

Technolution B.V.
Burg. Jamessingel 1
P.O. Box 2013
2800 BD Gouda
The Netherlands

T +31 (0)182 59 40 00
E info@technolution.nl
I www.technolution.com



Technolution

Memo

To:
From: Erwin Gribnau, Edwin Mein, Tobias Beckman
Subject: API specification for Scania in SMUTZ trial
Date: May 12th, 2021

Introduction

This short memo outlines an API specification for the purpose of providing identified geofences from the SMUTZ prototype system to the Scania Cloud system and also providing identified vehicles inside an geofence area from Scania Cloud system to SMUTZ prototype system.

Design constraints

In several consortium meetings the following decisions have already been agreed upon:

- The Scania Cloud system will **pull** identified geofences from the SMUTZ prototype system
- The Scania Cloud system will **push** identified vehicles inside geofences to the SMUTZ prototype system
- The data will transported using an HTTP POST and GET requests
- The data will be serialized in JSON format and the pull request will also use DateX II protocol.
- The SMUTZ prototype system will employ an IP-whitelist to ensure only previously agreed upon IP-addresses will be able to send data to the SMUTZ prototype system.



Pull geofences API

URL: /smutz/scania/geofences

Allowed methods: GET

Username:

Password:

Header: If-modified-since can be used

Get message

JSON object with a list of 'geofences' using DateX II protocol.

Example

```
{
  "@type": "com:GenericPublication",
  "@lang": "en-us",
  "@modelBaseVersion": "3",
  "@schemaLocation": "http://levelC/schema/3/d2Payload_LevelC_3_D2Payload.xsd",
  "publicationTime": "2021-05-06T13:00:00Z",
  "publicationCreator": {
    "country": "se",
    "nationalIdentifier": "STA01"
  },
},
"genericPublicationName": "DynamicSmartUrbanTrafficZonePublication",
"_genericPublicationExtension": {
  "razTablePublication": {
    "headerInformation": {
      "informationStatus": "real"
    },
  },
  "regulatedAccessZoneTable": [{
    "regulatedAccessZoneTableName": {
      "values": {
        "value": "Low Speed Zone"
      }
    }
  },
  "tableVersionTime": "2021-05-06T13:00:00Z",
  "regulatedAccessZone": {
    "@id": "1",
    "@version": "1",
    "regulatedAccessZoneName": {
      "values": {
        "value": "Test Zone Hammarby Alle Stockholm"
      }
    }
  },
  "regulatedAccessZoneRecordVersionTime": "2021-05-06T13:00:00Z",
  "validityOfRegulation": {
    "overallStartTime": "2021-01-09T13:00:00Z"
  },
  "regulatedAccessZoneRestriction": {
    "@id": "1",
    "@version": "1",
    "applicableLocation": {
      "gmlMultiPolygon": {
        "gmlPolygon": {
```



```
"exterior": {
  "geoJSONposList": [[
    18.08528780937195,
    59.302252024938475
  ],
  [
    18.083667755126953,
    59.30190696156252
  ],
  [
    18.083549737930298,
    59.301665962842975
  ],
  [
    18.08411836624145,
    59.30141948511376
  ],
  [
    18.0864679813385,
    59.30153998556006
  ],
  [
    18.089107275009155,
    59.30215891294375
  ],
  [
    18.088656663894653,
    59.30262994747695
  ],
  [
    18.085695505142212,
    59.301934347672606
  ],
  [
    18.08528780937195,
    59.302252024938475
  ]
]]
}
}
},
"accessProhibitedTo": {
  "restrictionActivationType": "dynamicallyActivated",
  "validity": {
    "validityStatus": "active",
    "validityTimeSpecification": {
      "overallStartTime": "2021-05-06T13:00:00Z"
    }
  }
},
"applicableToVehicleWithCharacteristic": {
  "regulatedCharacteristics": {
    "euVehicleCategory": [
      "n2",

```



```
        "n3"
      ]
    },
    "maxSpeedLimitation": {
      "speedvalue": "30"
    },
    "maxAxleWeight": {
      "axleWeight": "5"
    }
  }
}
}
}
}
}
}
}
}
}
}
```

About the fields

geoJSONposList	Array of float: A valid GeoJSON list of points with positions [[longitude, latitude],[]]
speedvalue	Float: Maximum speed allowed in km/h.
axleWeight	Float: Maximum heaviest axle weight allowed in tons.

Reponse

HTTP OK when correctly received.



Push vehicle positions API

URL: /smutz/scania/positions

Allowed methods: POST

Post body

JSON object with a list of vehicle positions.

Example

```
{
  "objects": [
    {
      "uuid": "1fa7047f-a6dc-4fde-a449-1d0c5d97c66e",
      "class": "truck",
      "position": [18.054, 59.32796],
      "timestamp": 1620292383000,
      "speed": 4.3,
      "heading": 120,
      "heaviestAxleWeight": 12,
      "weight": 2
    },
    {
      "uuid": "1fa7047f-a6dc-4fde-a449-1d0c5d97c66e",
      "class": "truck",
      "position": [18.058, 59.32796],
      "timestamp": 1620292384000,
      "speed": 4.5,
      "heading": 120,
      "heaviestAxleWeight": 12,
      "weight": 2
    },
    {
      "uuid": "1fa7047f-a6dc-4fde-a449-1d0c5d97c67e",
      "class": "truck",
      "position": [18.054, 59.325],
      "timestamp": 1620292383000,
      "speed": 12.7,
      "heading": 130,
      "heaviestAxleWeight": 10,
      "weight": 1
    }
  ]
}
```



About the fields

uuid	String: A string value representing a uuid
class	String: One of: "pedestrian", "bicycle", "truck", "..."
position	Array of float: A valid GeoJSON formatted position [longitude, latitude]
timestamp	Integer: Milliseconds since Unix Epoch
speed	Float: Speed over ground in kilometers per hour
heading	Float: Course over ground in degrees referencing true north
heaviestAxleWeight (Optional)	Float: Heaviest axle weight in tons.
weight (Optional)	Float: Total weight of the vehicle in tons.

Response

HTTP OK when correctly received.