Accounts - API

This document describes the API of a server that manage a database of accounts.
It's only to be used for teaching purposes.
Each account has the following structure:

```
{
  name: string,
  password: string,
  dateCreated: string (YYYY-MM-DD),
  loggedIn: boolean,
}
```

Address: https://maumt.reipino.com/du3/servers/accounts
From now on, shortened to **URL** in this document.

General

This server manages accounts and can be used for teaching purposes.
There cannot be two accounts with the same name.
Content-Type must be set to "application/json" except for GET-requests.
All responses are JSON-formatted.

Endpoints

/

Use this endpoint to manage the accounts in the database.

GET

A POST-request to this endpoint will return a JSON-formatted array of all the accounts in the DB

/manage

Use this endpoint to manage the accounts in the database.

POST

A POST-request to this endpoint will create a new account.
The request body must contain a JSON-formatted object:

```
{
  name: string,
  password: string
}
```

Potential response-status:

- OK
  200: Everything ok. Account was created.
- NOT OK (Account not created)
  400: Wrong Content-Type or incomplete data (no name or no password)
  409: The provided name already exists in the DB and cannot be used.

DELETE

A DELETE-request to this endpoint will remove an existing account.
The request body must contain a JSON-formatted object:

```
{
  name: string,
  password: string
}
```

The values of name and password must fit an existing account. If they do, that account will be removed. Potential response-status:

- OK
  200: Everything ok. Account was deleted.
- NOT OK (Account not deleted)
  400: Wrong Content-Type or incomplete data (no name or no password)
  404: The provided name does not exist in the DB and cannot be deleted.
  409: The provided name and password are incorrect.

/login

Use this endpoint to check if a pair name/password fits an existing account.

POST

A POST-request to this endpoint will check if a pair name/password fit an existing account.
The request body must contain a JSON-formatted object:

```
{
  name: string,
  password: string
}
```

Potential response-status:

- OK
  200: Login successful. Account exists and has those values.
- NOT OK
  400: Wrong Content-Type or incomplete data (no name or no password)
  404: Either there is no account with that name or the password is incorrect.