Wetr.WebService

Der Wetr.WebService ist eine ASP.NET REST-API. Anfragen auf diese API werden in verschiedenen Controllers bearbeitet.
Diese Controller rufen dafür Methoden aus dem Wetr.Server auf. CORS wird verwendet, um den Zugriff auf die API zu erlauben.

Um das Arbeiten mit dieser API zu erleichtern, wird Swagger verwendet. Mit diesem Programm kann Client-Code für diese API automatisch generiert werden.

MeasurementsController

In diesem Controller wird bei Fehler bei Abfragen vom Wetr.Server eine HttpResponseException ausgelöst. Dies gilt für alle GET Routen.

Insert (POST /measurements)

Über diese Route können neue Messdaten in das System gespeichert werden. Dazu muss im Body ein Measurement Objekt übergeben werden. Dieses wird dann vom Wetr.Server hinzugefügt. War das hinzufügen erfolgreich wird true zurückgegeben, ansonsten false.

GetMeasurementsForStation (GET /measurements)

Mithilfe dieser Route können Messwerte eines Messwerttyps für eine bestimmte Station in einem bestimmten Zeitraum abgefragt werden. Dazu müssen die geforderten Parameter (Startzeit, Endzeit, Typ, Station) als URI-Parameter übergeben werden. Es wird eine Liste von Messwerten zurückgeben, im Fehlerfall null.

GetAllUnits (GET /measurements/unit)

Mit dieser Route können alle Einheiten die für die Messewerte möglich sind abgefragt werden. Zurückgegeben wird eine Liste von Einheiten oder null.

GetLastMeasurementsForStation (GET /measurements/last)

Über diese Route können die letzten Messwerte eines bestimmten Typs einer bestimmten Station abgefragt werden. Dazu müssen die Parameter Typ, Stationsnummer und die Anzahl der letzten Messwerte als URI Parameter übergeben werden. Als Ergebnis wird eine Liste von Messwerten oder null zurückgegeben.

GetMeasurementsMin, GetMeasurementsMax, GetMeasurementsAvg, GetMeasurementsSum (GET /measurements/[sum, min, max, avg])

Diese Routen dienen zur Analyse der Messdaten. Jede Route benötigt als Parameter einen Startzeitpunkt, Endzeitpunkt, Messdatentyp, Gruppierungsmodus und eine Stationsnummer als URI-Parameter übergeben. Zurückgesendet wird eine Liste von Messdatenauswertungsobjekten oder null. Jede Route aggregiert (z.B. Summieren, Durchschnittsberechnung) die Messdaten in einem bestimmten Zeitabschnitt und bestimmten Zeitblöcken (z.B. stündlich, täglich).

UsersController

In diesem Controller wird bei Fehler bei Abfragen vom Wetr.Server eine HttpResponseException ausgelöst. Dies gilt für alle GET Routen.

GetUserByEmail (GET /users)

Diese Route gibt den Benutzer nach der eingegebenen Email, oder falls der Benutzer nicht vorhanden null, zurück. Dazu muss die Eingegebene Email als URI-Parameter übergeben werden.

GetUserById (GET /users/id)

Retourniert einen Benutzer nach der als URI-Parameter angegebenen ID. Ist der Benutzer nicht vorhanden wird null zurückgegeben.

CheckLogin (POST /users/login)

Im Body müssen Logindaten wie Benutzername und Passwort übergeben werden. Wetr.Server prüft dann ob der Benutzer angemeldet werden darf und gibt dann den Benutzer oder null zurück.

CheckOAuthLogin (POST /users/login/oauth)

Funktioniert wie die CheckLogin Route, nur das hier falls der OAuth Benutzer noch nicht im System bekannt ist, dieser auch gleich ins System eingepflegt wird. Auch hier wird entweder der Benutzer oder null zurückgegeben.

StationsController

Im Controller für Stationen gibt es folgende Routen:

GetAll (GET /stations)

Gibt alle Stationen zurück.

AddStation (POST /station)

Speichert eine neue Station in der Datenbank. Die neue Station wird aus dem Request-Body entnommen.

GetAllTypes (GET /stations/types)

Gibt alle Stationstypen zurück.

GetById (GET /stations/{id})

Gibt eine Station mit einer bestimmten ID zurück.

GetByCommunityId (GET /stations/community/{id})

Gibt alle Stationen in der Gemeinde mit einer bestimmten ID zurück.

UpdateStation (PUT /stations/{id})

Aktualisiert die Werte in der Datenbank einer Station mit einer bestimmten ID. Die aktualisierte Station wird aus dem Request-Body entnommen.

DeleteStation (DELETE /stations/{id})

Löscht eine Station mit einer bestimmten ID.

CommunitiesController

Der Controller für Gemeinden mit folgenden Routen:

GetAll (GET /communities)

Gibt alle Gemeinden zurück.

GetByName (GET /communities/name)

Gibt alle Gemeinden zurück, deren Namen mit dem URL-Parameter "name" übereinstimmen.

GetStations (GET /communities/stations)

Gibt alle Gemeinden zurück, in denen es Wetterstationen gibt. Hier kann optional ein Suchbegriff als URL-Parameter übergeben werden. Ist so ein String präsent, werden nur Gemeinden mit Stationen zurückgegeben, in deren Namen der Suchbegriff vorkommt.

MeasurementTypesController

Der Controller für Messwerttypen mit folgenden Routen:

GetAll (GET /measurementtypes)

Gibt alle Messwerttypen zurück.