

OsloCard

This document will describe the structure and the different messages that can be sent using OsloCard.

1 Changelog

Date	Document version	OsloCard Version	Comment
2017-06-01	1.0	1.0	Initial version

2 Table of contents

1	Changelog	1
2	Table of contents.....	1
3	Data types	1
4	Communication	2
4.1	The web service.....	2
5	Structure.....	2
6	Messages.....	2
6.1	Message To Drip	2
6.2	Message From Drip	3
7	DataPackets.....	3
7.1	Type: GetCardInformation.....	3
7.2	Type: ReportPerformedJourney	3
8	Examples.....	4
8.1	Example 1 - GetCardInformation	4
8.2	Example 2 – ReportPerformedJourney.....	5
9	Questions and answers.....	6

3 Data types

The data types used are a subset of the normal primitives found in .NET (which more or less correlates with the JAVA primitives). These data types will be used in this protocol:

Int	32-bit integer
-----	----------------

boolean	true or false
string	text
double	64-bit, floating point. Used for specifying fractions. Using dot (.) as separator (not comma)
datetime	Format for datetime is ISO 8601, for example "2001-01-01T12:00:00". The seconds-part can be omitted if precision not needed

4 Communication

Drip is publishing a web service that should be used to exchange messages between a central and the Drip-system. All communication is established and maintained by the central connecting to the server.

4.1 The web service

A web request is synchronous, meaning that a HTTP reply is always received (unless timeout), with data from the server-side. The service provided here is a "client-driven" service, meaning that communication and data exchange is only happening when requested by the client. The concept is that the client POSTs data to the webservice, and the reply received on the HTTP POST is the data from the server.

5 Structure

JSON is being used to format all messages.

- The order of the properties in the JSON-text is not guaranteed, and should never be assumed to arrive in any specific order.
- The property names are case-sensitive
- All text is encoded using UTF-8
- Line-breaks are CR+LF, or in code-terms: '\r\n'

To denote the type of the property, we will simply write an angle bracket < followed by the type and ended with the other angle bracket >, for example: <string> or <datetime>

6 Messages

6.1 Message To Drip

DESCRIPTION: This is the object to be used when sending information to Drip, it holds the central credentials and the payload. `PayloadType` is either "GetCardInformation" or "ReportPerformedJourney", depending on the type of packet sent.

```
{
  "Central":<string>,
  "Password":<string>,
  "PayloadType":<string>,
  "Payload": <DataPacket>
}
```

In the description of the messages that will follow below, we are excluding the Message for clarity - the only part we will print out when discussing the DataPackets below is the "<datapacket>".

6.2 Message From Drip

DESCRIPTION: This is the object that is used by Drip to send a reply to the central, it holds the type of the payload and the payload itself. The Faults-array may contain one or more Fault-objects, and if so, the IsSuccess will be false. The Faults-array contains all the faults that occurred, and should be inspected and processed.

The Payload *may be null*, and this should always be checked. Usually, if the Payload is null, there are one or more Faults.

```
{
  "Faults": [<Fault>],
  "IsSuccess": <boolean>,
  "PayloadType": <string>,
  "Payload": <DataPacket>
}
```

7 DataPackets

7.1 Type: GetCardInformation

DESCRIPTION: Sent by the central to Drip when information of a specific card is required.

```
{
  "CardNumber": <string>,
  "TravelType": <string>
}
```

Reply: GetCardInformationReply

DESCRIPTION: The reply received by the central from Drip upon sending GetCardInformation.

```
{
  "CardNumber": <string>,
  "IsValid": <bool>,
  "NbrJourneysLeft": <int>,
  "CustomerCostForJourney": <double>
}
```

7.2 Type: ReportPerformedJourney

DESCRIPTION: Sent by the central to Drip when a journey has been performed. `TravelType` is the type of journey. `Cost` is the price of the journey. `CustomerCost` is the sum paid by the customer.

```
{
  "CardNumber": <string>,
  "CarNumber": <string>,
  "StartAddress": <string>,
  "StartLocationLng": <double>,
  "StartLocationLat": <double>,
  "EndAddress": <string>,
  "EndLocationLng": <double>,
  "EndLocationLat": <double>,
}
```

```
"TravelType":<string>,  
"StartTime":<datetime>,  
"EndTime":<datetime>,  
"WaitTimeMinutes":<int>,  
"DistanceMeters":<int>,  
"Cost":<double>,  
"CustomerCost":<double>,  
"Sum":<double>  
}
```

Reply: ReportPerformedJourneyReply

DESCRIPTION: The reply received by the central from Drip upon sending ReportPerformedJourney.

