Interface description atlasFX REST

Version 3.1

Last update 6/11/2015

Publisher:

alta4 AG

Fleischstraße 57
54290 Trier
Germany

Fon: +49.651.96626.0
Fax: +49.651.96626.26

www.alta4.com
info@alta4.com

Table of contents

1 Preliminary notes

The REST interface of atlasFX is typically addressed via `http`. If the atlasFX's proxy is protected by SSL encryption, `https` can be used, too. This documentation consistently uses http, although calls can also be made via `https`.

Note: In future you'll find the most recent version of this document here.

2 Search Operation

Note: In future you'll find the most recent version of this document here.

2.1 URL

URL
`http://<atlasFX-url>/spring/rest/maps/<mapId>/search`

Note: In future you'll find the most recent version of this document here.

2.2 Parent Ressource

Parent Ressource
Map

Note: In future you'll find the most recent version of this document here.

2.3 Description

Description
This operation provides a search functionality allowing the embedding websites to implement search field with a list of proposals in combination with icons and a link to an atlasFX map.

Note: In future you'll find the most recent version of this document here.

2.4 New in 1.9.2

New in 1.9.2
This interface is available since version 1.9.2.

Note: In future you'll find the most recent version of this document here.

2.5 Ressource Hierachy

Ressource Hierachy

- **MapList** http://<atlasFX-url>/spring/rest/maps
    - **Map** /<mapId>
        - **Search Operation** /search

Note: In future you'll find the most recent version of this document here.

2.6 Parameters

Parameters

Parameter	Details
searchTerm	Here, an input in the shape of a search term has to be provided by the user.
urlTemplate	In this parameter a template for the URL, which has to be called, must be provided. The possibilities to call are documented in the respective clients. This template may contain the three template variables `$mapId$`, `$layerId$` and `$featureId$`, which are set for each result individually. This mechanism allows maximum flexibility in configuring the map. Thus, all functions of the clients in combination with all network configurations are usable.
useDefaultIcons	This parameter is optional. It allows the usage of the atlasFX default icons, if no layer icon is configured. The Boolean values `true` or `false` are allowed.

Note: In future you'll find the most recent version of this document here.

2.7 JSON Response Syntax

JSON Response Syntax

```
[
  {
    "preview" : <previewText>,
    "url" : <mapClientUrl>,
    "iconId" : <iconId>
  } //, ...
]
```

Note: In future you'll find the most recent version of this document here.

2.8 Example

Example

Here is an example, which can be reproduced online at http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=530.

Explanation of the test system

`<atlasFX-url>`	`http://roadshow.alta4cloud.com/atlasfx`	Dies ist die Url unter der der Context der a Webapplication verfügbar ist.
`<mapId>`	`530`	In diesem Beispiel verwenden wir die Karte

		530. Diese Id identifiziert eine atlasFX-Kar einfach im atlasFX CMS abgelesen werde

Initially, the parameters have to be equipped. The first parameter searchTerm is trivial. A search term has to be passed here. For the usage of HTTP GET, parameters have to be encoded (URL encoding). The atlasFX backend uses UTF-8 for URI encoding. Therefore, umlauts like 'ö' have to be coded as `%C3%B6`.

As a second parameter a template must be passed for the URL. With this interface, any URL can be generated which are parameterized with values of the search result. In this example the atlasFX JS client simply has to be called with the detected object. The first challenge is that the atlasFX backend does not know the URL where it can be accessed. Therefore, we attach the `<atlasFX-url>` in the template parameter. In our case, it looks like this: `http://roadshow.alta4cloud.com/atlasfx`. Since the atlasFX-url is not enough to open the JavaScript-client, the suffix /js/index.html is added to the path. Now, the URL template is: `http://roadshow.alta4cloud.com/atlasfx/js/index.html`.

