PEOPLE / USERS

Creating new User:

POST /api/people

Request field	Field type	Field Description
username	string	Should not be null, and should be unique
passwd	string	Should not be null
realName	string	Not required

Example:

```
{
        "username": "admin",
        "passwd": "*****",
        "realName": "********"
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError:

Status: 200

Error:

Status: 500

Unauthorized:

Status: 401

Read People / Fetch all users

GET /api/people

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Unauthorized: Status: 401

Add Role To People

PUT /api/people/${peopleId}/role/${roleId}

Example: url/api/people/1/role/2

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Unauthorized: Status: 401

Add Right To Role

PUT /api/role/${roleId}/right/${rightId}

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Unauthorized: Status: 401

Login

POST /api/auth/login

Request field	Field type	Field Description
username	string	Should not be null
password	string	Should not be null

Returns:

NoError: Status: 200

"Token":
"Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsImF1dGhvcml0aWVzIjpbeyJhdXRob3JpdHkiOiJBbmltYWw6UmVhZCJ9LHsiYXV0aG9yaXR5IjoiU3RhdGlzdGljczpSZWFkIn0seyJhdXRob3JpdHkiOiJBbmltYWw6V3JpdGUifSx7ImF1dGhvcml0eSI6IlJPTEVfTlVSU0UifSx7ImF1dGhvcml0eSI6IkFuaW1hbDpSZWFkIn0seyJhdXRob3JpdHkiOiJTdGF0aXN0aWNzOlJlYWQifSx7ImF1dGhvcml0eSI6IlJPTEVfVk9MVU5URVUifSx7ImF1dGhvcml0eSI6IQZW9wbGU6UmVhZCJ9LHsiYXV0aG9yaXR5IjoiQW5pbWFsOlJlYWQifSx7ImF1dGhvcml0eSI6IlN0YXRpc3RpY3M6UmVhZCJ9LHsiYXV0aG9yaXR5IjoiUGVvcGxlOldyaXRlIn0seyJhdXRob3JpdHkiOiJST0xFX0FETUlOIn1dLCJpYXQiOjE2NjQ2NjExMDQsImV4cCI6MTY2NTc5MjAwMH0.le7MFJjtbec-GJ0V82o5S9Sp-932r6hxCtvpiL9C-Mrsmdyt77TJjNf6M2A395IsXJ6QLKyIktwBAvqJWH2jxg
"

Unauthorized: Status: 401

RIGHTS / ROLES

Creating new role

POST /api/role

Request field	Field type	Field Description
roleName		Should not be null and should be unique

Example:

{
        "roleName": "admin"
}

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read Roles / Fetch all the roles

GET /api/role

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Unauthorized: Status: 401

Create new right

POST /api/right

Request field	Field type	Field Description
rightName	string	Should not be null and should be unique

Example:

```
{
        "rightName": "animal:read"
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read Rights / Fetch all rights

GET /api/right

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Unauthorized: Status: 401

SPECIES

Create new specie

POST /api/species

Request field	Field type	Field Description
name	string	Name of the specie

Example:

```
{
        "name": "Cat",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 400 Bad Request

Unauthorized: Status: 401

Read all species / Get all species

GET /api/species

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
[
    {
        "speciesId": 1,
        "name": "Koira"
    },
    {
        "speciesId": 2,
        "name": "Kissa"
    },
    {
        "speciesId": 3,
        "name": "kissa"
    }
]
```

NoError: Status: 200

Unauthorized: Status: 401

Read one specie / Get one specie

GET /api/species/${specieId}

Example:

/api/species/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

```
{
        "speciesId": 1,
        "name": "Koira"
}
```

NoError: Status: 200

Unauthorized:

```
"status": 500,
"error": "Internal Server Error",
"message": "Species with id 2 does not exist!",
```

Edit Species

PUT /api/species/${specieId}

Example:

/api/species/1

Request field	Field type	Field Description
name	string	New name of the specie

Example:

```
{
        "name": "Cat",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 400 Bad Request

Unauthorized: Status: 401

Delete one species

DELETE /api/species/${specieId}

Example:

/api/species/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Body: example.

[ ] (empty array)

Error: Status: 500 if the specie with the given ID doesn't exist.

Unauthorized: Status: 401

IN CLASSES

Create InClassess

POST /api/inclasses

Request field	Field type	Field Description
className	string	Name for the InClass. The name should be unique.

Example:

