CitySDK
Co-funded by the European Union

CitySDK Data Administration API

CitySDK Tourism Platform Installation Guide
André Oliveira (ISA)

Table of Contents

Introduction

This document describes the methods used to manipulate data to the CitySDK Platform, which are based on the RESTful API concept. Several request methods are available to upload, update and delete data elements (which can be POIs, Events or Routes) on the CitySDK API. Depending on the used request method (also called as HTTP Request Verbs), an appropriated action is performed. Table 1 describes the performed actions on each available HTTP Verb.

Table 1 - Actions table

HTTP Verb	Performed Action
PUT	Insert new data into the platform
POST	Update existent data in the platform
DELETE	Delete existent data in the platform
GET	Retrieve existent data in the platform

The API is protected against unauthorized data manipulation so, an authentication mechanism is available and described in detail in this document.

Authentication

In order to use operations that manipulate data in the platform require the requester must perform prior authentication in the platform. Failure to comply with the process will result in the error code 401 Unauthorized to be returned to the requester. After authenticated, the super-user may insert new data, update or delete any of the information present on the platform. If requester is a may perform new data insertion (by the use of the PUT) and update or delete data that which the authenticated user is in the authors list.

Authentication Roles

There are several authentication roles in the CitySDK platform:

- Super-User – Has permissions to perform any operation with any category or data element;

- Authenticated user – Authenticated users may perform insert operations (to insert new data elements or categories on the platform) and may also perform update/delete operations on any element in which they are listed as authors;

- Not authenticated – These users may only perform read operations.

Table 2 describes the roles and permissions on the CitySDK platform.

Table 2 - Permissions per role table

Action	Role			
	Super-User	Authenticated User (author)	Authenticated User (not author)	Not Authenticated
Insert	Allowed	Allowed	Allowed	Denied
Update	Allowed	Allowed	Denied	Denied
Delete	Allowed	Allowed	Denied	Denied
Retrieve	Allowed	Allowed	Allowed	Allowed

Authentication Procedure

Authentication is performed by login by sending a GET or POST request to the API on the endpoint shown on Table 3, replacing the template CITYSDK_API_ADDRESS with the effective IP address or hostname of the CitySDK platform.

Table 3 - Authentication endpoints

Action	URI

Authentication	https://CITYSDK_API_ADRESS/CitySDK/auth

The login request must contain the following parameters

Table 4 - Login operation parameters

Parameter name	Significance
username	Username of the user performing login on the API
password	Password of the user performing login on the API

The following example depicts a login request using a GET request to the API:

https://CITYSDK_API_ADDRESS/CitySDK/auth?username=admin&password=defaultCitySDKPassword

If the user and password is accepted, the API will reply with a 200 OK HTTP code and the following JSON object in the reply body, indicating the creation of a valid session:

```
{
  "SessionId": "1",
  "UserName": "admin",
  "ResponseStatus": {}
}
```

If the user tries to authenticate with invalid credentials, the API will reply with a 401 Unauthorized HTTP error code with the following body:

```
{
  "ResponseStatus": {
    "ErrorCode": "Invalid UserName or Password",
    "Message": "Invalid UserName or Password",
    "Errors": []
  }
}
```

The following table summarizes the authentication cases and their results.

Table 5 - Authentication cases

Authentication	HTTP Error code
Valid authentication	200 OK
Invalid authentication	401 Unauthorized

Data manipulation methods

This section describes the data manipulation operations used to insert, update and delete data elements from the platform.

Each one of the data element types (points of interest, events and routes) has a independent categories list. In order to create a new data element or change an existent one, it is necessary to ensure that the categories to which the data element belongs are previously created. Failure to comply with this condition will result in an error and the data element will not be committed to the platform.

In order to manipulate the existent data elements and categories, the appropriate URI must be used. The following table describes the suitable URI for the element to manipulate, replacing the CITYSDK_API_ADDRESS template with the real CitySDK platform's IP address or hostname.

