API Docs:

**Server**: https://vital-agro-api.onrender.com (first request after a 15-minute wait takes up to 30 seconds)

**User**: path: /api/users

**Weighings**: path: /api/weighings

**Constants**: path: /api/constants

Constants

**GET**: /subscriptions, /crops, /sourcesList, /destinationsList, /harvesters, /drivers: - **Response**: array of Strings  
**GET**: /autos: - **Response**: array of {“\_id”, “model”, ”licensePlate”}

User interface:

**Register:**

**POST***: /signup***BODY**: **JSON**: {“email”: email, “password”, string }  
**Response**: status 201, **JSON**: {user:{email, subscription}}, verificationToken is sent in an email to the address specified during registration.

**Verify**:

**GET**: /verify/:verificationToken  
**Response**: status 200, confirmation message ***//TODO: redirect to login page***

**In case verification e-mail is not delivered, to request a re-send:**

**POST**: /verify  
**BODY**: **JSON**: {“email”: email}  
**Response**: status 200, confirmation message.

**Login:**

**POST**: /login  
**BODY**: **JSON**: {“email”: email, “password”: string }  
**Response**: status 200, **JSON**: {“token”:JWT\_TOKEN, “user”:{“email”:email, “subscription”: subscription}}

JWT Token must be included in further communications by means of “Authorization” header:

Header: (“authorization”: ”Bearer JWT\_TOKEN”}

**To get current user information:**

**GET**: /current  
**Response**: status 200, **JSON**: {“email”:email, ”token”:JWT\_TOKEN, ”subscription”: subscription}

**To logout:**

**POST**: /logout  
**Response**: status 204

**To update user subscription (done only by owner user):**

**POST**: /  
**BODY**: **JSON**: {“email”: email, “subscription”:subscription}  
**Response**: status 200, **JSON**: {“email”: “”, “subscription”:””}

Weighings interface

JWT Token must be included in further communications by means of “Authorization” header:

**Header**: (“authorization”: ”Bearer JWT\_TOKEN”}

**Add weighing:**

**POST:** /api/weighings  
**BODY**: **JSON**: {  
“auto”:{“id”:ObjectId link to autos.\_id, “driver”: drivers },  
“crop”:{“name”: crops, “source”:sourcesList, “destination”:destinationsList},  
“weighing”:{“tare”: number, ”brutto”: number, “isIncoming”: boolean}, //”netto”: number may specified and equal to (“brutto” – “tare”).  
“harvesters”:[{“name”: harvesters, “weight”:number}] //all weight shares for harvesters must add up to netto

**Response**: **JSON**: {  
“warnings”:array of strings, // If any  
“weighing”:object} with data from BODY with added “\_id” and corrected netto and harvester share weights (if needed, with warnings returned)

**Get weighings** // TODO: Will use filters to sort by source, destination, driver, isIncoming, crop

**GET**: /api/weighings?source&destination&driver&incoming&crop