

Final Project: API Spec

CS 493: Cloud Application Development

Spring 2021

Oregon State University

Last Update: June 5, 2021. 10:30 pm Pacific

Change log	1
Base URL	2
Auth0 URL	2
Data Models	3
Relationships	3
Create a Boat	4
Get a Boat	6
List all Boats	8
PUT a Boat	10
PATCH a Boat	13
Delete a Boat	16
Create a Load	18
Get a Load	20
List all Loads	22
PUT a Load	24
PATCH a Load	26
Delete a Load	28
Assign a Load to a Boat	29
Remove a Load from a Boat	31
List all Users	33

Change log

Version	Change	Date
1.0	Initial version.	April 11 2021
2.0	Added Loads	April 19, 2021
3.0	Added Users, PUT/PATCH endpoints and additional status codes	May 31, 2021

Base URL

<https://cs-493-final.wl.r.appspot.com>

Auth0 URL

<https://cs-493-final.wl.r.appspot.com>

Data Models

Boat

Property Name	Type	Required	Notes
id	Integer		The id of the boat. Automatically generated
name	String	True	Name of boat
type	String	True	Type of boat
length	Integer	True	Length of boat
owner	String		Id of user. Generated from JWT

Load

Property Name	Type	Required	Notes
id	Integer		The id of the load. Automatically generated
volume	Integer	True	Volume of the load
carrier	String		If there is load is on a boat then id of boat, otherwise null. This is set using the PUT /boats/boat_id/loads/Load_id route.
content	String	True	Description of the load
creation_date	Date		Date the load was created. Automatically generated.

User

Property Name	Type	Required	Notes
id	String		Sub value from the JWT. Automatically generated
e-mail	String		User's e-mail address. Pulled from Auth0

Relationships

Each load may be placed on one boat. The boat's is stored as the carrier attribute of the load in order to store the relationship. If a load is not currently assigned to a boat the carrier attribute will be null.

Each boat is owned by a user. The user's id is stored in the boat entity in order to map this relationship.

The user's id is the SUB parsed from the JWT provided as authentication to the POST /boats route. No further information is required besides the attributes listed above in the data models section when creating a boat.

The postman environment includes the user_id1 and user_id2 variables as required by the project specification but these values do not need to be used in order to run the provided collection. Only the url, jwt1, and jwt2 variables are used by the collection.

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Protected

True

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>
Failure	401 Unauthorized	Invalid or missing authorization

Failure	406 Not Acceptable	This endpoint must only respond JSON
---------	--------------------	--------------------------------------

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.
- The `self` attribute will contain the live link to the REST resource corresponding to this boat. In other words, this is the URL to get this newly created boat. You must not store the `self` attribute in Datastore
- If a boat does not have a load, this should appear as an empty array.

Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "def456",
  "loads": [],
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization token"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint can only return JSON"
}
```

Get a Boat

Allows you to get an existing boat

GET /boats/:boat_id

Request

Protected

True

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid or missing authorization
Failure	403 Forbidden	This boat is owned by another user
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 200 OK

```
{
  "id": "1234",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "def456",
  "loads": [
    {
      "id": 5678,
      "self": https://appspot.com/loads/5678
    },
    {
      "id": 8765,
      "self": https://appspot.com/loads/8765
    }
  ],
  "self": "https://appspot.com/boats/1234"
```

```
}
```

Failure

Status: 401 Unauthorized

```
{  
"Error": "Invalid or missing authorization token"  
}
```

Status: 403 Forbidden

```
{  
"Error": "This boat is owned by another user"  
}
```

Status: 404 Not Found

```
{  
"Error": "No boat with this boat_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```

List all Boats

List all the boats.

GET /boats

Request

Protected

True

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid or missing authorization
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

```
Status: 200 OK
{ "results":[
  { "id": 1234
    "name": "Sea Witch",
    "type": "Catamaran",
    "length": 28,
    "owner": "def456",
    "loads":[
      { "id": 5678,
        "self":"https://appspot.com/loads/5678"},
    ],
    "self":"https://appspot.com/boats/1234"
  }, { "id": 1234,
    "name": "Sea Witch",
    "type": "Catamaran",
    "length": 28,
    "owner": "def456",
    "loads":[
      { "id": 5678,
        "self":"https://appspot.com/loads/5678"},
      { "id": 8765,
        "self":"https://appspot.com/loads/8765"}
    ]
  }
]
```



```
  },
  "self": "https://appspot.com/boats/1234"
}],
"next": "https://appspot.com/boats/123423412312"
}
```

Failure

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization token"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint can only return JSON"
}
```

PUT a Boat

Allows you to edit a boat.

PUT /boats/:boat_id

Request

Protected

True

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Body

Required

Request Body Format (MIME type)

application/json

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99
}
```

Response

Response Body Format (MIME type)

Success: none

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If the request is missing any of the 3 required attributes, the boat must not be updated, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never</p>

		contain any attribute that is not listed above).
Failure	401 Unauthorized	Invalid or missing authorization
Failure	403 Forbidden	This boat is owned by another user
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	This endpoint must only respond with JSON

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

Status: 200 OK

```
{
  "id": "1234",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "def456",
  "loads":[
    {
      "id": 5678,
      "self":https://appspot.com/loads/5678
    },
    {
      "id": 8765,
      "self":https://appspot.com/loads/8765
    }
  ],
  "self":"https://appspot.com/boats/1234"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization token"
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat is owned by another user"
}
```

Status: 404 Not Found

```
{  
  "Error": "No boat with this boat_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint can only return JSON"  
}
```

PATCH a Boat

Allows you to partially edit a boat.

