": {
    "VirtualMachineInfo":
"{\"vmProperties\":[{\"class\":\"network\",\"id\":\"pub_network\",\"outbound\":\"yes\
",\"outports\":\"22/tcp-
22/tcp\",\"provider_id\":\"public\"},{\"class\":\"network\",\"id\":\"priv_network\",\
"provider_id\":\"INFN_Cloud_HSMDIS-
net\"},{\"class\":\"system\",\"id\":\"simple_node\",\"instance_name\":\"simple-node-
85d7d822-8ab3-11ed-b5a4-
fa163ed94b64\",\"instance_tags\":\"PAAS_URL=http://localhost:8080,PAAS_DEP_UUID=11ed8
ab3-6a40-968a-82b7-
62f0862dbe46,PAAS_DEP_USER_EMAIL=Marica.Antonacci@ba.infn.it\",\"disk.0.os.flavour\":
\"ubuntu\",\"disk.0.os.version\":\"20.04\",\"disk.0.image.url\":\"ost://keystone.clou
d.infn.it/2b25c2c6-6398-45c6-9e00-
3e7c58dbd750\",\"cpu.count\":1,\"memory.size\":2147483648,\"instance_type\":\"small\"
,\"net_interface.1.connection\":\"pub_network\",\"net_interface.0.connection\":\"priv
_network\",\"cpu.arch\":\"x86_64\",\"disk.0.free_size\":10737418240,\"disk.0.os.crede
ntials.username\":\"cloudadm\",\"provider.type\":\"OpenStack\",\"provider.host\":\"ke
ystone.cloud.infn.it\",\"disk.0.os.credentials.private_key\":\"-----BEGIN RSA PRIVATE
KEY-----
\\nMIIEowIBAAKCAQEAxTIykFKQKLinzT8QqeJplp2VwegWZGCs07Yn9Su/veJYKdpq\\ny0S/6LKyNikCwYG
u2o6ZRIEtj7Per3SexalK94fGLok+HwBkoLLYP3DJu93dgarB\\nM8ze/71gQhEj1YqhWj8ff31/M40/Wf7T9
1fv0FUNj/aBnpQS83UKDdUp5m5criRt\\n5s7Ji3vexmZN2a6peZytbzvUChlzJKH+vVaWAkte+H5QK7kVldy
RjSiOsmArk4ng\\n4xoJtW3KvVDJTR5XHgmJgPmQv7lHGZVMSxcy9tUyZxGJ7Vj4wUIqR86VadgQ8Uez\\nR/
khvIIB/yNyKkuLMOMc5MhFq/FkEctqXvZKvQIDAQABAoIBADQKFjvt/esxJ99L\\nSEt8256Kwa2lYNtmg9Vv
```

