

# HTTP API / JSON API

## Summary

Used by some services, such as Butler, to communicate with the server.

RS Production has a HTTP API that exposes the RS Production in memory database. The API calls are made over HTTP with standard Get / Put and Post methods, the data can be serialized in either XML or JSON format.

### Table of Contents

- [Summary](#)
- [Component settings](#)
- [JSON format](#)
- [Get data](#)
  - [Get\(EntityTypeID\)](#)
  - [Get\(EntityTypeID, filter\)](#)
- [Post / Create](#)
- [Put / Update](#)
- [Webhooks](#)
- [Troubleshooting](#)
- [Exists from](#)

## Component settings

- Datamanager Guid: 22f96847-e204-4c65-ac7a-f753cc90f7be

When installing the server component it is possible to set the base URL and the port that the service will bind to.

- baseURL: <http://somehost:port/somepath/>

The API will then be available at:

- Data: <http://somehost:port/somepath/api/data/>
- Webhook: <http://somehost:port/somepath/api/webhook/>

## Best practice and suggested base URL

- <http://somehost:80/<INSTALLATIONNUMBER>/>

Where <INSTALLATIONNUMBER> is the current installation-id.

## JSON format

All dates that are return from RS Production are UTC+0, to get the local time the date needs to be adjusted to the local timezone.

## Get data

There are two methods for getting data, one which takes an entity type id and returns all instances of the type and one where it is possible to filter instances by a lambda expression.

To get the data use method and content type:

- HTTP GET
- Content-Type: application/json

## Get(EntityTypeID)

Returns all instances of a specific entity type.

## Parameter

- `EntityTypeID`: GUID entity type id

## Call

- <http://somehost:port/somepath/api/data/nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn>

## Returns

A JSON-array with all instances of the supplied entity type.

### Get(EntityTypeID, filter)

Returns all instances of a specific entity type.

## Parameter

- `EntityTypeID`: GUID entity type id

## Filter

String a lambda expression to pre-filter the resulting instances.

## Call

- <http://somehost:port/somepath/api/data/nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn/filter/lambdaexpression>

### Lambda expression example

- `lambda x: x.Timestamp > DateTime.Now.AddMinutes(-20)`

The lambda expression contains `%20` and `%3A`, which are spaces and ':' respectively.

If we return the above "formatted" expression to the actual result we'll get: `"lambda x: x.Timestamp > DateTime.Now.AddMinutes(-20)"`. This expression is used in the mobile pages fetcher, where it tries to fetch mobile pages that were updated within the last 20 minutes.

This method should obviously not be used for complicated lambdas and there are better methods for passing on parameters and stuff into *webhooks*, see the [HTTP API / JSON API#webhooks](#) section below.

## Returns

JSON-array with all instances of the supplied entity type accepted by the supplied lambda filter.

## Post / Create

Creates a new instance of the specified EntityTypeID.

Method and content type:

- HTTP POST
- Content-Type: application/json

## Parameter (body of request)

```
{
  EntityTypeID: GUID,
  PropertyName1: PropertyValue1,
  PropertyName2: PropertyValue2,
  PropertyName3: PropertyValue3,
  PropertyNameN: PropertyValueN
}
```

## Call

Post request body to:

- <http://somehost:port/somepath/api/data/>

## Returns

Returns the created instance

## Put / Update

Updates the instance with the specified ID with the supplied property values.

Method and content type:

- HTTP PUT
- Content-Type: application/json

## Parameter (body of request)

```
{
  ID: GUID,
  PropertyName1: PropertyValue1,
  PropertyName2: PropertyValue2,
  PropertyName3: PropertyValue3,
  PropertyNameN: PropertyValueN
}
```

## Call

Put request body to:

- <http://somehost/somepath:port/api/data/>

## Returns

Returns the updated instance

## Webhooks

Executes a Python script at the server

- <http://somehost/somepath:port/api/webhook/>

The python script parameters are posted as a JSON object including the webhookid (ID of the python script to execute).

#### Parameter (body)

```
{
  webhookId: GUID
  ParameterName1: ParameterValue1,
  ParameterName2: ParameterValue2,
  ParameterName3: ParameterValue3,
  ParameterNameN: ParameterValueN
}
```

#### Call

Post request body to:

- <http://somehost/somepath:port/api/webhook/>

#### Returns

Returns the result of the executed python script.

---

#### Troubleshooting

Try surfing to <http://SERVERADDRESS:80/<INSTALLATION>/api/data/>

Example for 273566: <http://273566.rsproduction.se:80/273566/api/data/>

If the component is running and you're able to reach it, the response should be **"Welcome to RS Production HTTP-API, you must supply a entity type id in order to get any items."**

---

#### Exists from

17.x