# Swedavia
# RVU API

apideveloper.swedavia.se

# Using the RVU API

## Table of contents

# 1 THE RVU API

The RVU API offers a way of sending the result of a passenger survey through an API over HTTPS.

## 1.1 Date and time values

Datetimes should be sent in UTC format.

# 2 ENDPOINTS

This section explains the endpoint of the RVU API. Further, request body and possible responses are explained.

## 2.1 Survey

The endpoint is designed to receive data from surveys conducted at Swedavia airports. One request should be performed per survey response, and the request body should contain the data from the survey.

```
POST https://api.swedavia.se/rvu/v1/survey
```

### 2.1.1 Request Body

The column ‘Null’ describes whether the property is nullable or not.

| *Property* | *Null-* | *Data Type* | *Description* |
|---|---|---|---|
| *CorrelationId* | No | String | Unique identifier for the survey response. Should be generated by the party responsible for conducting the survey. |
| *SupplierId* | No | String | Identifier of the party responsible for conducting the survey. This constant will be agreed upon in cooperation with Swedavia. |
| *AirportIata* | Yes | String | The IATA of the airport where the survey was conducted. |
| *DateOfInterview* | Yes | String (UTC-formatted DateTime string) | Date and time when the survey was conducted, in the format yyyy-MM-ddTHH:mm:ss.FFFZ. |
| *PassengerInformation* | No | Object (PassengerInformation) | Data regarding the passenger responding to the survey. See 2.1.1.1 for explicit object details. |
| *FlightInformation* | No | Object (FlightInformation) | Data related to the flight used by the respondent of the survey. See 2.1.1.2 for explicit object details. |
| *AirportTransferInformation* | No | Object (AirportTransfer-Information) | Data regarding how the respondent traveled to the airport. See 2.1.1.3 |

| | | | |
|---|---|---|---|
| | | | for explicit object details. |
| *TripDetails* | No | Object (TripDetails) | Data related to the overall trip the respondent is traveling. See 2.1.1.4 for explicit object details. |
| *AirportStayDetails* | No | Object (AirportStay-Details) | Data regarding how the respondent behaved during their stay at the airport. See 2.1.1.5 for explicit object details. |
| *InterviewMethod* | Yes | Integer | Key representing which type of method that was used for this interview. See 2.1.2.6.1 for valid options. |
| *Weight* | Yes | String | Weight for the response. |
| *ReasonForRejection* | Yes | Integer | Key representing the reason why the respondent did not participate in the survey. See 2.1.2.6.2 for valid options. |

#### *2.1.1.1 PassengerInformation*

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *Age* | Yes | Integer | Key representing a range of ages, mapping to the age of the respondent. See 2.1.2.1.3 for explicit object details. |
| *Sex* | Yes | Integer | Key representing the sex of the passenger. See |

| | | | |
|---|---|---|---|
| | | | 2.1.2.1.1 for valid options. |
| *HomeCountryCode* | Yes | String | The country code where the respondent lives. |
| *HomePostalCode* | Yes | String | String with digits representing the postal code where the respondent lives. |
| *NoOfDomTripsR12* | Yes | Integer | Number of domestic trips the respondent has done the last twelve months. |
| *NoOfIntTripsR12* | Yes | Integer | Number of international trips the respondent has done the last twelve months. |
| *CopenhagenMalmoTravelsR12* | Yes | Integer | Number of trips the respondent has done that originates from Copenhagen and results in a flight from Sturup airport. |
| *TravelFrequency* | Yes | Integer | Key representing the category of travel frequency the respondent belongs to. See 2.1.2.1.2 for valid options. |