```
{
        "className": "pet",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

No Error: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read All InClasses/ Get All InClasses

GET /api/inclasses

Example:

/api/inclasses

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
[
    {
        "inclassId": 1,
        "className": "pet"
    }
]
```

NoError: Status: 200

Unauthorized: Status: 401

Read One InClass / Get One InClass

GET /api/inclasses/${inclassId}

Example:

> /api/inclasses/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
{
    "inclassId": 1,
    "className": "pet"
}
```

NoError: Status: 200

Unauthorized:
```
"status": 500,
"error": "Internal Server Error",
"message": "Inclass with id 2 does not exist!",
```

Edit InClass

PUT /api/inclases/${inclassId}

Example:

> /api/inclasses/1

Request field	Field type	Field Description
className	string	New name of the InClass. The name should be unique.

Example:

```
{
        "className": "fox",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Delete One InClass

DELETE /api/inclasses/${inclassId}

Example:

/api/inclasses/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Body: example.

[ ] (empty array)

Error: Status: 500 if the inclass with the given ID doesn't exist.

Unauthorized: Status: 401

STATUS

Create Status

POST /api/status

Request field	Field type	Field Description
name	string	Name of the status.

Example:

```
{
        "status": "Status Name",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read All Status / Get All Status

GET /api/status

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
[
    {
        "statusId": 1,
        "name": "test"
```

```
    },
    {
        "statusId": 2,
        "name": null
    },
    {
        "statusId": 3,
        "name": "SISÄLLÄ"
    },
    {
        "statusId": 4,
        "name": "LOPETETTU"
    },
    {
        "statusId": 5,
        "name": "KUOLLUT"
    }
]
```

NoError: Status: 200

Unauthorized: Status: 401

Read one status / Get one status

GET /api/status/${status}

Example:

/api/status/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
{
    "statusId": 1,
    "name": "test"
}
```

NoError: Status: 200

Unauthorized:

```
"status": 500,
"error": "Internal Server Error",
"message": "Status with Id 99 does not exist!"
```

Edit Status

PUT /api/status/${statusId}

Example:

/api/inclasses/1

Request field	Field type	Field Description
name	string	New name of the status.

Example:

```
{
        "status": "New Status Name",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200
Error: Status: 500 Bad Request

Unauthorized: Status: 401

Delete One Status

DELETE /api/status/${inclassId}

Example:

/api/status/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Body: example.

        [ ] (empty array)

Error: Status: 500 if the inclass with the given ID doesn't exist.

Unauthorized: Status: 401

OBSERVATION

Creating New Observation

POST /api/observation

Request field	Field type	Field Description
date	string	Format: YYYY-MM-DD
value	string	

Example:

```
{
        "date": "2022-09-03",
        "value": "alright"
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read One Observation

GET /api/observation/${observationId}

Example:

/api/observation/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

```
{
        "observationId": 1,
        "date": "2022-09-03T00:00:00.000+00:00",
        "value": "alright",
        "animal": null,
        "observationIncluded": [],
        "people": null
}
```

NoError: Status: 200

Unauthorized:

```
"status": 500,
"error": "Internal Server Error",
"message": "Observation with Id 2 does not exist!"
```

Read All Observations / Get All Observations

GET /api/observation

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

```
[
    {
        "observationId": 1,
        "date": "2022-09-03T00:00:00.000+00:00",
        "value": "alright",
        "animal": null,
        "observationIncluded": [],
        "people": null
    }
]
```

NoError: Status: 200

Unauthorized: Status: 401

Edit Observation

PUT /api/observation/${observationId}

Example:

/api/observation/1

Request field	Field type	Field Description
value	string	New observation value.

