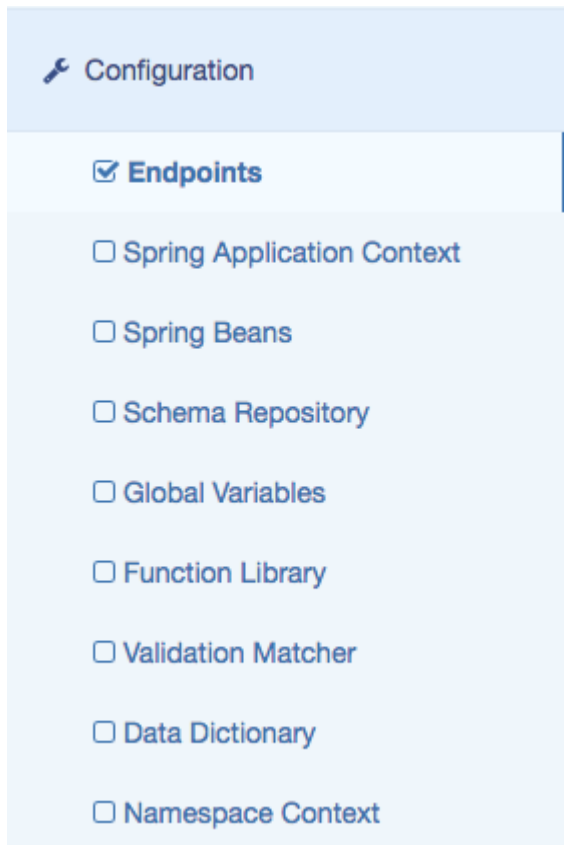


# Configuration

# Table of Contents

Endpoint configuration .....	1
Spring application context.....	11
Spring beans .....	12
Global Variables .....	12
Schema Repositories .....	12
Namespace Context .....	12
Data Dictionaries .....	12
Functions .....	12
Validation Matcher .....	12

The Citrus components such as endpoints, variables, functions, schemas and dictionaries are configured in a Spring application context. The administration UI is able to read and change the Citrus components configuration. Each component category is represented with a separate page in the configuration section.



## Endpoint configuration

Endpoints are essential in a Citrus project. They define client and server components as well as producer and consumer for different message transports.

s for sending and receiving messages to/from various endpoint destinations. Endpoint components can be of synchronous and asynchronous messages.

	Properties
Client	<b>destination-name:</b> JMS.Citrus.v1.FieldForceOrder <b>pub-sub-domain:</b> false <b>timeout:</b> 5000
Endpoint	<b>destination-name:</b> JMS.Citrus.v1.FieldForceNotification <b>pub-sub-domain:</b> false <b>timeout:</b> 5000
	<b>request-url:</b> http://localhost:18001/incident/IncidentManager/v1 <b>interceptors:</b> clientInterceptors <b>fault-strategy:</b> throwsException <b>timeout:</b> 5000
Server	<b>port:</b> 18002 <b>auto-start:</b> true <b>root-parent-context:</b> true <b>interceptors:</b> serverInterceptors <b>timeout:</b> 10000
	<b>port:</b> 18005 <b>auto-start:</b> true <b>root-parent-context:</b> true <b>handle-mime-headers:</b> true <b>timeout:</b> 10000

s for sending and receiving messages to/from various endpoint destinations. Endpoint components can be of synchronous and asynchronous messages.





















	Properties
Client	<b>destination-name:</b> JMS.Citrus.v1.FieldForceOrder <b>pub-sub-domain:</b> false <b>timeout:</b> 5000
Endpoint	<b>destination-name:</b> JMS.Citrus.v1.FieldForceNotification <b>pub-sub-domain:</b> false <b>timeout:</b> 5000
	<b>request-url:</b> http://localhost:18001/incident/IncidentManager/v1 <b>interceptors:</b> clientInterceptors <b>fault-strategy:</b> throwsException <b>timeout:</b> 5000
Server	<b>port:</b> 18002 <b>auto-start:</b> true <b>root-parent-context:</b> true <b>interceptors:</b> serverInterceptors <b>timeout:</b> 10000
	<b>port:</b> 18005 <b>auto-start:</b> true <b>root-parent-context:</b> true <b>handle-mime-headers:</b> true <b>timeout:</b> 10000

ng and receiving messages to/from various endpoint destinations. Endpoint components can be of synchronous and asynchronous

	Properties
	<b>destination-name:</b> JMS.Citrus.v1.FieldForceOrder <b>pub-sub-domain:</b> false <b>timeout:</b> 5000
	<b>destination-name:</b> JMS.Citrus.v1.FieldForceNotification <b>pub-sub-domain:</b> false <b>timeout:</b> 5000
	<b>request-url:</b> http://localhost:18001/incident/IncidentManager/v1 <b>interceptors:</b> clientInterceptors <b>fault-strategy:</b> throwsException <b>timeout:</b> 5000
	<b>port:</b> 18002 <b>auto-start:</b> true <b>root-parent-context:</b> true <b>interceptors:</b> serverInterceptors <b>timeout:</b> 10000
	<b>port:</b> 18005 <b>auto-start:</b> true <b>root-parent-context:</b> true <b>handle-mime-headers:</b> true <b>timeout:</b> 10000

First of all the list of all available endpoints in the project is displayed. Each endpoint represents a message transport such as SOAP, JMS, REST, Mail, FTP and so on. You can add new endpoints using the *New* context menu on the right. You need to chose the endpoint type first. Then a HTML form is displayed holding the endpoint settings.

New ▾

-  camel
-  channel
-  ftp-client
-  ftp-server
-  http-client
-  http-server
-  jms
-  jmx-client
-  jmx-server
-  mail-client
-  mail-server
-  rmi-client
-  rmi-server
-  ssh-client
-  ssh-server
-  vertx
-  ws-client
-  ws-server
-  websocket-client
-  websocket-server

http-client

ST

propagateError

)

00

ve

Close



http-client

ST

propagateError

)

00

ve

Close

[illegible]

Click save to add the new endpoint. Citrus is working with Spring XML configuration files. This means that the new endpoint component is saved as XML Spring bean to the basic Spring application context file. Usually this is a file located in `src/test/resources/citrus-context.xml` in your project. After you have saved the new component you will see that a new entry has been added to this file.

You can also edit endpoint components in the administration UI. Just click an existing endpoint

[illegible]

9

client

entHttpClient

/localhost:18001/incident/IncidentManager/v1

Interceptors

sException

Close

Manager/v1

If you save the changes Citrus will again change the Spring bean component in the XML configuration file. You can manually review the changes made. All manual changes in the Spring application context will also affect the administration UI. Just hit the reload button in your browser to reload the configuration.

## Spring application context

TODO

# Spring beans

TODO

# Global Variables

TODO

# Schema Repositories

TODO

# Namespace Context

TODO

# Data Dictionaries

TODO

# Functions

TODO

# Validation Matcher

TODO