Table 6 - List of URIs per data type

Data Element / Element Category	URI
POI Categories	http://CITYSDK_API_ADDRESS /CitySDK/categories?List=poi
Route Categories	http://CITYSDK_API_ADDRESS/CitySDK/categories?List=route
Event Categories	http://CITYSDK_API_ADDRESS/CitySDK/categories?List=event
POI Data Element	http:// CITYSDK_API_ADDRESS/CitySDK/pois/
Route Data Element	http://CITYSDK_API_ADDRESS/CitySDK/routes/
Event Data Element	http://CITYSDK_API_ADDRESS/CitySDK/events/

The requests involving new data inserts or data update must contain in their body the data formatted according to the W3C POI standard as described on CitySDK Deliverable D5.1.

Data Insertion

In order to insert new data in the API (data elements or categories), it is required that the requesting user is previously authenticated among the platform by using the process described in the Authentication section of this document.
Data insertion is performing by using the HTTP PUT verb, providing the data to insert in the request body.

When inserting a new data element, the element categories must previously exist and should be referred in the new element by its category ID (instead of the full category description)

Table 7 - Data insertion method description

Request verb	PUT
Request body	JSON object with the new data element to insert. For format specification refer to CitySDK Deliverable D5.1.
Expected reply	Success: HTTP OK result code with copy of the inserted object, containing the generated object ID in the reply body; Fail: HTTP Internal Server Error code with object containing the error and verbose description of the error in the reply body
Authentication conditions	User must be authenticated on the platform.
Remarks	All data element's categories must previously exist; All data element's categories must be represented in the object solely by its category ID.

Data Updating

Data elements or categories existing previously in the platform can be updated by using the HTTP POST verb in a request containing the updated version of the element to update in the request body. In order to update an element on the platform, the user must be previously authenticated and must be either the platform administrator or one of the authors registered on the element to update; failure to meet the authentication and conditions will cause the request to fail and therefore avoid the requested data update.

Table 8 - Data updating method description

Request verb	POST
Request body	JSON object with the updated data element (element ID should be the same). For object format specification refer to CitySDK Deliverable D5.1.
Expected reply	Success: HTTP OK result code with copy of the updated element in the body;

	Fail: HTTP Internal Server Error code with object containing the error and verbose description of the error in the reply body.
Authentication conditions	User must be authenticated on the platform. User must be either the platform administrator or one of the element authors
Remarks	All data element's categories must previously exist; All data element's categories must be represented in the object solely by its category ID.

Data Removal

Similarly with the previously described data manipulation methods, in order to delete existent data on the CitySDK platform, the requesting user must be previously authenticated on the platform and must be either the platform administrator or one of the authors registered on the element to delete; failure to meet the authentication and conditions will cause the request to fail and therefore not delete the requested element.

For data deletion, an HTTP DELETE request containing the object ID in the request body must be issued by the requester.

Table 9 - Data removal method description

Request verb	DELETE
Request body	JSON object with the element ID to delete.
Expected reply	Success: HTTP OK code with verbose message indicating success in the body; Fail: HTTP Internal Server Error code with object containing the error and verbose description of the error in the reply body
Authentication conditions	User must be authenticated on the platform. User must be either the platform administrator or one of the element authors
Remarks	The category removal process will also remove any reference to it on other data elements

Data Administration API Usage

This section describes some usage examples to help with the correct deployment of the platform.

Platform bootstrap procedure

In order to start populating a newly installed platform, the following phases should be used:

1. Insert POIs into the platform
    a. Creation of the categories for POIs
    b. Creation of the POIs in the platform

2. Insert Events and Routes into the platform
    a. Creation of the Events and Routes categories
    b. Creation of the Events and Routes in the platform

As described above, the first thing to insert into the platform are the POIs. These elements must be the first ones to insert because the other element types must reference the POIs (by their IDs) in their selves. The following sections describe in detail the process of insertion/update of the elements in the platform.

POIs administration

Before inserting a new POI into the platform, it is necessary to ensure that its category is already present in the platform. When submitting a new POI (or when updating an existent POI), its categories must be referenced by their IDs.

New POI example

The first task to do is to ensure that all the POI categories are already existent in the platform. If any of the categories is not existent in the platform, it must be created before. In order to create a new POI category, the PUT request should be used, having the category contents (formatted in JSON, according to the specifications) in its body.