```
GcZioDwtElUhNdOktfxjxBH06qxEW++DeEES0gI9VoP5\\nA7ln/seqBgvb3g/yW5vo9pZvXl18pGsGs+vGog
bdZByrR0igSAp35AkSBlKFcPWA\\nAVIh28Cf4W9ffz9pFkM3XMrfuSn8+TF21l2bZgRFxp2BnxUR7wm9jf9+
IegaHYsN\\nHMPCslV1TMX3qFQsVNh/2OUAa/cER2ADmt6uF2QfpySXdPdvhvo8TKx3APqBgHEd\\nL7xXKAC
vsEb/X/sshC7G7QgZ6Hug/w0Dt2Nb57WbSKJMp5BNPgoPHlcTjFI2+Xf9\\n19tEbmUCgYEA1jsBxkuJs1/bN
6btGR2QQ8B4sJ6b+W6QS2KvEl0VI8b86A+qE2/U\\nVonRbV506dqhuLoIOg6vUjI7ir69IWD+k1a7O3RS8es
gCopWzp7GA9gACC0nEzeB\\nF9ZAeku35V44+CEyAS+PJ/rIU0wTYVmHmyvYNvuYZW6RSGbNk1Yk/QsCgYEA6
6Tx\\ni1fOVgQjk+93mUJyQ4GBijPeo1GZgH4wzFx5bQTilZcKtD9eoBlAXQYy6WYin16e\\n7oLuv9P4T6a7
qEtDiPWDdcgIoeqhsHMnzT2dZPFWBD/AxFLetjmLUabBlSCiF3c3\\nmd0J+BOYWEbmiIya7lt/ntgA6xBBMQ
NLpDAkZFcCgYEAthaBc6JQ8H3RPoRLksVm\\nEbBO5RBrUp9/VhQS6nDjvv0ribOGw+6RdNy/fU6D4SFLxE+b
Hn3M1viLU2q8Jtri\\nhyl5CxfcSHOpUEw1bHnXpPy7SjfWtqa+uxwIweoJ7JMhqQXOEuPxsfGew+4tqtkG\\
nQgYKSPGgntZT9k4q/ciCt/ECgYBv8fYF+nq9hk3Yk6S/nD9oLnf5zdZww+0mi8xm\\n6W+OCQoTgqPFKW2sp
RJcnvtEwg6ko3DhYjZFz09hl5YCx26X+/qt04+2TB7wEmTQ\\nLfs93yC7+AV9vtiqvP+glJXWYKBggXRalhW
MUtLG7TfLmvBnV/Ry467b/Obbf/NJ\\nVs2TFQKBgH/VMnMItvJZNouemf0OxePr5nsQKc9eQvRlcXqwhvaHg
FjPVzvhUKbN\\nEHSf2PpegkPodrqHgW4l9QezO2xHNkx4LMbHZE1R6dmzi6YYkEar9JuYyzvpwpC/\\nnXxK
1hrvni/13VoWc2zZUn0YcEi8HXVBby7Y8hIkrg9GAgVzbt2k\\n-----END RSA PRIVATE KEY-----
\\n\",\"state\":\"stopped\",\"instance_id\":\"38e9014e-a303-4f40-92a2-
fae26e1fddfa\",\"net_interface.0.dns_name\":\"vnode-
0\",\"net_interface.0.ip\":\"192.168.163.69\",\"net_interface.1.ip\":\"90.147.174.56\
"}]}"
  },
}
```

## Retrieve accessible resources




*Description*

The GET request is used to retrieve all the resources (TOSCA nodes) allocated for a specific deployment.

A standard user can retrieve only resources belonging to owned deployments. Administrators can see any resource.

