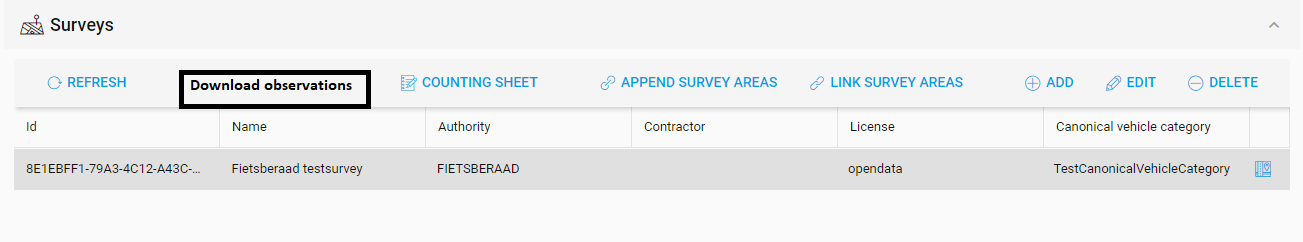
Export dynamic data

Data from a Survey must be downloadable by click a button ‘Download observations’ at Survey level:



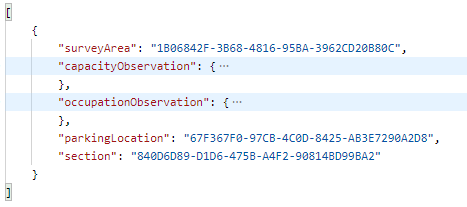
Get the data by endpoint:

GET http://remote.veiligstallenontwikkel.nl/rest/api/v2/surveys/<survey.id>/combinedobservations



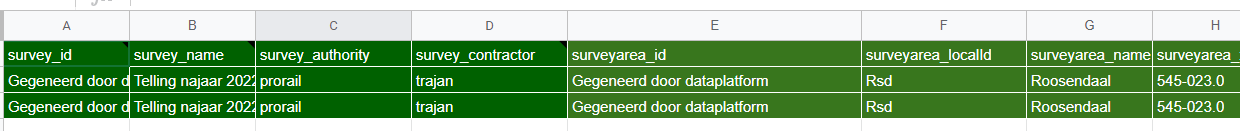
You’ll receive an array of so called Combined Observations. A Combined Observation is an occupation observation combined with a capacity observation:



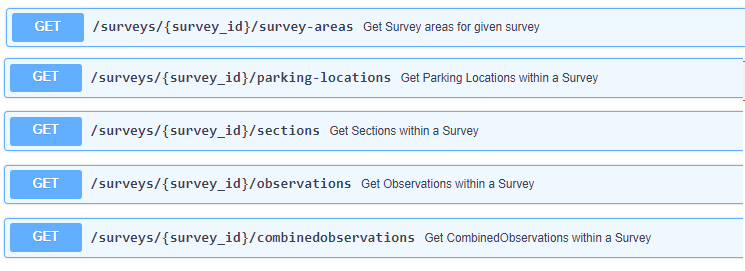


This response must be converted into a **Excel sheet** (no CSV!) with columns like in:

<https://docs.google.com/spreadsheets/d/1z3ECyDDUbYOoj8uu6Y-oyZTM6v1fIyJ_jDd_qXGfhr0/edit#gid=0>



Properties for Section, Parking-Location and Survey-Area, such as names and local IDs can be retrieved by these endpoints.



# Mapping:

survey\_id: Survey.id

survey\_name: Survey.name

survey\_authority: Survey.authority

survey\_contractor: Survey.contractor

surveyarea\_id: SurveyArea.id (SurveyArea with empty or no property ‘parent’)

surveyarea\_localId: SurveyArea.localId

surveyarea\_name: SurveyArea.name

surveyarea\_xtrainfo: SurveyArea.xtrainfo

surveyarea\_localId\_child: child SurveyArea.id (optional, child is surveyArea with prop parent filled)

surveyarea\_name\_child: child SurveyArea.name (optional, child is surveyArea with prop parent filled)

geolocation: https://remote.veiligstallenontwikkel.nl/rest/api/v2/parking-locations/<parking-location.id>, https://remote.veiligstallenontwikkel.nl/rest/api/v2/sections/<section.id>

parkinglocation\_id: parkinglocation.id

parkinglocation\_localId: parkinglocation.localId

parkinglocation\_name : parkinglocation.name

parkinglocation\_locationFeatureType : comma separated list of parkinglocation.features

parkinglocation\_xtrainfo: parkinglocation.xtrainfo

section\_id: section.id

section\_localId: section.localId

section\_name: section.name

section\_layout: section.layout

section\_parkingSystemType: section.

section\_vehicleOwnerType: section.

section\_level: section.level

observation\_capacity\_id: capacityObservation.id

observation\_capacity\_timestamp\_start: capacityObservation.timestampStart

observation\_capacity\_timestamp\_end: capacityObservation.timestampEnd

observation\_capacity\_parkingCapacity: capacityObservation.parkingCapacity

observation\_capacity\_note: capacityObservation.note

observation\_occupation\_id: occupationObservation.id

observation\_occupation\_timestamp\_start: occupationObservation.timestampStart

observation\_occupation\_timestamp\_end: occupationObservation.timestampEnd

occupation\_totalParked: occupationObservation.measurement.totalParked

observation\_occupation\_note: occupationObservation.note

Get Canonical Vehicles for survey:

[https://remote.veiligstallenontwikkel.nl//rest/api/v2/canonical-vehicle-categories/<survey.canonicalVehicleCategory>/canonical-vehicles](https://remote.veiligstallenontwikkel.nl//rest/api/v2/canonical-vehicle-categories/%7Bsurvey.canonicalVehicleCategory%7D/canonical-vehicles)

Create a column for each Canonical Vehicle and get its data from the occupationObservation.measurement.vehicleTypeCount with prop canonicalVehicleCode = canonicalVehicle.code

<canonicalVehicle.code> <canonicalVehicle.description>: occupationObservation.measurement.vehicleTypeCount.numberOfVehicles

All timestamps in **local time** format:

**d-m-yyyy H:mm:ss**

So no leading zeros:

Example:

6-7-2023 9:03:02 = 6 July 2023 09:03:02

23-12-2023 19:03:02 = 23 December 2023 19:03:02

(If converting UTC timestamps to local time is too complicated, keep UTC timestamp (like 2023-01-01T02:00:00+0100).