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=poi
Request Type	PUT

Request Contents	{
	"list" : "poi",
	"category": {
	"label": [
	{
	"lang": "pt-PT",
	"term": "primary",

	"value": "Categoria teste"
	},
	{
	"lang": "en-GB",
	"term": "primary",
	"value": "test"
	}
	],
	"lang": "pt-PT",
	"term": "category",
	"value": "teste"
	}
	}

Request reply	"519b90344109bd0090f34603"

Figure 1 - Example request for POI Category creation

The above request will create a new category and return the newly created category ID (which can be used in the new POIs which belong to this category). This process must be repeated for each one of the new categories the POI belongs.

The next task to do is to insert the new POI into the platform. The next request illustrates the creation of a new POI, belonging to the previously created category.

Field	Values
Request URL	http://API_ADDRESS /CitySDK/pois
Request Type	PUT

Request Contents	{
	"poi":
	{
	"location":{
	"point":[
	{
	"Point":{
	"posList":"38.7623018491608 -9.09537155864194",
	"srsName":"http://www.opengis.net/def/crs/EPSG/0/4326"
	},
	"term":"entrance"
	}
	]
	},
	"label":[
	{
	"term":"primary",
	"value":"Primary Label"

```
      },
      {
        "term":"note",
        "value":"Some note"
      },
      {
        "term":"note",
        "value":"Another note"
      }
    ],
    "description":[
      {
        "value":"Description 1",
        "lang":"pt-GB"
      },
      {
        "value":"Descricao 1",
        "lang":"pt-PT"
      }
    ],
    "category":[
      {
        "id":"519b907c4109bd0090f34604"
      }
    ],
    "time":[
      {
        "term":"open",
        "value":"Opening time"
      }
    ],
    "link":[
     ],
    "base":"http://citysdk.url.pt/pois/",
    "lang":"pt-PT",
    "created":"2013-02-05T12:31:11.0000000Z",
    "author":{
      "term":"primary",
      "value":"CitySDK"
    },
    "license":{
      "term":"primary",
      "value":"open-data"
    }
  }
}
```

Request Reply

```
{
  "location": {
    "point": [{
      "Point": {
        "posList":          "38.7623018491608          -
9.09537155864194",
        "srsName":
"http://www.opengis.net/def/crs/EPSG/0/4326"
      },
      "term": "entrance"
    }]
  },
  "label": [{
    "term": "primary",
    "value": "Primary Label"
  }, {
    "term": "note",
    "value": "Some note"
  }, {
    "term": "note",
    "value": "Another note"
  }],
  "description": [{
    "value": "Description 1",
    "lang": "pt-GB"
  }, {
    "value": "Descricao 1",
    "lang": "pt-PT"
  }],
  "category": [{
    "id": "519b907c4109bd0090f34604"
  }],
  "time": [{
    "term": "open",
    "value": "Opening time"
  }],
  "link": [],
  "id": "519b960a1699b71234f1e0b6",
  "base": "http://citysdk.url.pt/pois/",
  "lang": "pt-PT",
  "created": "2013-05-21T15:43:06.9130205Z",
  "author": {
    "term": "primary",
    "value": "admin"
  },
  "license": {
    "term": "primary",
    "value": "open-data"
  }
}
```

Figure 2 - Example request for POI creation

The above request will create a new POI in the platform. If the POI is requested by a GET request, the following result will be presented to the requested:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/pois/
Request Type	GET
Request Contents	

Request Reply	{
	  "location":{
	    "point":[
	      {
	        "Point":{
	          "posList":"38.7623018491608 -9.09537155864194",
	