*Submission*

| | |
|---|---|
| **Method** | GET |
| **Target** | /api/v1/deployments/{dep_id}/resources |
| **Content-Type** | application/json |
| **Params** | Any of the resource entity's attributes + page={int}&size={int}&sort={str} |
| **Possible response codes** | 200, 400, 401, 403, 404, 422, 500 |

*Request fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **type** | Parameter to filter the resources by TOSCA type. It can be the exact type of the node or a parent type. | String | Yes |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/f75f8724-f290-417a-abc6-1833a1ad3819/resources' -i -H 'Accept: application/json' -H 'Authorization: Bearer <access token>'
```

```
GET /deployments/f75f8724-f290-417a-abc6-1833a1ad3819/resources HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

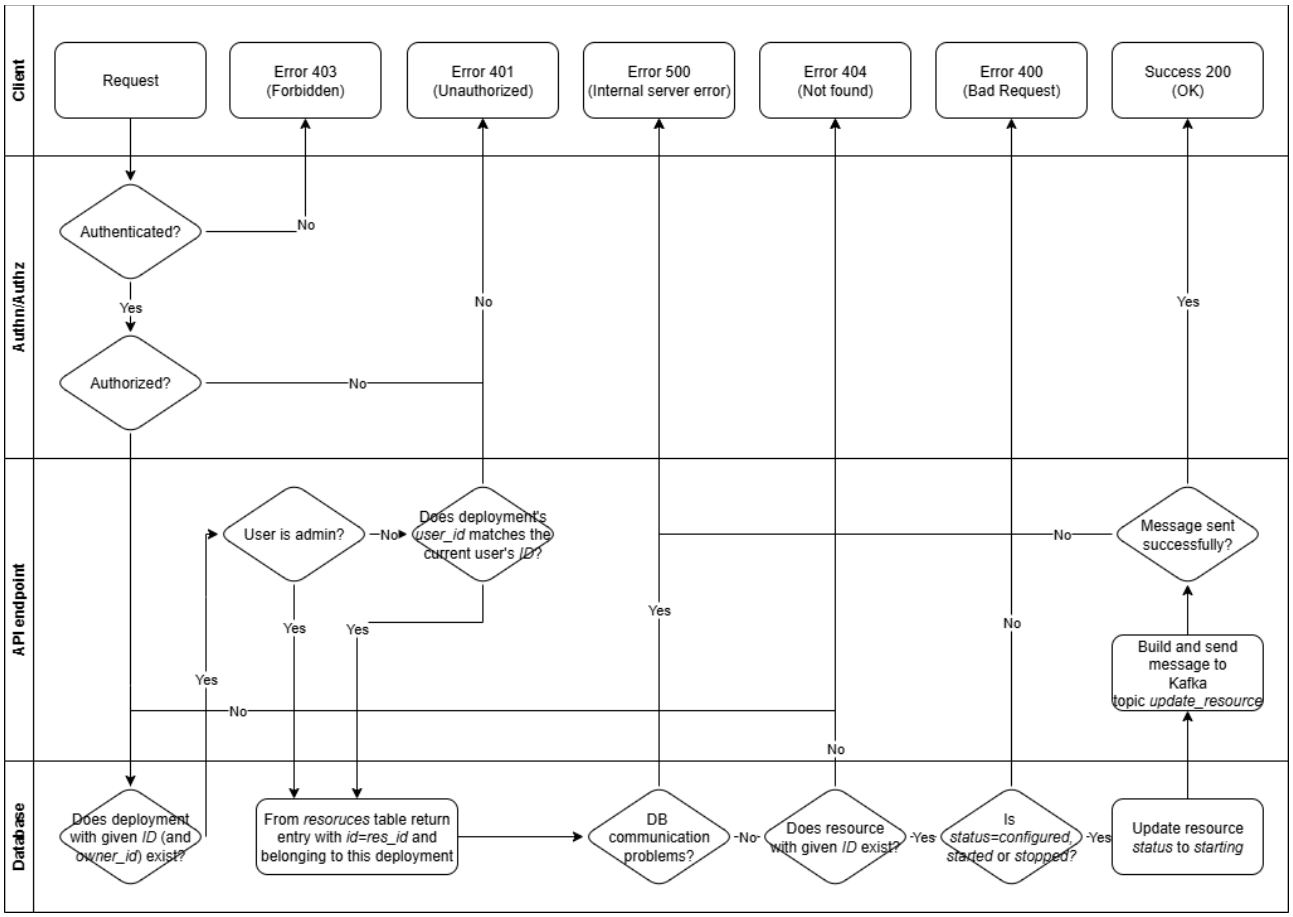
*Response fields description*

See Pagination for main fields. The *data* field contains the list of deployments. Each item has the shape of the response returned by Retrieve accessible resources.

*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2059

{
  "data": [List of Resources],
  "links": {
    "first": "http://example.com/api/v1/deployments/f75f8724-f290-417a-abc6-
1833a1ad3819/resources?page=1&size=5",
    "prev": "http://example.com/api/v1/deployments/f75f8724-f290-417a-abc6-
1833a1ad3819/resources?&page=1&size=5",
    "next": "http://example.com/api/v1/deployments/f75f8724-f290-417a-abc6-
1833a1ad3819/resources?page=3&size=5",
    "last": "http://example.com/api/v1/deployments/f75f8724-f290-417a-abc6-
1833a1ad3819/resources?page=10&size=5",
  }
  "page": {
    "size": 5,
    "total_elements": 50,
    "total_pages": 10
    "number": 3
  }
}
```