#### *2.1.1.2 FlightInformation*

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *TimeTableDeparture* | Yes | String (UTC-formatted DateTime string) | The date and time when the respondent's flight is scheduled to depart. |
| *ActualDepartureDate* | Yes | String (UTC-formatted DateTime string) | The date and time when the |

| | | | |
|---|---|---|---|
| | | | respondents flight actually departs. |
| *FlightId* | Yes | String | The id of the flight which the respondent is booked to fly. |
| *CodeShare* | Yes | String | Alternate flight-id if the booked flight is shared by multiple airlines. |
| *FlightType* | Yes | String | Indicates if the flight traveled by the respondent is an arrival (“A”) or departure (“D”). |
| *SubsequentDestination* | Yes | String | If the trip contains multiple transfers, this correlates to the airport IATA of the next transfer flight destination. |
| *FinalDestination* | Yes | String | The airport IATA of the final destination of the trip. |
| *Terminal* | Yes | Integer | Key representing the terminal where the flight is departing. See 2.1.2.2.1 for valid options. |
| *Gate* | Yes | String | The gate where the flight is departing from. |

#### *2.1.1.3 AirportTransferInformation*

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *TransportType* | Yes | Integer | Key representing the type of transport used for traveling to the airport. See 2.1.2.3.1 for valid options. |
| *Car* | No | Object (Car) | If the respondent was traveling to the |

| | | | |
|---|---|---|---|
| | | | airport with car, this property is used for additional details about the car. See 2.1.1.3.1 for explicit object details. |

2.1.1.3.1 Car

| *Property* | | *Data Type* | *Description* |
|---|---|---|---|
| *EnvironmentCar* | Yes | Integer | Key deciding whether the car used for traveling to the airport could be classified as an environment friendly car. See 2.1.2.3.2 for valid options. |
| *PeopleInCar* | Yes | Integer | Key representing how many passengers the car carried to the airport. See 2.1.2.3.4 for valid options. |
| *ParkingLot* | Yes | Integer | Key representing the parking lot where the car was parked. See 2.1.2.3.3 for valid options. |

#### *2.1.1.4 TripDetails*

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *ArrivalTimeToAirport* | Yes | String (UTC-formatted DateTime string) | Date and time when the respondent arrived at the airport. |
| *DaysOfTravel* | Yes | Integer | How many days the trip will last. |
| *DirectFlight* | Yes | Boolean | True if the respondent will reach their destination through a direct flight. |
| *Transfer* | Yes | Object (Transfer) | Leave null if respondent is using a direct flight. Contains additional information when the respondent is using transfer flights. See 2.1.1.4.1 for explicit object details |
| *CheckInType* | Yes | Integer | Key representing how the respondent was checking in. See 2.1.2.4.2 for valid options. |
| *PurposeOfTravel* | Yes | Integer | Key representing the purpose of the respondent's travel. See 2.1.2.4.3 for valid options. |
| *OriginAirportIata* | Yes | String | The IATA of the airport where the trip started. |
| *OriginPostalCode* | Yes | String | String with digits representing the postal code where the trip started. |

2.1.1.4.1 Transfer

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *MultipleTransfers* | Yes | Integer | Key describing however the transfer has multiple legs or not. See 2.1.2.4.1 for valid options. |
| *FirstPointOfTransfer* | Yes | String | Airport IATA of the first transfer flight destination. |
| *SecondaryTransferFlight* | Yes | String | Airport IATA of the second transfer flight destination. |

*2.1.1.5 AirportStayDetails*

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *VisitedFoodAndBeverages* | Yes | Boolean | True if the respondent has visited any establishment serving food or beverages. |
| *TakeAwayFood* | Yes | Integer | Key representing if the respondent did takeaway food or beverages from any establishment. See 2.1.2.5.3 for valid options. |
| *ConsumedFood* | Yes | [Integer] | Array of keys representing the types of food the respondent has consumed during the airport stay. See 2.1.2.5.1 for valid keys. |
| *FoodSpending* | Yes | [Object] (Spending) | An array of spending objects representing the food spending at the airport. One |

| | | | |
|---|---|---|---|
| | | | object represents the total amount of spending for one currency. See 2.1.1.5.1 for explicit object details. |
| *VisitedShops* | Yes | Boolean | True if the respondent has visited any shops. |
| *DidPurchase* | Yes | Boolean | True if the respondent did buy anything from a store at the airport. |
| *ShopTypesVisited* | Yes | [Integer] | An array of keys representing the different types of shops. See 2.1.2.5.2 for valid options. |
| *RetailSpending* | Yes | [Object] (Spending) | An array of spending objects representing the retail spending at the airport. One object represents the total amount of spending for one currency. See 2.1.1.5.1 for explicit object details. |
| *GroupSpendingDetails* | Yes | Object (GroupInfo) | An object containing information regarding the overall spending of the group the respondent is traveling together with. See 2.1.1.5.2 for explicit object details. |