	"srsName":"http://www.opengis.net/def/crs/EPSG/0/4326"
	        },
	        "term":"entrance"
	      }
	    ]
	  },
	  "label":[
	    {
	      "term":"primary",
	      "value":"Updated Primary Label"
	    },
	    {
	      "term":"note",
	      "value":"Updated some note"
	    },
	    {
	      "term":"note",
	      "value":"Another note"
	    }
	  ],
	  "description":[
	    {
	      "value":"Description 1 updated",
	      "lang":"pt-GB"
	    },
	    {
	      "value":"Descricao 1",
	      "lang":"pt-PT"
	    }
	  ],
	  "category":[
	    {
	      "term":"category",
	      "value":"Categoria teste",

```
      "lang":"pt-PT"
    },
    {
      "term":"category",
      "value":"test",
      "lang":"en-GB"
    }
  ],
  "time":[
    {
      "term":"open",
      "value":"updated opening time"
    }
  ],
  "link":[

  ],
  "id":"519b960a1699b71234f1e0b6",
  "base":"http://citysdk.url.pt/pois/",
  "lang":"pt-PT",
  "updated":"2013-05-21T15:50:32.3940000Z",
  "created":"2013-05-21T15:43:06.9130000Z",
  "author":{
    "term":"primary",
    "value":"admin"
  },
  "license":{
    "term":"primary",
    "value":"open-data"
  }
}
```

Figure 3 - Retrieval of the inserted example POI

POI Update example

Taking the previously created POI, this example will perform an update on some of its fields. In order to perform the update, it is only requested to know the POI Id, which in this case is "519b960a1699b71234f1e0b6". All the fields of the POI will be updated to match the update request's defined POI.

Field	Values
Request URL	http://API_ADDRESS /CitySDK/pois/
Request Type	POST

Request Contents	{
	"poi":
	{
	"location":{
	"point":[
	{
	"Point":{

```
                    "posList":"38.7623018491608                    -
              9.09537155864194",

              "srsName":"http://www.opengis.net/def/crs/EPSG/0/4326"
                    },
                    "term":"entrance"
                  }
                ]
              },
              "label":[
                {
                  "term":"primary",
                  "value":"Updated Primary Label"
                },
                {
                  "term":"note",
                  "value":"Updated some note"
                },
                {
                  "term":"note",
                  "value":"Another note"
                }
              ],
              "description":[
                {
                  "value":"Description 1 updated",
                  "lang":"pt-GB"
                },
                {
                  "value":"Descricao 1",
                  "lang":"pt-PT"
                }
              ],
              "category":[
                {
                  "id":"519b907c4109bd0090f34604"
                }
              ],
              "time":[
                {
                  "term":"open",
                  "value":"updated opening time"
                }
              ],
              "link":[

              ],
              "base":"http://citysdk.url.pt/pois/",
              "id": "519b960a1699b71234f1e0b6",
        "lang":"pt-PT",
        "created":"2013-02-05T12:31:11.0000000Z",
        "author":{
          "term":"primary",
          "value":"CitySDK"
        },
        "license":{
          "term":"primary",
          "value":"open-data"
        }
      }
    }
```

Request Reply

```
{
   "location": {
      "point": [{
         "Point": {
            "posList":             "38.7623018491608            -
9.09537155864194",
            "srsName":
"http://www.opengis.net/def/crs/EPSG/0/4326"
         },
         "term": "entrance"
      }]
   },
   "label": [{
      "term": "primary",
      "value": "Updated Primary Label"
   }, {
      "term": "note",
      "value": "Updated some note"
   }, {
      "term": "note",
      "value": "Another note"
   }],
   "description": [{
      "value": "Description 1 updated",
      "lang": "pt-GB"
   }, {
      "value": "Descricao 1",
      "lang": "pt-PT"
   }],
   "category": [{
      "id": "519b907c4109bd0090f34604"
   }],
   "time": [{
      "term": "open",
      "value": "updated opening time"
   }],
   "link": [],
   "id": "519b960a1699b71234f1e0b6",
   "base": "http://citysdk.url.pt/pois/",
   "lang": "pt-PT",
   "updated": "2013-05-21T15:50:32.3940597Z",
   "created": "2013-05-21T15:43:06.9130000Z",
   "author": {
      "term": "primary",
      "value": "admin"
   },
   "license": {
      "term": "primary",
      "value": "open-data"
   }
}
```

Figure 4 - Example of POI update request

After the POST request has been issued, the new version of the POI will be served to the requesters.