## Start a resource




### *Description*

The PATCH request can start a node. Send to the IM the request to start the resource. If the resource is in CONFIGURED, STOPPED or STARTED state, the state will change to STARTING. If the server is in another state, the request is discarded.

### *Submission*

| | |
|---|---|
| **Method** | PATCH |
| **Target** | /api/v1/deployments/{dep_id}/resources/{id}/start |
| **Content-Type** | application/json |
| **Possbile response codes** | 202, 204, 400, 401, 403, 404, 500 |

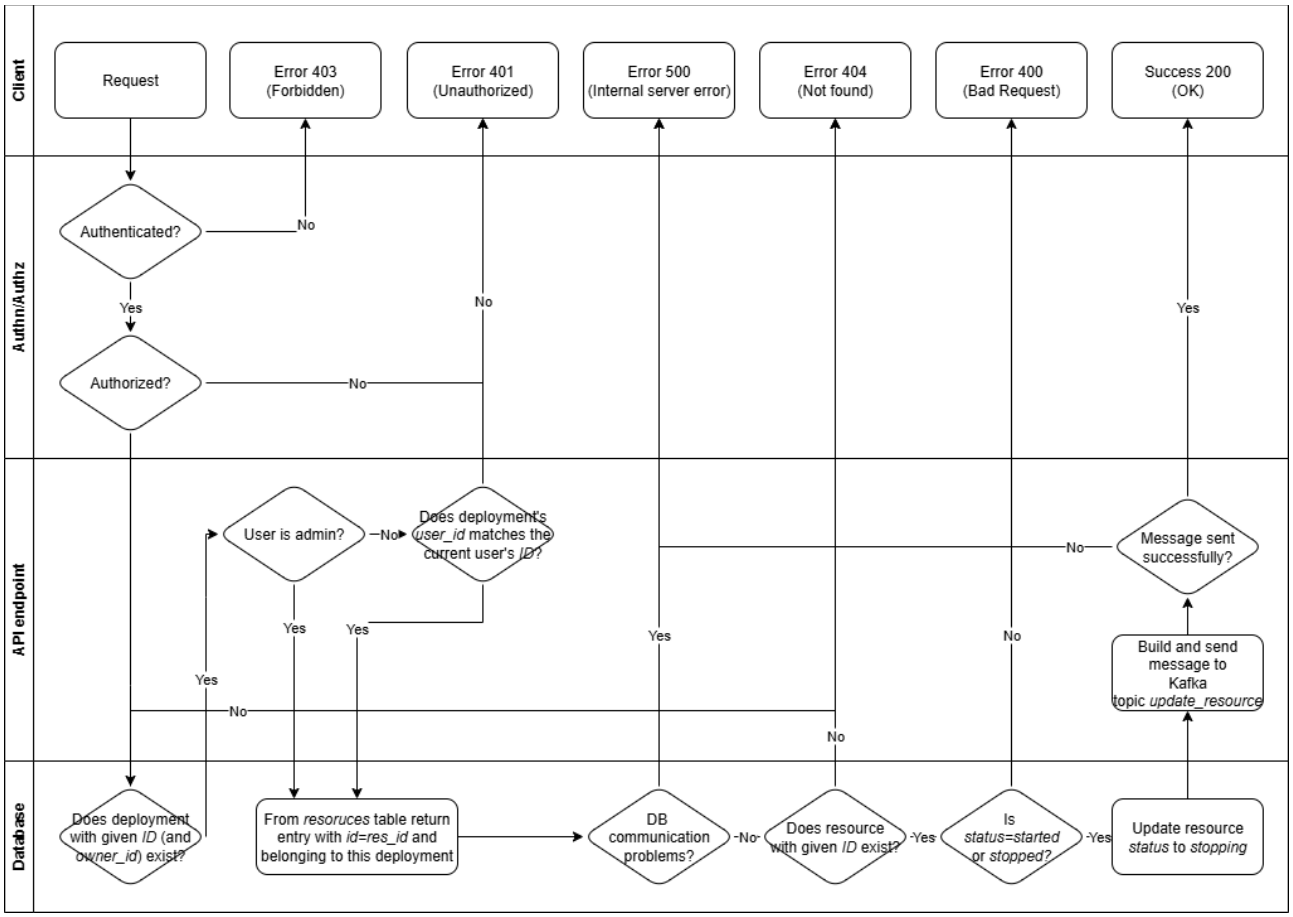
### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/resources/9c77b87f-18f7-4cfb-8782-f7052932b69f/start ' -i -X PATCH -H
'Content-Type: application/json' -H 'Authorization: Bearer <access token>'

PATCH /deployments/34483-d937-4578-bfdb-ebe196bf82dd/resources/9c77b87f-18f7-4cfb-
8782-f7052932b69f/start HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

*Response from server example*

```
HTTP/1.1 202 Accepted
```

## Stop a resource




### *Description*

The PATCH request can start a node. Send to the IM the request to stop the resource. If the resource is in CONFIGURED, STOPPED or STARTED state, the state will change to STOPPING. If the server is in another state, the request is discarded.