2.1.1.5.1 Spending

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *SpendingAmount* | No | Integer | The total amount spent with this currency. |
| *SpendingCurrency* | Yes | String | The ISO 4217 code for the currency. |

2.1.1.5.2 GroupInfo

| *Property* | *Null* | *Data Type* | *Description* |
|---|---|---|---|
| *PeopleInGroup* | Yes | Integer | The number of people in the group. |
| *GroupSpending* | Yes | [Object] (Spending) | An array of spending objects representing the retail spending at the airport for the group. One object represents the total amount of spending for one currency. See 2.1.1.5.1 for explicit object details. |

2.1.2 Constants

Throughout the API some constants are used as keys for fixed value options. The keys are represented as integers. In the tables below, these keys are listed from the overall API contract structure.

*2.1.2.1 PassengerInformation*

2.1.2.1.1 Sex

| *Key* | *Value* |
|---|---|
| *1* | Male |
| *2* | Female |
| *3* | Other |
| *4* | Opt out |

2.1.2.1.2 TravelFrequency

| *Key* | *Value* |
|---|---|
| *1* | Every week |
| *2* | Once every month |
| *3* | Once every quarter |
| *4* | Once every six months |
| *5* | Once every year |
| *6* | Less often |
| *7* | Unknown |

2.1.2.1.3 Age

| *Key* | *Value* |
|---|---|
| *1* | 16 – 19 |
| *2* | 20 – 25 |
| *3* | 26 – 34 |
| *4* | 35 – 49 |
| *5* | 50 – 64 |
| *6* | 65+ |

### *2.1.2.2 FlightInformation*

2.1.2.2.1 Terminal

| *Key* | *Value* |
|---|---|
| *0* | Missing |
| *1* | Arlanda T2 |
| *2* | Arlanda T3 Domestic |
| *3* | Arlanda T4 Domestic |
| *4* | Arlanda T5a |
| *5* | Arlanda T5b |
| *6* | Arlanda T5f |
| *11* | Landvetter Domestic |
| *12* | Landvetter International |
| *13* | Sturup |
| *14* | Bromma |
| *15* | Luleå |
| *16* | Umeå |

### *2.1.2.3 AirportTransferInformation*

2.1.2.3.1 TransportType

| *Key* | *Value* |
|---|---|
| *-1* | N/A |
| *1* | Arlanda – Returning car (dropped off/picked up at the airport). |
| *10* | Arlanda – Upptåget |

| | |
|---|---|
| *100* | Arlanda – Shuttle bus from Stockholm. |
| *101* | Arlanda – Airport coach from Västerås. |
| *11* | Arlanda – Other train |
| *12* | Arlanda – Transfer flight |
| *14* | Arlanda – Conference at the airport |
| *15* | Arlanda – Other |
| *16* | Arlanda – Swebus from Stockholm |
| *17* | Arlanda – Stay at airport, arrived with other flight |
| *18* | Arlanda – Supershuttle bus. |
| *19* | Arlanda – SL Pendeltåg |
| *2* | Arlanda – Car parked at the airport |
| *21* | Landvetter – Returning car (dropped off/picked up at the airport). |
| *23* | Landvetter – Taxi or limo |
| *24* | Landvetter – Rental car |
| *25* | Landvetter – Airport bus from Göteborg |
| *26* | Landvetter – Airport bus from Borås |
| *27* | Landvetter – Other bus |
| *28* | Landvetter – Transfer flight |
| *3* | Arlanda – Taxi or limo |
| *30* | Landvetter – Conference |
| *31* | Landvetter – Other |
| *32* | Landvetter – Swebus from Göteborg |
| *33* | Landvetter – Stay at airport, arrived with other flight |
| *34* | Landvetter – Shuttle bus |
| *35* | Landvetter – Hotel |
| *4* | Arlanda – Rental car |
| *41* | Sturup – Returning car (dropped off/picked up at the airport). |
| *42* | Sturup – Car parked at the airport |
| *43* | Sturup – Airport bus from Malmö |
| *44* | Sturup – Airport bus from Lund |
| *47* | Sturup – Taxi or limo |
| *48* | Sturup – Rental car |
| *49* | Sturup – Hotel |
| *5* | Arlanda – Airport bus from Stockholm |
| *50* | Sturup – Tranfer flight |
| *51* | Sturup – Other |
| *52* | Sturup – Stay at airport, arrived with other flight |
| *53* | Sturup – Bus from Copenhagen |

