# REST calls (**http://localhost:8082**)

## [UserController] = **…/api/User**

**HttpGet [{userId}]**   
Haalt data van een enkele User op om te displayen op de pagina.  
*Returns:* **UserDataDTO**

**HttpGet []**  
Haalt data van alle Users op om te displayen  
*Returns:* **List<UserDataDTO>**

**HttpGet [page/{pageNumber}]**  
Haalt data op van de x-aantal users op de {pageNumber}-ste pagina met users, om te displayen voor admins.  
*Returns:* **List<UserDataDTO>**

**HttpDelete [{userId}]**   
Verwijdert een gebruiker uit de database.  
*Returns:* Ok of Badrequest

## [AuthController] = **…/api/Authentication**

**HttpPost [register] + registerDTO**  
Stuurt userDTO mee om een nieuwe gebruiker te registreren.  
*Returns:* Ok of BadRequest, ***later waar nodig een token.***

**HttpPost [login] + loginDTO**  
Stuurt een userDTO met ingevulde data om gebruiker proberen in te loggen.  
*Returns:* Ok of BadRequest, ***later waar nodig een token.***

**HttpPut [] + editUserDTO**   
Stuurt een userDTO met ingevulde data om deze aan te passen.  
BackEnd kijkt welke data anders is en past alleen deze data aan.  
*Returns:* Ok of BadRequest

## [StationController] = **…/api/Station**

**HttpGet [{stationId}]**   
Haalt een station op gebaseerd op zijn Id.  
*Returns:* **stationDTO**

**HttpGet [user/{userID}]**  
Haalt de flatStations op van een user, om te displayen op een lijst in het front-end.  
Zodra op de frontend een van de opties wordt geselecteerd, worden de details opgehaald via de GET hierboven.  
*Returns:* **List<StationTitleDTO>**

**HttpGet []**  
Haalt alle stations op  
*Returns:* **List<stationTitleDTO>**

**HttpGet [page/{pageNumber}]**  
Haalt data op van de x-aantal users op het {pageNumber}-ste station, om te displayen voor admins.  
*Returns:* **List<stationTitleDTO>**

**HttpPost [] + createStationDTO**   
Stuurt stationDTO mee om een nieuw station te registreren.  
Hiervoor moeten we nog kijken waarmee stations geregistreerd worden.  
*Returns:* Created of BadRequest

**HttpDelete [{stationId}]**   
Verwijdert een station, zijn sensoren en locatie uit de database.  
*Returns:* Ok of Badrequest

**HttpPut [] + editStationDTO**Stuurt een stationDTO met ingevulde data om deze aan te passen.  
BackEnd kijkt welke data anders is en past alleen deze data aan.  
*Returns:* Ok of BadRequest

## [SensorController] = **…/api/Sensor**

**HttpGet [{sensorId}]**  
Haalt een sensor op aan de hand van het sensorId.  
*Returns:* **sensorDTO**

**HttpGet []**  
Haalt alle sensoren op.  
*Return:* **List<sensorDTO>**

**HttpGet [type]**  
Haal alle sensortypes op om te displayen op het front-end.  
*Returns:* **List<sensorTypeDTO>**

**HttpGet [type/{typeId}]**   
Haalt alle sensoren van dit type op.  
*Returns:* **List<sensorDTO>**

**HttpGet [average]**   
Haalt de gemiddeldes van alle verschillende soorten sensoren op.  
*Returns:* **sensorAverageDTO**

**// COPY  
HttpGet [type/{typeId}]**   
Haalt alle sensoren van dit type op.  
*Returns:* **List<sensorDTO>  
// COPY**

**HttpGet [station/{stationId}]**  
Haalt alle sensoren op aan de hand van hun station  
*Returns:* **List<sensorDTO>**

**HttpPost [] + sensorDTO**   
Stuurt sensorDTO mee om een nieuwe sensor te registreren op het stationId dat erin staat.  
*Returns:* Ok of BadRequest

**HttpPost [newType] + sensorTypeDTO**  
Creëert een nieuw sensortype voor de database om mee te werken.  
*Returns:* Ok of BadRequest

**HttpDelete [{sensorId}]**Verwijdert een sensor.  
*Returns:* Ok of Badrequest

# Models

## userDataDTO

id : int  
username : string  
mailAddress : string  
admin : bool

## userDTO

id : int  
username : string  
mailAddress : string  
passwordHash : string  
passwordSalt : string

## registerDTO

username : string  
mailAddress : string  
password : string

## editUserDTO

username : string  
Id : int  
mailAddress : string  
password : string

## loginDTO

mailAddress : string  
password : string

## editStationDTO

Id : int  
name : string  
height : double  
address : string  
longitude : double  
latitude : double   
isPublic : bool

## registerStationDTO

userId : int  
name : string  
height : double  
address : string  
longitude : double  
latitude : double   
isPublic : bool

## stationDTO

id : int  
name : string  
height : double  
locationId : int  
locationName : string  
latitude : double  
longitude : double  
isPublic : bool  
sensors : List<sensorDTO>

## stationTitleDTO

id : int  
name : string

## sensorDTO

id : int  
stationId : int  
typeId : int  
data : double

## sensorTypeDTO

id : int  
name : string

## sensorAverageDTO

temperatuur : double  
stikstof : double  
koolstofdioxide : double  
fijnstof : double  
luchtvochtigheid : double  
windsnelheid : double