Furthermore, it is necessary to define a MapId. Here, our example MapId 500 goes into action. It is important to open the same map where the search has been performed because the results are assigned explicitly to a map. The MapId is passed as a URL parameter mapId. Further information concerning opening the client can be received in the documentation of the client. The Flex client can be opened in the same way, but a different URL has to be used. Now, the URL template looks like this: http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=530. To avoid errors, the template variable $mapId$ can also be used. This variable is replaced by the ID of the map where the search is performed: `http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=$mapId$`. After processing the request at the REST interface, the template variable is replaced with the current value. A request, which uses the map 530, will return the URL: http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=530. Mostly, the meaning and purpose of the search request is to show the object found in the GIS-context feature. Therefore, the feature which has to be shown must be passed to the map client. For this purpose, the atlasFX JS client provides the URL parameters mapLayerId and featureId. In atlasFX, a feature is clearly describable through the feature ID and the respective layer ID. These two ID are also available as template variables. We now complement our URL template to: `http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=$mapId$%26featureId=$featureId$%26mapLayerId=$layerId$`. Please note that the character `&` is escaped in the URL template with `%26` guaranteeing that it is not misinterpreted when calling the search interface.

The optional parameter `useDefaultIcons` is left out in the first example.

Example Parameter

Now we suppose that the user inputs 'Martin-Grundschule' into the search field.

Parameter	**example value**

`searchTerm`	`Martin-Grundschule`
`urlTemplate`	`http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=` `26featureId=$featureId$%26mapLayerId=$layerId$`

This results in the following call of the atlasFX REST interface
http://roadshow.alta4cloud.com/atlasfx/spring/rest/maps/530/search?searchTerm=Martin-Grundschule&urlTemplate=http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=$mapId$%26featureId=$featureId$%26mapLayerId=$layerId$"

The response of the search service

```
[
  {
    "iconId": 4,
    "preview": "Trier,GS Trier Martin",
    "url": "http://roadshow.alta4cloud.com/atlasfx/js/index.html?
mapId=530&featureId=657&mapLayerId=2889"
  }
]
```

It is noticeable that only one object has been returned. The result object contains the three properties `iconId`, `preview` and `url`. How to retrieve the icon based on its ID is explained in section icon ressource. The property preview contains a string describing the result. It can be displayed for instance in a suggestion list. The property `url` contains the URL, which is arised from variable replacement in the URL template. It can be called for example to start the map client.

Note: In future you'll find the most recent version of this document here.

2.9 Demo

A demo application is available.

Note: In future you'll find the most recent version of this document here.

3 Icon Ressource

Note: In future you'll find the most recent version of this document here.

3.1 URL

URL
`http://<atlasFX-url>/spring/rest/icons/<iconId>`

Note: In future you'll find the most recent version of this document here.

3.2 Parent Ressource

Parent Ressource
IconList

Note: In future you'll find the most recent version of this document here.

3.3 Description

Description
This ressource contains the image data of the icons.

Note: In future you'll find the most recent version of this document here.

3.4 New in 1.9.2

New in 1.9.2
This ressource is available since version 1.9.2.

Note: In future you'll find the most recent version of this document here.

3.5 Ressource Hierachy

Ressource Hierachy

- **IconList** `http://<atlasFX-url>/spring/rest/icons`
    - **Icon** `/<iconId>`

Note: In future you'll find the most recent version of this document here.

3.6 Parameters

Parameters
This ressource does not support URL parameters.

Note: In future you'll find the most recent version of this document here.

3.7 Response

Response

The image file is returned binary coded. The response header are set accordingly so that the URL can be used directly in a browser.

Note: In future you'll find the most recent version of this document here.

3.8 Example

Example

The following example can be reproduced online at http://roadshow.alta4cloud.com/atlasfx/js/index.html?mapId=530

Explanation of the test system

`<atlasFX-url>`	`http://roadshow.alta4cloud.com/atlasfx`	This is the url where the context of atlasF application is available.
`<iconId>`	`4`	In this example we try to call the icon 4.

The icon can simply be called at the URL http://roadshow.alta4cloud.com/atlasfx/spring/rest/icons/4.

Note: In future you'll find the most recent version of this document here.