### *Submission*

| | |
|---|---|
| **Method** | PATCH |
| **Target** | /api/v1/deployments/{dep_id}/resources/{id}/stop |
| **Content-Type** | application/json |
| **Possbile response codes** | 202, 204, 400, 401, 403, 404, 500 |

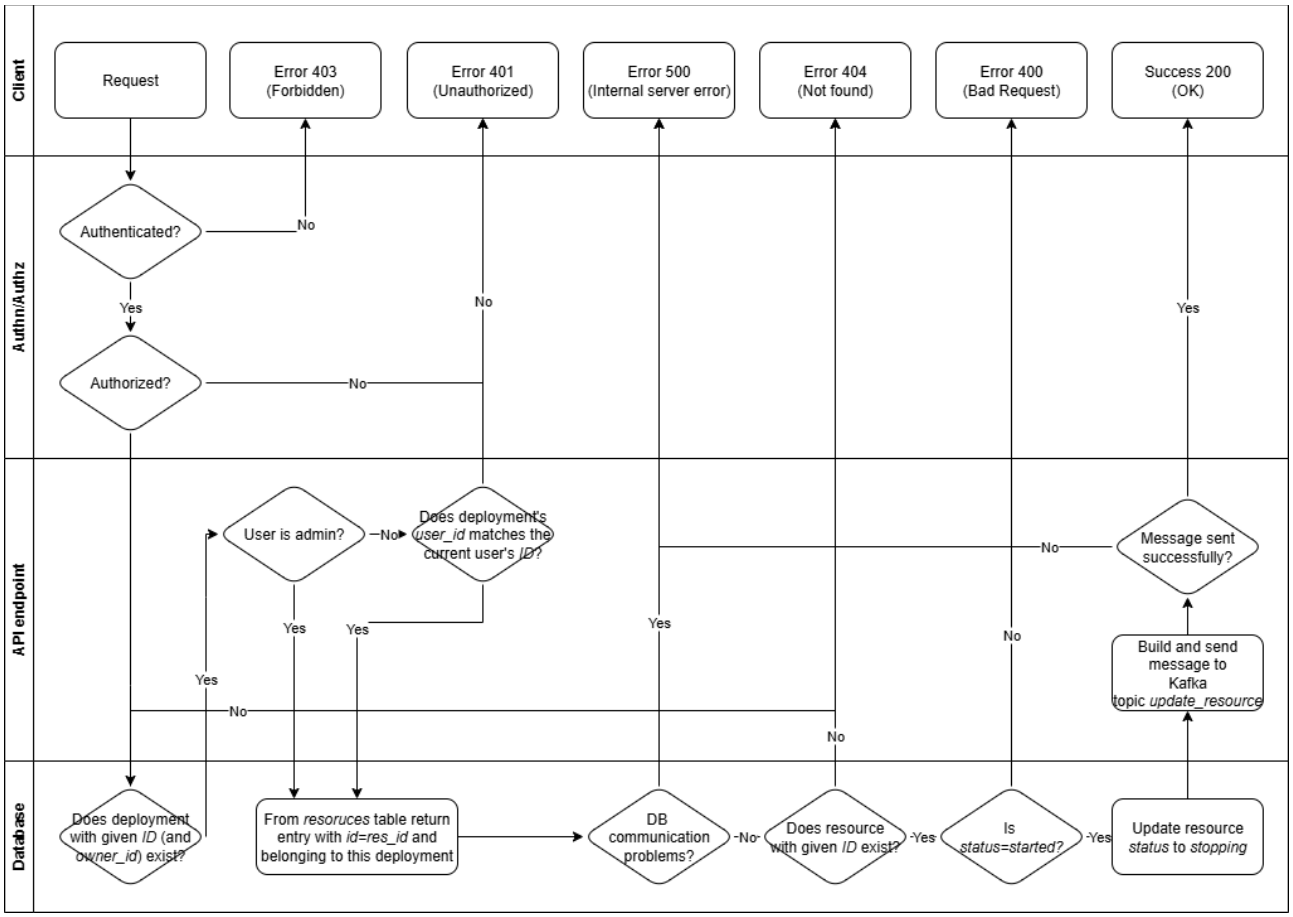
### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/resources/9c77b87f-18f7-4cfb-8782-f7052932b69f/stop ' -i -X PATCH -H
'Content-Type: application/json' -H 'Authorization: Bearer <access token>'