PATCH /boats/:boat_id

Request

Protected

True

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Body

Required

Request Body Format (MIME type)

application/json

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	No
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	No
length	Integer	Length of the boat in feet.	No

Request Body Example

```
{
  "name": "Sea Witch"
}
```

Response

Response Body Format (MIME type)

Success: none

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If the request is missing at least one of the 3 required attributes or if it contains all three required attributes, the boat must not be updated, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>

Failure	401 Unauthorized	Invalid or missing authorization
Failure	403 Forbidden	This boat is owned by another user
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	This endpoint must only respond with JSON

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

Status: 200 OK

```
{
  "id": "1234",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "def456",
  "loads":[
    {
      "id": 5678,
      "self":https://appspot.com/loads/5678
    },
    {
      "id": 8765,
      "self":https://appspot.com/loads/8765
    }
  ],
  "self":"https://appspot.com/boats/1234"
}
```

Failure

Status: 400 Bad Request (missing attributes)

```
{
  "Error": "The request object does not contain at least one of the required attributes"
}
```

Status: 400 Bad Request (all 3 attributes)

```
{
  "Error": "The request object contains name, type, and length. Please use the PUT /boats/:boat_id endpoint."
}
```

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization token"
}
```

```
}
```

Status: 403 Forbidden

```
{  
"Error": "This boat is owned by another user"  
}
```

Status: 404 Not Found

```
{  
"Error": "No boat with this boat_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```

Delete a Boat

- Allows you to delete a boat. Deleting a boat will unload any loads that were loaded on to it.

DELETE /boats/:boat_id

Request

Protected

True

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Invalid or missing authorization
Failure	403 Forbidden	This boat is owned by another user
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization token"
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat is owned by another user"
}
```

Status: 404 Not Found


```
{  
"Error": "No boat with this boat_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```

Create a Load

Allows you to create a new slip.

POST /loads

Request

Protected

False

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
volume	integer	The volume of the load	Yes
content	string	The item contained in the load	Yes

Request Body Example

```
{
  "volume": 5, # The volume of the load, an integer
  "content": "LEGO Blocks",
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the volume or content attributes, the load must not be created, and 400 status code must be returned.
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

- Datastore will automatically generate an ID and creation_date and store them with the entity being created. This value needs to be sent in the response body as shown in the example.
- The value of the attribute `carrier` is the ID of the boat the load is assigned to. If there is no assignment, the value of `carrier` should be null.

- The value of the attribute `self` is a live link to the REST resource corresponding to this slip. In other words, this is the URL to get this newly created load. You must not store this attribute in Datastore.

Success

Status: 201 Created

```
{ "id": 5678,  
  "volume": 5,  
  "carrier": null, # The boat carrying the load  
  "content": "LEGO Blocks",  
  "creation_date": "10/18/2020"  
  "self": "https://appspot.com/loads/5678"  
}
```

Failure

Status: 400 Bad Request

```
{  
"Error": "The request object is missing at least one of the required attributes"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```

Get a Load

Allows you to get an existing load.

GET /loads/:load_id

Request

Protected

False

Path Parameters

Name	Type	Description
load_id	String	ID of the load

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 200 OK

```
{ "id": 5678, # automatically generated by the Datastore
  "volume": 5,      # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"
  }, # The boat carrying the load
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "self": "https://appspot.com/loads/5678"
}
```

Failure

Status: 404 Not Found

```
{
```

```
"Error": "No load with this load_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```

List all Loads

List all the slips.

GET /loads

Request

Protected

False

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 200 OK

{ "results": [

```
{ "id": 5677, # automatically generated by the Datastore
```

```
  "volume": 2,      # The volume of the load, an integer
```

```
  "carrier": {
```

```
    "id": 1234,
```

```
    "name": "Sea Witch",
```

```
    "self": "https://appspot.com/boats/1234"
```

```
  },
```

```
  "content": "Lincoln Logs",
```

```
  "creation_date": "10/18/2020" # Date the load was created
```

```
  "self": "https://appspot.com/loads/5678"
```

```
},
```

```
{ "id": 5678, # automatically generated by the Datastore
```

```
  "volume": 5,      # The volume of the load, an integer
```

```
  "carrier": {
```

```
    "id": 1234,
```

```
    "name": "Sea Witch",
```

```
    "self": "https://appspot.com/boats/1234"
```

```
  },
```

```
"content": "LEGO Blocks",  
"creation_date": "10/18/2020" # Date the load was created  
"self": "https://appspot.com/loads/5678"  
}],  
"next": "https://appspot.com/loads/5678"  
}
```

Failure