| | |
|---|---|
| *54* | Sturup – Shuttle bus from Svedala |
| *55* | Sturup – Other bus |
| *6* | Arlanda – Bus from Uppsala |
| *61* | Bromma – Returning car |
| *62* | Bromma – Car parked at the airport |
| *63* | Bromma – Taxi or limo |
| *64* | Bromma – Rental car |
| *65* | Bromma – Airport bus from Stockholm |
| *66* | Bromma – Airport bus from Västerås or Enköping |
| *67* | Bromma – SL Bus |
| *68* | Bromma – Other bus |
| *69* | Bromma – Transfer flight |
| *7* | Arlanda – Other bus |
| *70* | Bromma – Stay at airport, arrived with other flight |
| *71* | Bromma – Other |
| *72* | Bromma – Tram |
| *8* | Arlanda – Arlanda Express |
| *81* | Luleå – Returning car |
| *82* | Luleå – Car parked at the airport |
| *83* | Luleå – Taxi or limo |
| *84* | Luleå – Rental car |
| *85* | Luleå – Airport bus |
| *86* | Luleå – Other bus |
| *87* | Luleå – Stay at airport, arrived with other flight |
| *88* | Luleå – Other |
| *89* | Luleå – Transfer flight |
| *9* | Arlanda – X200 Train |
| *91* | Umeå – Car returning (dropped off/picked up at the airport). |
| *92* | Umeå – Car parked at the airport |
| *93* | Umeå – Taxi or limo |
| *94* | Umeå – Rental car |
| *95* | Umeå – Airport bus |
| *96* | Umeå – Other bus |
| *97* | Umeå – Stay at airport, arrived with other flight |
| *98* | Umeå – Other |
| *99* | Umeå – Transfer flight |
| *20* | Arlanda – Hotel |
| *102* | Arlanda – Carpool |
| *103* | Arlanda – Airport Coach, door to gate. |
| *104* | Bromma – Carpool |

| | |
|---|---|
| *105* | Bromma – Bicycle |
| *106* | Landvetter – Carpool |
| *107* | Luleå – Bicycle |
| *108* | Umeå – Bicycle |
| *22* | Landvetter – Parked car |

2.1.2.3.2 EnvironmentCar

| *Key* | *Value* |
|---|---|
| *1* | Yes |
| *2* | No |
| *3* | Unknown |

2.1.2.3.3 ParkingLot

| *Key* | *Value* |
|---|---|
| *1* | Arlanda – Swedavia Multistorey |
| *-1* | N/A |
| *11* | Bromma – Swedavia Multistorey |
| *12* | Bromma – Bromma Blocks |
| *13* | Bromma – Other |
| *14* | Bromma – Swedavia parking outside, close to terminal. |
| *21* | Landvetter – Swedavia Multistorey |
| *23* | Landvetter – Swedavia outdoor car park |
| *24* | Landvetter – Other company car park |
| *31* | Sturup – Swedavia Multistorey |
| *33* | Sturup – Swedavia car park, no shuttle |
| *34* | Sturup – Other company car park |
| *4* | Arlanda – Other company car park |
| *5* | Arlanda – Swedavia outdoor parking, walking distance. |
| *7* | Arlanda – Outdoor car park shuttle |

