

# attractions

Last updated by | morten.tranholm.jensen | Oct 31, 2024 at 10:30 AM GMT+1

---

## Attraction data Schema

The purpose of this document is to define a data schema for attraction data on the platform VisitData. The schema definition distinguishes between required and optional fields. Required fields are used in either benchmark calculations or in analytics.

If a client or platform makes data available through an external API, the attraction data schema should be used as a check-list. The data-structure on the client site, will likely not match the structure of the attraction schema. The important point is, that client data can be mapped to the attraction data schema.

If a client or platform sends data to the visitdata platform using the visitdata rest API, the data structure should adhere to the attraction data schema.

The data schema contains the following tables. The Ticket Transactions table contains all ticket transactions. Other transactions contains other transactions as souvenir and restaurant sales. The payments (sometimes called transaction header) contain payment details including customer details.

### Attraction Tables

- Ticket Transactions
- Other Transactions
- Payments
- Attraction metadata

## Attraction Schema Tables

### *Attraction metadata*

Attraction metadata. All attractions are mapped to a municipality, so we need either an address or a municipalityId to link an attraction to a specific municipality.

```
{
  "metaData": [
    {
      "attractionId": "123",
      "name": "Attraktion x",
      "address": "",
      "attractionChain": "",
      "municipalityId": "101",
    }
  ]
}
```



Field	Data type	Definition	Required
attractionId	string/guid	Unique attraction identifier	Yes
name	string	Name of the attraction	Yes
address	string	Attraction address. Required if municipalityId cannot be supplied	No
attractionChain	string	Name of the chain the attraction belongs to, if any.	No
municipalityId	integer	Municipality code	Yes

### ***Ticket transactions***

The ticket Transactions table contains all ticket transactions, including entry tickets and tickets to events and excursions outside of the attraction. Multiple transaction lines can be linked to a receipt. The table is linked to the payments table via the receiptId. The transactions table should also include entries from season ticket holders.

```
{
  "ticketTransactions": [
    {
      "transactionLineId": 1790,
      "receiptNumber": 123456,
      "transactionDate": "2024-10-11T09:49:00",
      "entryDate": "2024-10-11T00:00:00",
      "ticketId": 9812,
      "ticketName": "Voksen med rabat",
      "ticketType": "Voksen",
      "quantity": "1",
      "salesAmount": "100",
      "discountAmount": "25",
      "paidAmount": "75",
      "vatAmount": "0"
    }
  ],
  "attractionId": "123"
}
```



Field	Data type	Definition	Required
attractionId	integer/guid	ClientID of the attraction	Yes
transactionLineId	integer/guid	Transaction line identifier	Yes
receiptNumber	integer/guid	Receipt number	Yes
transactionDate	datetime	When the ticket is purchased. Formatted as yyyy-MM-ddThh:mm:ss	Yes
entryDate	datetime	The date of the entry, either actual or planned. formatted as yyyy-MM-ddThh:mm:ss	Yes
ticketId	integer	Ticket/product identified. Link to product table	No
ticketName	string	name of the ticket	Yes
ticketType	string	Ticket type, i.e. annual pass, adult, kids	Yes
quantity	integer	Number of tickets purchased	Yes
salesAmount	float	Line Amount	Yes
paidAmount	float	Paid line amount. Sales amount minus Discount	Yes
discountAmount	float	Line discount amount	Yes
vatAmount	float	Vat line amount	No

### Other Transactions

The table other transactions include all other transactions, that are not ticket sales. This includes cafe sales, restaurant sales and sales from souvenir shops etc.

```

{
  "otherTransactions": [
    {
      "transactionLineId": 1790,
      "receiptNumber": 123456,
      "transactionDate": "2024-10-11T09:49:00",
      "productId": 9812,
      "productName": "Voksen med rabat",
      "productType": "Voksen",
      "quantity": "1",
      "salesAmount": "100",
      "discountAmount": "25",
      "paidAmount": "75",
      "vatAmount": "0"
    }
  ],
  "attractionId": "123"
}

```



Field	Data type	Definition	Required
attractionId	integer/guid	ClientID of the attraction	Yes
transactionLineId	integer/guid	Transaction line identifier	Yes
receiptNumber	integer/guid	Receipt number	Yes
transactionDate	datetime	When the ticket is purchased. Formatted as yyyy-MM-ddThh:mm:ss	Yes
productId	integer	Product identifier. Link to product table	No
productName	string	Name of the product	Yes
productType	string	Product type	Yes
quantity	integer	Number of products purchased	Yes
salesAmount	float	Line Amount	Yes
paidAmount	float	Paid line amount. Sales amount minus Discount	Yes
discountAmount	float	Line discount amount	Yes
vatAmount	float	Vat line amount	No

## Payments

Payments or transaction header is the actual payment of the receiptnumber. The payment table is linked to the transactions tables via the receiptnumber. Payments include customer country and postalcode if possible.

```

{
  "payments": [
    {
      "receiptNumber": "123456",
      "totalAmount": "100",
      "transactionDate": "2023-11-11T14:49:00",
      "customer": {
        "country": "DK",
        "zipCode": "2300"
      },
      "salesChannel": "Web",
      "paymentMethod": "Card",
      "transactionType": "Sales"
    }
  ],
  "AttractionID": "123"
}

```



Field	Data type	Definition	Required
receiptNumber	integer/guid	Receipt number	Yes
totalAmount	float	Total receipt amount. Based on paid sales amount	Yes
customer.country	string	Customer country, iso 2 digit format	Yes
customer.zipCode	integer	Customer postal code	Yes
salesChannel	string	Sales channel i.e. web, shop. Distinguish web sales from physical sales.	No
paymentMethod	string	Payment method as cash, card. Categories defined by booking platform	No
transactionType	string	Transaction type, sales, return	No
transactionDate	datetime	Date time of payment. formatted as yyyy-MM-ddThh:mm:ss	Yes

Sales channel should include a value for web sales, Web. Other values is defined by the booking platform.

salesChannel
Web
BoxOffice

## Mapping and Language Model

As attractions and museums have many different ticket types and very different product hierarchies a language model is used to classify tickets and products. Transaction are classified according to categories by visitData.

The language model receives a ticket/product name and a ticket/product type. From this information the ticket or product is classified in two levels. Level 1 is the most generic, and displayed below. Level 2 is more detailed and includes sub-categories.

Level 1	Description
Billet	All entry transactions. Includes tickets to museums/attractions etc. Does not include tickets for activities or tours outside of the attraction.
Arrangementer & Events	Includes tickets to activities outside of the attraction including safaris, excursions etc
Mad og Drikke	Includes all transactions in the cafe or restaurant
Retail	Includes all transactions from the attraction store or similar

## Load Schedule

Data is loaded weekly. The load includes future bookings, i.e. for the next 6 months. In an initial data load, historic data is included, current year - 2 years. After the initial load, data is loaded using a delta load in which new or modified records are returned. Details around the delta load mechanism is discussed per integration.