The "JobId" contains the id from Drip of the journey that was sent to Drip. This is > 0 if it was successful. If 0, then inspect the Faults-array.

```
{  
  "JobId":<long>  
}
```

8 Examples

Note: the data in the examples are entirely fictional and matches real data only in type, not in format or value.

8.1 Example 1 - GetCardInformation

In this example a central wishes to query the server for information of a card.

The central sends a POST to the server containing:

```
{  
  "Central": "CentralTransportTest",  
  "Password": "PasswordPassword",  
  "PayloadType": "GetCardInformation",  
  "Payload":  
  {  
    "CardNumber": "87654321",  
    "TravelType": "ARB"  
  }  
}
```

The server will then reply with a GetCardInformationReply. The total reply will be:

```
{  
  "Faults": [],  
  "IsSuccess": true,  
  "PayloadType": "GetCardInformationReply",  
  "Payload":  
  {  
    "CardNumber": "87654321",  
    "IsValid": true,  
    "NbrJourneysLeft": 5,  
    "CustomerCostForJourney": 30.5  
  }  
}
```

Reply if GetCardInformation fails because of failed authentication, the HTTP status code will be 401 and the reply will look like this:

```
{
  "Faults":
  [
    {
      "FaultCode": "401",
      "FaultMessage": "Wrong username/password"
    }
  ],
  "IsSuccess": false,
  "PayloadType": "GetCardInformationReply",
  "Payload": null
}
```

8.2 Example 2 – ReportPerformedJourney

```
{
  "Central": "CentralTransportTest",
  "Password": "PWPWPassord",
  "PayloadType": "ReportPerformedJourney",
  "Payload":
  {
    "CardNumber": "123456",
    "CarNumber": "451",
    "StartAddress": "Långholmen 31",
    "StartLocationLng": 55.13,
    "StartLocationLat": 13.55,
    "EndAddress": "Stortinget 13",
    "EndLocationLng": 55.412,
    "EndLocationLat": 13.41932,
    "TravelType": "ARB",
    "StartTime": "2014-08-22T13:00",
    "EndTime": "2014-08-22T14:23",
    "WaitTimeMinutes": 3,
    "DistanceMeters": 2330,
    "Cost": 430.5,
    "CustomerCost": 48.0,
    "Sum": 400.0
  }
}
```

A successful reply:

```
{
  "Faults": [],
  "IsSuccess": true,
  "PayloadType": "ReportPerformedJourneyReply",
  "Payload":
  {
    "JobId": 30012
  }
}
```

An unsuccessful reply:

```
{
  "Faults":
  [
    {
      "FaultCode": "101",
      "FaultMessage": "No card found with number 123456"
    }
  ],
  "IsSuccess": false,
  "PayloadType": "ReportPerformedJourneyReply",
}
```

```
"Payload":null  
}
```

9 Questions and answers

Q: What section of the data on the magnetic stripe constitutes the card-number?

A: The full content of data stored on the magnetic stripe is
NNNNNNNNN03bdfkkkkkc=NNNNNNNNNNNNNNNNNN where 03bdfkkkkk is the part that
constitutes the card-number.

Q: What is "TravelType"?

A: Any of "FRI", "ARB", "AVL", "BHG", "DAG", "STØ", "UTD", "VGS", "VOX" or "X". Is used to
indicate the type of travel undertaken by the customer in case a card can be used for multiple
types. Can be omitted in which case "FRI" is assumed, if available. Usage of this field is
regulated by VEL.

Q: Which fields in ReportPerformedJourney are required?

A: CardNumber, CarNumber, StartAddress, StartTime, EndTime, DistanceMeters,
CustomerCost, Cost, StartLocationLng, StartLocationLat, EndLocationLng and
EndLocationLat. The other fields should be populated if the data is available.

Q: What is the field "Cost" in ReportPerformedJourney?

A: The full cost of the journey charged by the taxi company. The money paid by the customer
is not in any way included in this figure.

Q: What is the field "WaitTimeMinutes" in ReportPerformedJourney?

A: The number of minutes the car has been standing still during the journey.

Q: What is the field "Sum" in ReportPerformedJourney?

A: Should be set to 0 for now, may be used later.

Q: May "CarNumber" contain letters?

A: Yes

Q: Should VAT be included in "CustomerCost" and "Cost"?

A: VAT is to be included in "CustomerCost" but not in "Cost".

Q: In GetCardInformationReply, does the field "CustomerCostForJourney" include VAT?

A: Yes

Q: Is the data in the examples valid?

A: No, the data is entirely made up and matches real data only in type, not in format or value.

Q: How can we test the functions?

A: Contact DRT Solutions: info@drtsolutions.eu

* * * *