

SysConf REST-API

V 1.0.0

This is the reference for the RESTful API to the SysConf provided by each Sportics Distribution. This API ist for developers who wants to integrate Sportics Services or who wants to query the current configuration of the distribution. The current version supports the retrieval of properties and translations.

Content

- 1 [Introduction](#)
- 2 [Basic Usage](#)
 - 2.1 [Getting URL](#)
 - 2.2 [Localization](#)
 - 2.3 [Authentication](#)
- 3 [API Commands](#)
 - 3.1 [GET sysconf/props](#)
 - 3.1.1 [Parameters](#)
 - 3.1.2 [Return](#)
 - 3.2 [GET sysconf/l10n](#)
 - 3.2.1 [Parameters](#)
 - 3.2.2 [Return](#)

Introduction

The SysConf REST-API is designed to access the System Configuration of a running distribution. Therefore three types of visibility are defined:

- private
This settings can't be accessed via the REST-API
- protected
Access to this settings requires authentication and the appropriate role (defined via SysConf) must be granted to the authenticated requester
- public
Access to this settings requires no authentication

The values can be accessed via their defined keys or via tags. With tags a set of suitable configuration data will is grouped to reduce the number of requests.

Basic Usage

For GET/POST requests, use the following [Accept](#) and [Content-Type](#) headers:


- [Accept](#): application/json
- [Content-Type](#): application/json

Getting URL

To access the real URL you can send a bootstrap call, and get the URI for the component **SysConf API** (Line 21).

bootstrap example output

```
{
  "System":{
    "State":"online",
    "Name":"Sportics Hub",
    "Version":"1.0.0",
    "URI":"http://hub.sportics.net"
  },
  "Components":{
    "Converter":{
      "State":"online",
      "Version":"1.10.0"
    },
    "DT API":{
      "state":"online",
      "version":"1.09.0",
      "URI":"http://hub.sportics.net/dt"
    },
    "SysConf API":{
      "state":"online",
      "version":"1.09.0",
      "URI":"http://hub.sportics.net/sysconf"
    }
  },
  "Principal":{
    "name":"sandbox",
    "partner":[
      "a",
      "b"
    ]
  },
  "Actor":{
  }
}
```

 In this API documetation \$URI will be used as placeholder for the real URL.

Localization

To receive text in languages other than English, pass the appropriate HTTP header [Accept-Language](#). For example, to receive text in German use

[Accept-Language](#): de

Authentication

This API can be accessed authorized and unauthorized.

- Unauthorized Request
Only settings, defined as *public* will be returned.
- Authorized Request
Use [basic access authentication](#). Settings, defined as *public* or *protected* will be returned.

See [Authentication Policy](#) for details.

API Commands

GET sysconf/props

The properties are simply values for defined configuration keys. The concrete value depends on some parameters, e.g. the principal.

URI	\$URI/sysconf/props
-----	---------------------

Parameters

Name	Binding	Note	Example
principal	Depends	<ul style="list-style-type: none"> • Mandatory, if the distribution supports more principals. • Without effect, if the distribution doesn't support more principals. 	principal=sportics
component	Optional	<ul style="list-style-type: none"> • Restricts the value set to a specific component 	component=Converter
keys	Depends	<ul style="list-style-type: none"> • Defining keys for which the values are requested. More keys are separated via + or white space • Optional, if "tags" parameter is also given. Mandatory, if "tags" parameter is not given 	keys=file.upload.sizeLimit+sys.licence.max_user

tags	Depends	<ul style="list-style-type: none">• Per Tag, one ore more keys are combined and accessible in one request• Optional, if "keys" parameter is also given. Mandatory, if "keys" parameter is not given	tags=mst+rt
------	---------	--	-------------

Return

Data

The JSON Data seperates the requested keys and values as shown in the following example

```
{
  "aTag": [
    {
      "key1": "value 1"
    },
    {
      "key2": "value 2"
    }
  ],
  "anotherTag": [
    {
      "key 3": "value 3"
    },
    {
      "key 1": "value 1"
    }
  ],
  "aKey": "aValue",
  "anotherKey": "anotherValue"
}
```

HTTP Codes

GET sysconf/l10n

Getting localized resources like strings, or formatter. The localization depends on the principal (if the distribution supports multi-principals)

URI	\$URI/sysconf/l10n
-----	--------------------

Parameters

Name	Binding	Note	Example
------	---------	------	---------

principal	Depends	<ul style="list-style-type: none"> • Mandatory, if the distribution supports more principals. • Without effect, if the distribution doesn't support more principals. 	principal=sportics
lang	Optional	<ul style="list-style-type: none"> • If no lang parameter is given, the service try to match on the languages, specified via Accept-Language in the HTTP header. This parameter only accepts one language. It's not possible to concatenate more languages. 	lang=de
keys	Depends	<ul style="list-style-type: none"> • Defining keys for which the values are requested. More keys are seperated via + or white space • Optional, if "tags" parameter is also given. Mandatory, if "tags" parameter is not given 	keys=sport.sport_archery_run.label+file.upload.cancel
tags	Depends	<ul style="list-style-type: none"> • Per Tag, one ore more keys are combined and accessible in one request • Optional, if "keys" parameter is also given. Mandatory, if "keys" parameter is not given 	tags=mst+rt

Return

Data

The JSON Data separates the requested keys and values as shown in the following example

```
{
  "aTag":[
    {
      "key1":"value 1"
    },
    {
      "key2":"value 2"
    }
  ],
  "anotherTag":[
    {
      "key 3":"value 3"
    },
    {
      "key 1":"value 1"
    }
  ],
  "aKey":"aValue",
  "anotherKey":"anotherValue"
}
```

HTTP Codes