2.1.2.3.4 PeopleInCar

| *Key* | *Value* |
|---|---|
| *0* | Unknown |
| *1* | One Person |
| *2* | Two Persons |
| *3* | Three Persons |
| *4* | Four Persons |
| *5* | Five Persons |
| *6* | Six Persons |
| *7* | Seven Persons |

#### *2.1.2.4 TripDetails*

##### 2.1.2.4.1 TransferType

| *Key* | *Value* |
|---|---|
| *1* | Transfer |
| *2* | Multiple transfers |

##### 2.1.2.4.2 CheckInType

| *Key* | *Value* |
|---|---|
| *1* | Check-in desk |
| *2* | Automatic machine |
| *3* | Mobile phone |
| *4* | Computer |
| *5* | At gate |
| *6* | VIP Service |
| *7* | Transfer |
| *8* | Other |

##### 2.1.2.4.3 PurposeOfTravel

| *Key* | *Value* |
|---|---|
| *1* | Leisure – Tourist |
| *2* | Leisure – Friends & relatives |
| *3* | Leisure – Other |
| *4* | Business – Meeting |
| *5* | Business – Conference |
| *6* | Business – Commute |
| *7* | Business – Other |
| *8* | Leisure – Family Property |
| *9* | Leisure – Rental property above 15 days |

#### *2.1.2.5 AirportStayDetails*

##### 2.1.2.5.1 FoodType

| *Key* | *Value* |
|---|---|
| *1* | Hot food |
| *2* | Salad, sandwich & yoghurt |
| *3* | Bun cake |
| *4* | Non-alcoholic beverage |
| *5* | Alcoholic beverage |
| *6* | No answer |

##### 2.1.2.5.2 ShopType

| *Key* | *Value* |
|---|---|
| *1* | Perfume & cosmetics |

| | |
|---|---|
| *2* | Wine, beer, alcohol & tobacco |
| *3* | Electronics & camera |
| *4* | Magazines |
| *5* | Books |
| *6* | Sweets & chocolates |
| *7* | Food delicatessen |
| *8* | Fashion, watches & clothes |
| *9* | Toys |
| *10* | Souvenirs & gifts |
| *11* | Other |
| *12* | No information |

2.1.2.5.3 TakeAwayFood

| *Key* | *Value* |
|---|---|
| *-1* | Missing |
| *0* | Did not order |
| *1* | Did order, but not takeaway. Everything was consumed at the restaurant. |
| *2* | Did order, only takeaway. |
| *3* | Did order, both takeaway and restaurant order. |

*2.1.2.6 SurveyResponse*

2.1.2.6.1 InterviewMethod

| *Key* | *Value* |
|---|---|
| *-1* | N/A |
| *1* | Passenger |
| *2* | Interviewer |

2.1.2.6.2 ReasonForRejection

| *Key* | *Value* |
|---|---|
| *1* | Refusal |
| *2* | Lack of time |
| *3* | Language difficulties |
| *4* | Other |
| *5* | Ok for interview |

2.1.3 Responses

Upon requesting the survey endpoint, the API will respond with an HTTP response-code after processing the request. The various codes all have different meaning, and marks if the request was successful. In addition to the HTTP

response-code, the response may contain a body with an error message. Below, all possible responses are explained.

#### *2.1.3.1 200 – OK*

If the request was accepted, the API will respond with HTTP-code 200. The request was valid, and the server could successfully receive and process the request.

#### *2.1.3.2 400 – Bad Request*

When clients perform requests with an invalid body, the API will respond with HTTP-code 400. The error could be considered to originate from the client. The API may in some cases provide an error message in the response body to help locate the part of the request which is considered to be invalid.

#### *2.1.3.3 500 – Internal Server Error*

If a valid request is performed from the client but any error occur on the server after the request is received, the API will respond with an HTTP status-code 500. The client has to send the request again to make sure the data is received by the server. A retry pattern on the client side is recommended to avoid temporary errors of this type. However, if the error persists for a longer time, contact api@swedavia.se for support.