POI delete example

This example will delete both the created POI and category created in the scope of this example.

When removing categories, if the category is associated with any POI, in order to maintain the platform's data consistency, it will be removed automatically from each one of the associated POIs, so the removed category will not appear in those POIs after the removal.

In order to remove the created POI it is necessary to issue the following DELETE request:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/pois/
Request Type	DELETE

Request Contents	{
	"id":"519b960a1699b71234f1e0b6"
	}

Request Reply	POI with id "519b960a1699b71234f1e0b6" removed successfully.

Figure 5 - Example of POI removal

In order to remove the created POI category it is necessary to issue the following DELETE request:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=poi
Request Type	DELETE
Request Contents	{ "id":"519b90344109bd0090f34603"

	}
Request reply	Category with id "519b90344109bd0090f34603" removed successfully.

Figure 6 - Example of POI category removal

Events Administration

This section illustrates the manipulation of events in the platform. In order to associate events to the city's POIs, it is necessary that the POIs have been created previously in the platform. It is also necessary to create the appropriate events categories in the platform before creating a event belonging to those categories.

New Event example

In order to create a new event, it is necessary to ensure that the event's hosting POI is already created and its ID is known (the creation process is described in the previous sections of this document). It is also necessary to ensure that the event's category is already in the platform before the event is created. In order to create an event category, a PUT request should be issued. The following example creates an example event category:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=event
Request Type	PUT

Request Contents	{
	 "list" : "event",
	  "category": {
	   "label": [
	    {
	     "lang": "pt-PT",
	     "term": "primary",
	     "value": "Categoria teste"
	    },
	    {
	     "lang": "en-GB",
	     "term": "primary",
	     "value": "test"
	    }
	   ],
	   "lang": "pt-PT",
	   "term": "category",
	   "value": "teste",
	   "base": "baseURL_Example"
	  }
	}

Request reply	"519ba2451699b71234f1e0b7"

Figure 7 - Example of event category creation

After the above request is issued, a new category was created for our example event. After this step, we can proceed to the creation of the event in the platform using the following PUT request.

Field	Values
Request URL	http://API_ADDRESS /CitySDK/events
Request Type	PUT

Request Contents	{
	"event":{
	"location":{
	"relationship":[
	{
	"targetPOI":"51966ba1c0eb8f28e87e8c8d",
	"term":"equal",
	"base":"http://tourism.citysdk.cm-lisboa.pt/pois/"
	}
	]
	},
	"label":[
	{
	"term":"primary",
	"value":"Lançamento Livro \"A imagem técnica e as suas crenças\"",
	"lang":"pt-PT"
	}
	],
	"description":[
	{
	"value":"Example Event",
	"lang":"en-GB"
	}
	],
	"category":[
	{
	"id":"519ba2451699b71234f1e0b7",
	}
	],
	"time":[
	{
	"term":"open",
	"value":"(iCalendar object contents)",
	"type":"text/icalendar"
	}
	],
	"base":"ExampleBase",
	"lang":"pt-PT",
	"created":"2012-11-13T14:27:34.0000000Z",
	"author":{
	"term":"primary",
	"value":"CitySDK"
	},
	"license":{
	"term":"primary",

```
      "value":"open-data"
    }
  }
}
```

Request reply

```
{
  "location": {
    "relationship": [{
      "targetPOI": "51966ba1c0eb8f28e87e8c8d",
      "term": "equal",
      "base": "http://tourism.citysdk.cm-lisboa.pt/pois/"
    }]
  },
  "label": [{
    "term": "primary",
    "value": "Lançamento Livro \"A imagem técnica e as suas crenças\"",
    "lang": "pt-PT"
  }],
  "description": [{
    "value": "Example Event",
    "lang": "en-GB"
  }],
  "category": [{
    "id": "519ba2451699b71234f1e0b7"
  }],
  "time": [{
    "term": "open",
    "value": "(iCalendar object contents)",
    "type": "text/icalendar"
  }],
  "id": "519ba7041699b71234f1e0b8",
  "base": "ExampleBase",
  "lang": "pt-PT",
  "created": "2013-05-21T16:55:32.6667754Z",
  "author": {
    "term": "primary",
    "value": "admin"
  },
  "license": {
    "term": "primary",
    "value": "open-data"
  }
}
```

Figure 8 - Example of event creation

After the above request has been issued, the new event is made available to the requesters by using the GET method.