Example:

```
{
        "value": "Something write here as a new observation",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Add People To Observation

PUT /api/observation/${observationId}/people/${peopleId}

Example:

/api/observation/1/people/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Add Animal To Observation

PUT /api/observation/${observationId}/animal/${animalId}

Example:

/api/observation/15/animal/2

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Add ObservationType To Observation

PUT /api/observation/${observationId}/observationtype/${observationTypeId}

Example:

/api/observation/16/observationtype/7

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Delete One Observation

DELETE /api/observation/${observationId}

Example:

/api/observation/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Body: example.

[ ] (empty array)

Error: Status: 500

Unauthorized: Status: 401

OBSERVATION TYPE

Creating New Observation:

POST /api/observationtype

Request field	Field type	Field Description
obsTypeName	string	

Example:

```
{
        "obsTypeName": "Observation Type Name",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read All ObservationTypes / Get All ObservationTypes

GET /api/observationtype

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
[
    {
        "typeId": 1,
        "obsTypeName": "Initial Observation",
        "speciesObsJoinIncluded": []
    }
]
```

NoError: Status: 200

Unauthorized: Status: 401

Read One ObservationType

GET /api/observationtype/${observationTypeId}

Example:

/api/observationtype/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:
```
    {
        "typeId": 1,
        "obsTypeName": "Initial Observation",
        "speciesObsJoinIncluded": []
    }
```

NoError: Status: 200

Unauthorized:
```
"status": 500,
```

```
"error": "Internal Server Error",
"message": "Observation type with Id 2 does not exist!"
```

Edit ObservationType

PUT /api/observationtype/${observationTypeId}

Example:

/api/observationtype/1

Request field	Field type	Field Description
obsTypeName	string	New observation type name.

Example:

```
{
        "obsTypeName": "Something write here as a new observation",
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Add SpeciesObsJoin To ObservationType

PUT /api/observationtype/${observationTypeId}/speciesobsjoin/${speciesObsJoinId}

Example:

/api/observationtype/8/speciesobsjoin/13

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Delete One ObservationType

DELETE /api/observationtype/${observationTypeId}

Example:

/api/observationtype/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

[ ] (empty array)

Error: Status: 500

Unauthorized: Status: 401

SPECIES OBS JOIN

Creating New SpeciesObsJoin:

POST /api/speciesobsjoin

Request field	Field type	Field Description

Example:

> {}

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500

Unauthorized: Status: 401

Read All SpeciesObsJoin / Get All SpeciesObsJoin

GET /api/speciesobsjoin

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

```
[
    {
        "speciesObsJoinId": 1,
        "species": null,
        "observation": null
    },
    {
        "speciesObsJoinId": 2,
        "species": {
            "speciesId": 7,
            "name": "Initial"
        },
        "observation": null
    }
]
```

NoError: Status: 200

Read One SpeciesObsJoin

GET /api/speciesobsjoin/${speciesobsjoinId}

Example:

/api/speciesobsjoin/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

```
{
    "speciesObsJoinId": 1,
    "species": null,
    "observation": null
}
```

NoError: Status: 200

Unauthorized:

```
"status": 500,
"error": "Internal Server Error",
"message": "SpeciesObsJoin with Id 3 does not exist!"
```

Add Specie To SpeciesObsJoin

PUT /api/speciesobsjoin/${speciesObsJoinId}/species/${specieId}

Example:

/api/speciesobsjoin/11/species/4

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Add Observation To SpeciesObsJoin

PUT /api/speciesobsjoin/${speciesObsJoinId}/observation/${observationId}

Example:

/api/speciesobsjoin/11/observation/4

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Error: Status: 500 Bad Request

Unauthorized: Status: 401

Delete One SpeciesObsJoin

DELETE /api/speciesobsjoin/${speciesObsJoinId}

Example:

/api/speciesobsjoin/1

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError: Status: 200

Body: example.

[ ] (empty array)

Error: Status: 500

Unauthorized: Status: 401

ANIMAL

Count Animals:

POST /api/animal/total

Request field	Field type	Field Description
startDate	date	Format: YYYY-MM-DD
endDate	date	Format: YYYY-MM-DD

Example:

```
{
        "startDate": "2021-05-01",
        "endDate": "2022-10-08
}
```

VALIDATED TOKEN NEEDS TO BE PROVIDED IN "Header.Authorization"

Returns:

NoError:

Status: 200

Error: Status: 500

Unauthorized: Status: 401