

# Accessing Birmingham Data on the oneTRANSPORT Service Portal using the oneTRANSPORT API

http://service.onetransport.io/

### Introduction

The oneTRANSPORT Data Marketplace provides a means for organisations to publish dynamic (ie regularly changing or real-time) data and Static resource files (ie Asset location / Dataset models) for discovery and consumption by others. The primary means for publishing and consuming data through the platform is via open system APIs. The Portal is a web service that enables developers to engage with the oneTRANSPORT service via a web browser, both to interact with data directly and to access detailed documentation about the system APIs.

## API Authentication

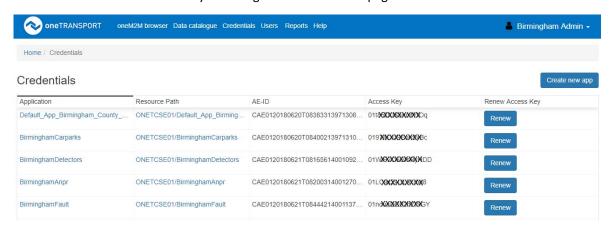
To utilize any of the oneTRANSPORT API's, you will need to authenticate each call to the API with an **AE-ID** and an **Access Key**. Both of these values are created when an Application is added to the portal. An Application within oneTRANSPORT is both the Application that publishes data, and the Application that consumes data. In this instance we refer to Application as that which is consuming the shared data.

Authentication is set within headers sent to the oneTRANSPORT API:

X-M2M-Origin: Your-applications-AE-ID

Authorization: Bearer Your-applications-access-key

These credentials can be found by browsing to the **Credentials** page in oneTRANSPORT:

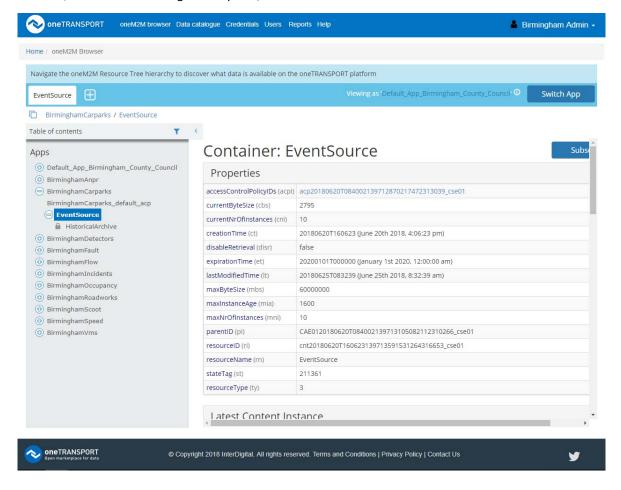


# **Retrieving Data**

Within oneTRANSPORT, data is organized into Containers. A Container can, for example, be a specific sensor, or a specific location such as a car park. Containers are defined by the publisher of the data, therefore you need to be able to discover and understand the data that you wish to consume. Data transactions are stored within oneTRANSPORT as Content Instances. A single Content Instance contains a single piece of data send by the publishing application.

We shall retrieve the latest Car Park data from the BirminghamCarparks Container.

Within the portal, browse to the oneM2M browser, find BirminghamCarparks in the resource tree browser, and click on BirminghamCarparks/EventSource:

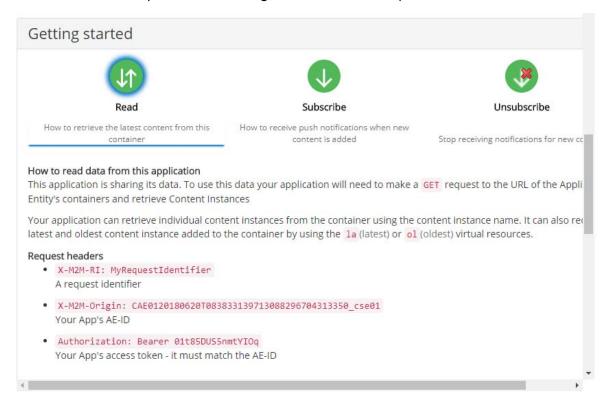


On the right hand pane, you will see the information for the Container, which in this instance is called EventSource.