Event Update example

In order to update any event existent in the platform, a POST request must be performed. In this example an update on the example event will be made.

To update the example event, the following request must be performed:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/events
Request Type	POST

Request Contents	{
	"event":{
	"location":{
	"relationship":[
	{
	"targetPOI":"51966ba1c0eb8f28e87e8c8d",
	"term":"equals",
	"base":"http://tourism.citysdk.cm-lisboa.pt/pois/"
	}
	]
	},
	"label":[
	{
	"term":"primary",
	"value":"Updated label",
	"lang":"pt-PT"
	}
	],
	"description":[
	{
	"value":"Updated Example Event",
	"lang":"en-GB"
	}
	],
	"category":[
	{
	"id":"519ba2451699b71234f1e0b7",
	}
	],
	"time":[
	{
	"term":"open",
	"value":"(iCalendar object contents)",
	"type":"text/icalendar"
	}
	],
	"base":"ExampleBase",
	"id":"519ba7041699b71234f1e0b8",
	"lang":"pt-PT",
	"created":"2012-11-13T14:27:34.0000000Z",

```
  "author":{
    "term":"primary",
    "value":"CitySDK"
  },
  "license":{
    "term":"primary",
    "value":"open-data"
  }
 }
}
```

Request reply

```
{
  "location": {
    "relationship": [{
      "targetPOI": "51966ba1c0eb8f28e87e8c8d",
      "term": "equals",
      "base": "http://tourism.citysdk.cm-lisboa.pt/pois/"
    }]
  },
  "label": [{
    "term": "primary",
    "value": "Updated label",
    "lang": "pt-PT"
  }],
  "description": [{
    "value": "Updated Example Event",
    "lang": "en-GB"
  }],
  "category": [{
    "id": "519ba2451699b71234f1e0b7"
  }],
  "time": [{
    "term": "open",
    "value": "(iCalendar object content)",
    "type": "text/icalendar"
  }],
  "id": "519ba7041699b71234f1e0b8",
  "base": "ExampleBase",
  "lang": "pt-PT",
  "updated": "2013-05-21T17:15:38.3745361Z",
  "created": "2013-05-21T16:55:32.6660000Z",
  "author": {
    "term": "primary",
    "value": "admin"
  },
  "license": {
    "term": "primary",
    "value": "open-data"
  }
}
```

Figure 9 - Example of event update

After the request has been issued, the new version of the event is served to the requesters.

Event Delete example

Similarly to the POI delete example explanation, in order to maintain data consistency, if a event category is removed from the platform, it will be also subtracted from each event that belonged to that removed category.

In order to remove the example event, the following DELETE request must be issued:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=event
Request Type	DELETE

Request Contents	{
	"id":"519ba7041699b71234f1e0b8"
	}

Request reply	Event with id "519ba7041699b71234f1e0b8" removed successfully.

Figure 10 - Example of event removal

In order to remove the example event category, the following DELETE request must be issued:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=event
Request Type	DELETE

Request Contents	{
	"id":"519ba2451699b71234f1e0b7"
	}

Request reply	Category with id "519ba2451699b71234f1e0b7" removed successfully.

Figure 11 - Example of event category removal

Routes Administration

This section exemplifies the administration of the routes in the platform. Similarly to the events, before the creation of the route, all its categories and POIs should be created.