PATCH /deployments/34483-d937-4578-bfdb-ebe196bf82dd/resources/9c77b87f-18f7-4cfb-
8782-f7052932b69f/stop HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

*Response from server example*

```
HTTP/1.1 202 Accepted
```

## Reboot a resource




### *Description*

The PATCH request can start a node. Send to the IM the request to start the resource. If the resource is in STARTED state, the state will change to STARTING. If the server is in another state, the request is discarded.

### *Submission*

| | |
|---|---|
| **Method** | PATCH |
| **Target** | /api/v1/deployments/{dep_id}/resources/{id}/reboot |
| **Content-Type** | application/json |
| **Possbile response codes** | 202, 204, 400, 401, 403, 404, 500 |

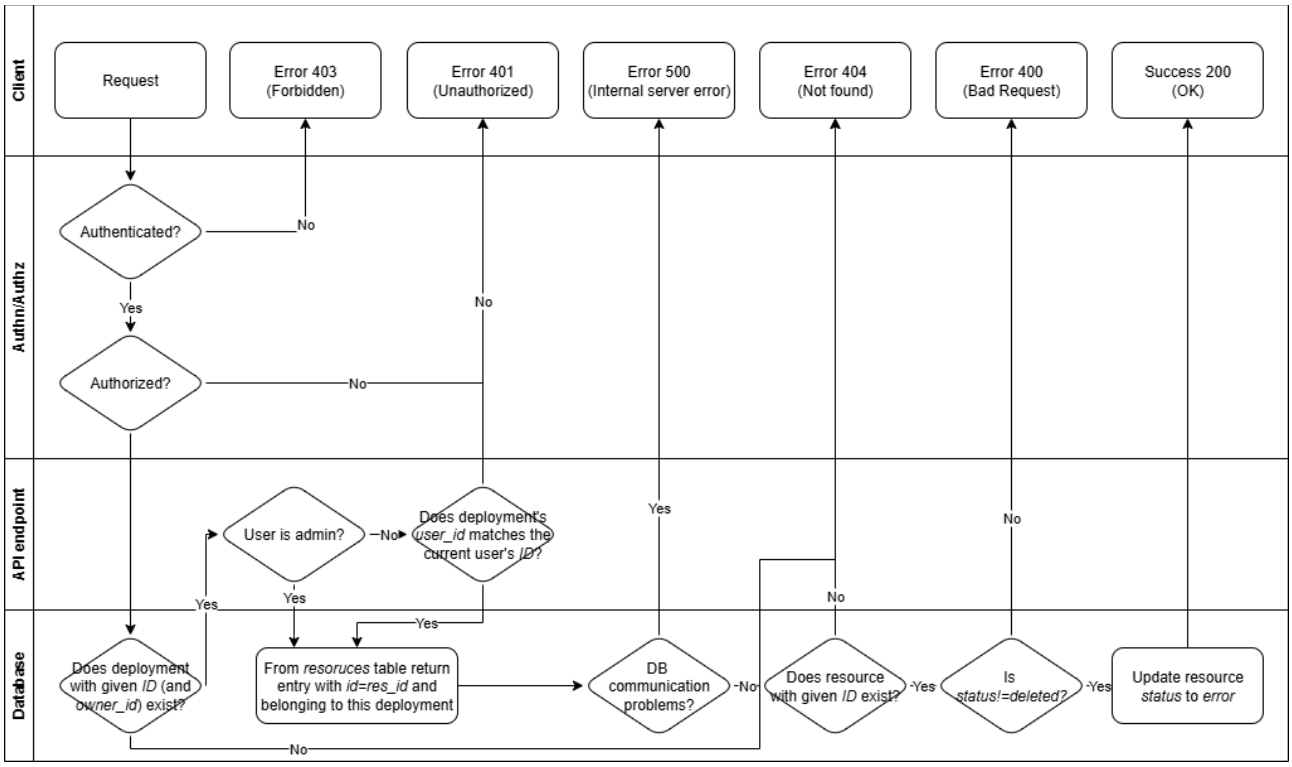
### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/resources/9c77b87f-18f7-4cfb-8782-f7052932b69f/reboot ' -i -X PATCH -H
'Content-Type: application/json' -H 'Authorization: Bearer <access token>'