Scroll down to see the Latest Content Instance. Here we can see specific information about the data, with the actual data being located within **content**:

contentinfo (cnf)	application/json
contentSize (cs)	279
creationTime (ct)	20180625T083239 (June 25th 2018, 8:32:39 am)
expirationTime (et)	20200101T000000 (January 1st 2020, 12:00:00 am)
lastModifiedTime (lt)	20180625T083239 (June 25th 2018, 8:32:39 am)
parentID (pi)	cnt20180620T160623139713591531264316653_cse01
resourceID (ri)	cin20180625T08323914001286169574472753_cse01
resourceName (rn)	AGoiyYbzBZnLcdeuoLfP1529915559749
stateTag (st)	211361
resourceType (ty)	4
content (con)	Event=UTMC:Carpark,EventCategory=UTMC,EventVariant=Carpark,ExitRate=0.0,FillRate=0.0,LastUpdate-25 09:30:58,Occupancy=1268,OccupancyPercentage=28.0,OccupancyTrend_TypeID=0,QueueTime=0,R 2806134,State_TypeID=999,System=Birmingham,SystemCodeNumber=Others-CCCPS133

Scroll down further and you will see a **Getting Started** section that explains how to access data:



Your Application is able to Subscribe to data. Subscribing to data means that you will receive a notification including the content (data) of the Content Instance. To do this, you would need to provide a HTTP End Point that oneTRANSPORT would be able to send data to.

In this tutorial, we shall be reading data from oneTRANSPORT, by sending a web service call asking for the latest Content Instance. To do this you will need to make a GET request to the URL of the Application and its Container, in this instance BirminghamCarparks/EventSource.

```
https://onem2m.onetransport.io/ONETCSE01/BirminghamCarparks/EventSource/la
```

You will notice the end of this request contains a virtual resource named **la**. This states "return me the **latest** Content Instance". You are also able to return the **oldest** Content Instance by specifying **ol**.

### Request headers

```
X-M2M-RI: A-identifier-for-this-transaction
X-M2M-Origin: Your-applications-AE-ID
Authorization: Bearer Your-applications-access-key
HTTP Response
```

The request will return a response indicating the status of the request. Successful requests will contain a X-M2M-RSC: 2000 header. The body of the response shall contain the oneM2M output. The "con" attribute will contain the data for this Content Instance.

```
HTTP/1.1 200 Content
Server: nginx/1.9.10
Date: Fri, 1 January 2018 10:11:35 GMT
Content-Type: application/vnd.onem2m-res+json
Content-Length: 235
Connection: keep-alive
X-M2M-RI: cin-retrieve
X-M2M-RSC: 2000
   "m2m:cin":{
      "cnf": "application/json",
      "con":"Event=UTMC:Carpark,EventCategory=UTMC,EventVariant=Carpark
,ExitRate=0.0,FillRate=2.0,LastUpdated=2018-06-25 09:50:16,Occupancy=18
4,OccupancyPercentage=50.0,OccupancyTrend_TypeID=3,QueueTime=0,Referenc
e=62807093, State_TypeID=2, System=Birmingham, SystemCodeNumber=BHMBCCTHL0
1",
      "cs":272,
      "ct": "20180625T085228",
      "et": "20200101T000000",
      "lt": "20180625T085228",
      "pi": "cnt20180620T160623139713591531264316653_cse01",
      "ri": "cin20180625T08522814001089225702473356_cse01",
      "rn": "zcHPzRgNtHbpUxkmuusW1529916748673",
      "st":212749,
      "ty":4
   }
}
```