Route Creation Example

Before the creation of a route, all its categories have to be present in the platform. In order to create a category, the following PUT request should be issued to the platform:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=route
Request Type	PUT

Request Contents	{
	"list" : "route",
	"category": {
	"label": [
	{
	"lang": "pt-PT",
	"term": "primary",
	"value": "Categoria teste"
	},
	{
	"lang": "en-GB",
	"term": "primary",
	"value": "test"
	}
	],
	"lang": "pt-PT",
	"term": "category",
	"value": "teste",
	"base": "baseURL_Example"
	}
	}

Request reply	"519bafc61699b71234f1e0b9"

Figure 12 - Example of route category creation

In order to create the route, the following PUT request must be issued:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/routes
Request Type	PUT

Request Contents	{
	"route":{
	"pois":[
	{
	"location":{
	"relationship":[
	{
	"targetPOI":"51966bd1c0eb8f28e87e9501",
	"term":"equals",
	"base":"http://tourism.citysdk.cm-lisboa.pt/pois/"
	}
	]
	},
	"label":[
	{
	"term":"primary",
	"value":"POI1 Name in the route context",
	"lang":"pt-PT"
	}

```
      ],
      "description":[
        {
          "value":"POI1 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI1 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966ba7c0eb8f28e87e8d98",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI2 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI2 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI2 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966befc0eb8f28e87e99ab",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
```

```
        }
      ]
    },
    "label":[
      {
        "term":"primary",
        "value":"POI3 Name in the route context",
        "lang":"pt-PT"
      }
    ],
    "description":[
      {
        "value":"POI3 PT Description in the route context",
        "lang":"pt-PT"
      },
      {
        "value":"POI3 EN Description in the route context",
        "lang":"en-GB"
      }
    ]
  }
],
"label":[
  {
    "term":"primary",
    "value":"PT Route name",
    "lang":"pt-PT"
  },
  {
    "term":"primary",
    "value":"EN Route name",
    "lang":"en-GB"
  }
],
"description":[
  {
    "value":"Route PT Description",
    "lang":"pt-PT"
  },
  {
    "value":"Route EN Description",
    "lang":"en-GB"
  }
],
"category":[
  {
       "id":"519bafc61699b71234f1e0b9"
  }
],
  "base":"http://tourism.citysdk.cm-lisboa.pt/routes/",
  "lang":"pt-PT",
  "created":"2013-05-17T17:42:16.3250000Z",
  "author":{
    "term":"primary",
    "value":"CitySDK"
  },
  "license":{
    "term":"primary",
    "value":"open-data"
  }
 }
}
```

Request reply

```
{
  "pois":[
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966bd1c0eb8f28e87e9501",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI1 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI1 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI1 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966ba7c0eb8f28e87e8d98",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI2 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI2 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI2 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966befc0eb8f28e87e99ab",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI3 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI3 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI3 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    }
  ],
  "label":[
    {
      "term":"primary",
      "value":"PT Route name",
      "lang":"pt-PT"
    },
    {
      "term":"primary",
      "value":"EN Route name",
      "lang":"en-GB"
    }
  ],
  "description":[
    {
      "value":"Route PT Description",
      "lang":"pt-PT"
    },
    {
      "value":"Route EN Description",
      "lang":"en-GB"
    }
  ],
  "category":[
    {
      "id":"519bafc61699b71234f1e0b9"
    }
  ],
  "id":"519be9b31699b715fcf2072e",
  "base":"http://tourism.citysdk.cm-lisboa.pt/routes/",
  "lang":"pt-PT",
  "created":"2013-05-21T21:40:03.0798942Z",
  "author":{
    "term":"primary",
    "value":"admin"
  },
  "license":{
    "term":"primary",
    "value":"open-data"
  }
}
```

Figure 13 - Route creation example

After this request has been issued, the route is served to the clients immediately.

Route Update Example

In order to update any route existent in the platform, a POST request must be performed. In this example an update on the example route will be made.

To update the example route, the following request must be performed:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/routes
Request Type	POST