PATCH /deployments/34483-d937-4578-bfdb-ebe196bf82dd/resources/9c77b87f-18f7-4cfb-
8782-f7052932b69f/reboot HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

*Response from server example*

```
HTTP/1.1 202 Accepted
```

## Reset a resource status




### *Description*

The PATCH request can forcefully reset a node's status to the ERROR state.

### *Submission*

| | |
|---|---|
| **Method** | PATCH |
| **Target** | /api/v1/deployments/{dep_id}/resources/{id}/reset-status |
| **Content-Type** | application/json |
| **Possbile response codes** | 202, 204, 400, 401, 403, 404, 500 |

### *Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/34483-d937-4578-bfdb-
ebe196bf82dd/resources/9c77b87f-18f7-4cfb-8782-f7052932b69f/reset-status ' -i -X
PATCH -H 'Content-Type: application/json' -H 'Authorization: Bearer <access token>'

PATCH /deployments/34483-d937-4578-bfdb-ebe196bf82dd/resources/9c77b87f-18f7-4cfb-
8782-f7052932b69f/reset-status HTTP/1.1
Content-Type: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

### *Response from server example*

```
HTTP/1.1 202 Accepted
```

# Deployment logs

Logs related to a specific deployment and generated by the IM or any log generated by the orchestrator's components and published on kafka. Deployments are automatically updated regularly reading the kafka topic with all services logs.

## Retrieve deployment logs

*Description*

Authenticated users only.

The GET request is used to retrieve all the logs related to a specific deployment.

A standard user can retrieve only logs belonging to owned deployments. Administrators can see any logs.

*Submission*

| | |
|---|---|
| **Method** | GET |
| **Target** | /api/v1/deployments/{dep-id}/logs |
| **Content-Type** | application/json |
| **Possible response codes** | 200, 400, 401, 403, 404, 500 |

*Request submission example*

```
$ curl 'http://orchestrator-api.example.com/api/v1/deployments/f75f8724-f290-417a-abc6-1833a1ad3819/logs -i -H 'Accept: application/json' -H 'Authorization: Bearer <access token>'

GET /deployments/f75f8724-f290-417a-abc6-1833a1ad3819/logs HTTP/1.1
Accept: application/json
Authorization: Bearer <access token>
Host: orchestrator-api.example.com
```

*Response fields description*

| Name | Description | Type | Optional |
|---|---|---|---|
| **timestamp** | Last update timestamp | String ISO 8601 | No |
| **im_logs** | Logs generated by the IM. They can be obtained calling the IM endpoint *infrstrucutres/{infID}/contmsg*. | String | Yes |
| **orchestrator_logs** | Logs generated by the orchestrator components and uploaded to the specific kafka topic for operational logs. | String | Yes |

*Response from server example*

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2059

{
  "timestamp": "2024-01-01T10:00:00.000000Z",
  "im_logs": "IM logs",
  "orchestrator_logs": "Procedure logs",
```

```
}
```

## Delete deployment logs

*Description*

Delete deployment logs. Only administrators or periodic batch scripts.

*Submission*

| **Method** | **DELETE** |
|---|---|
| **Target** | /api/v1/deployment/{id}/logs |
| **Content-Type** | application/json |
| **Possible response codes** | 204, 401, 403, 409 |

*Response from the server example*

```
< HTTP/1.1 204 No Content
< Connection: Keep-Alive
< Content-Length: 0
< Date: Tue, 01 Mar 2022 11:30:06 GMT
```