```
Status: 406 Not Acceptable  
{  
"Error": "This endpoint can only return JSON"  
}
```

PUT a Load

Allows you to edit a Load.

PUT /loads/:load_id

Request

Protected

False

Path Parameters

Name	Type	Description
load_id	String	ID of the load

Request Body

Required

Request Body Format (MIME type)

application/json

Request JSON Attributes

Name	Type	Description	Required?
volume	integer	The volume of the load	Yes
content	string	The item contained in the load	Yes

Request Body Example

```
{
  "volume": 5,  # The volume of the load, an integer
  "content": "LEGO Blocks"
}
```

Response

Response Body Format (MIME type)

Success: none

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure	400 Bad Request	<p>If the request is missing either of the two required attributes, the load must not be updated, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>

Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	This endpoint must only respond with JSON

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

Status: 200 OK

```
{ "id": 5678, # automatically generated by the Datastore
  "volume": 5,      # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"
  }, # The boat carrying the load
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "self": "https://appspot.com/loads/5678"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 404 Not Found

```
{
  "Error": "No load with this load_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint can only return JSON"
}
```

PATCH a Load

Allows you to partially edit a Load.

PATCH /loads/:load_id

Request

Protected

False

Path Parameters

Name	Type	Description
load_id	String	ID of the load

Request Body

Required

Request Body Format (MIME type)

application/json

Request JSON Attributes

Name	Type	Description	Required?
volume	integer	The volume of the load	No
content	string	The item contained in the load	No

Request Body Example

```
{
  "volume": 5
}
```

Response

Response Body Format (MIME type)

Success: none

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure	400 Bad Request	<p>If the request is missing at least one attribute or if it has both attributes, the load must not be updated, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>
Failure	404 Not Found	No load with this load_id exists

Failure	406 Not Acceptable	This endpoint must only respond with JSON
---------	--------------------	---

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

Status: 200 OK

```
{ "id": 5678, # automatically generated by the Datastore
  "volume": 5,      # The volume of the load, an integer
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://appspot.com/boats/1234"
  }, # The boat carrying the load
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020" # Date the load was created
  "self": "https://appspot.com/loads/5678"
}
```

Failure

Status: 400 Bad Request (missing attributes)

```
{
  "Error": "The request object does not contain at least one of the required
attributes"
}
```

Status: 400 Bad Request (all 3 attributes)

```
{
  "Error": "The request object contains both volume and content. Please use the PUT
/loads/:load_id endpoint."
}
```

Status: 404 Not Found

```
{
  "Error": "No load with this load_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint can only return JSON"
}
```

Delete a Load

Allows you to delete a Load.

DELETE /loads/:load_id

Request

Protected

False

Path Parameters

Name	Type	Description
load_id	String	ID of the load

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No load with this load_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint can only return JSON"
}
```

Comments:

The boat-load relationship is implemented by saving the carrier as an attribute of each load. If a load is deleted, no action needs to be taken on the boat entity.

Assign a Load to a Boat

Allows you add a load to a boat.

PUT /boats/:boat_id/loads/:load_id

Request

Protected

True

Path Parameters

Name	Type	Description
load_id	String	ID of the load
boat_id	String	ID of the boat

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Invalid or missing authorization
Failure	403 Forbidden	This boat is owned by another user
Failure	404 Not Found	The specified load and/or boat does not exist
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization"
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat is owned by another user"
}
```

Status: 404 Not Found

```
{  
"Error": " This load is already on a boat"  
}
```

Status: 404 Not Found

```
{  
"Error": "The specified load and/or boat does not exist"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```

Remove a Load from a Boat

Allows you remove a load from a boat.

DELETE /boats/:boat_id/loads/:load_id

Request

Protected

True

Path Parameters

Name	Type	Description
load_id	String	ID of the load
boat_id	String	ID of the boat

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Invalid or missing authorization
Failure	403 Forbidden	This boat is owned by another user
Failure	404 Not Found	No load with this load_id exists at the boat with this boat_id
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing authorization"
}
```

Status: 403 Forbidden

```
{
  "Error": "This boat is owned by another user"
}
```

Status: 404 Not Found

```
{  
"Error": " No load with this load_id exists at the boat with this boat_id"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "This endpoint can only return JSON"  
}
```


List all Users

List all the users.

GET /users

Request

Protected

False

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	This endpoint must only return JSON

Response Examples

Success

```
Status: 200 OK
[
  {
    "id": "abc123" #Sub value from JWT,
    "email": "bob@bob.com"
  }, {
    "id": "def456",
    "email": "bill@bill.com"
  }
]
```

Failure

```
Status: 406 Not Acceptable
{
  "Error": "This endpoint can only return JSON"
}
```