Request Contents	{
	"route":{
	"pois":[
	{
	"location":{
	"relationship":[
	{
	"targetPOI":"51966bd1c0eb8f28e87e9501",
	"term":"equals",
	"base":"http://tourism.citysdk.cm-lisboa.pt/pois/"
	}
	]
	},
	"label":[
	{
	"term":"primary",
	"value":"POI1 Name in the route context",
	"lang":"pt-PT"
	}
	],
	"description":[
	{
	"value":"POI1 PT Description in the route context",
	"lang":"pt-PT"
	},
	{
	"value":"POI1 EN Description in the route context",
	"lang":"en-GB"
	}
	]
	},
	{
	"location":{
	"relationship":[
	{
	"targetPOI":"51966ba7c0eb8f28e87e8d98",
	"term":"equals",
	"base":"http://tourism.citysdk.cm-lisboa.pt/pois/"

```
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI2 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI2 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI2 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966befc0eb8f28e87e99ab",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI3 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI3 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI3 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    }
  ],
  "label":[
    {
      "term":"primary",
      "value":"PT Route name",
      "lang":"pt-PT"
    },
    {
      "term":"primary",
      "value":"EN Route name",
      "lang":"en-GB"
    }
  ],
  "description":[
    {
      "value":"Route PT Description",
      "lang":"pt-PT"
    },
    {
      "value":"Route EN Description",
      "lang":"en-GB"
    }
  ],
  "category":[
    {
         "id":"519bafc61699b71234f1e0b9"
    }
  ],
  "base":"http://tourism.citysdk.cm-lisboa.pt/routes/",
  "lang":"pt-PT",
  "created":"2013-05-17T17:42:16.3250000Z",
  "id":"519be9b31699b715fcf2072e",
  "author":{
    "term":"primary",
    "value":"CitySDK"
  },
  "license":{
    "term":"primary",
    "value":"open-data"
  }
 }
}
```

Request reply

```
{
  "pois":[
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966bd1c0eb8f28e87e9501",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"Updated POI1 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"Updated POI1 PT Description in the route
context",
          "lang":"pt-PT"
        },
        {
          "value":"POI1 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966ba7c0eb8f28e87e8d98",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI2 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI2 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI2 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    },
    {
      "location":{
        "relationship":[
          {
            "targetPOI":"51966befc0eb8f28e87e99ab",
            "term":"equals",
            "base":"http://tourism.citysdk.cm-
lisboa.pt/pois/"
          }
        ]
      },
      "label":[
        {
          "term":"primary",
          "value":"POI3 Name in the route context",
          "lang":"pt-PT"
        }
      ],
      "description":[
        {
          "value":"POI3 PT Description in the route context",
          "lang":"pt-PT"
        },
        {
          "value":"POI3 EN Description in the route context",
          "lang":"en-GB"
        }
      ]
    }
  ],
  "label":[
    {
      "term":"primary",
      "value":"PT Route name",
      "lang":"pt-PT"
    },
    {
      "term":"primary",
      "value":"EN Route name",
      "lang":"en-GB"
    }
  ],
  "description":[
    {
      "value":"Route PT Description",
      "lang":"pt-PT"
    },
    {
      "value":"Route EN Description",
      "lang":"en-GB"
    }
  ],
  "category":[
    {
      "id":"519bafc61699b71234f1e0b9"
    }
  ],
  "id":"519be9b31699b715fcf2072e",
  "base":"http://tourism.citysdk.cm-lisboa.pt/routes/",
  "lang":"pt-PT",
  "updated":"2013-05-21T21:49:47.1096174Z",
  "created":"2013-05-21T21:40:03.0790000Z",
  "author":{
    "term":"primary",
    "value":"admin"
  },
  "license":{
    "term":"primary",
    "value":"open-data"
  }
}
```

Figure 14 - Route update example

Immediately after the request is issued, information is updated.

Route Delete Example

Similarly to the POI delete example explanation, in order to maintain data consistency, if a route category is removed from the platform, it will be also subtracted from each event that belonged to that removed category.

In order to remove the example route, the following DELETE request must be issued:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/routes
Request Type	DELETE

Request Contents	{
	  "id":"519be9b31699b715fcf2072e"
	}

Request reply	Route with id "519be9b31699b715fcf2072e" removed successfully.

Figure 15 - Route removal example

In order to remove the example route category, the following DELETE request must be issued:

Field	Values
Request URL	http://API_ADDRESS /CitySDK/categories?List=route
Request Type	DELETE

Request Contents	{
	"id":"519bafc61699b71234f1e0b9"
	}

Request reply	Category with id "519bafc61699b71234f1e0b9" removed successfully.

Figure 16 - Route category removal example