

# MashZone NextGen Explorer Administration Guide

Version 10.1

October 2017

---

This document applies to MashZone NextGen Explorer Version 10.1 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2016-2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

---

# Table of Contents

<b>Preface</b> .....	<b>5</b>
<b>Create user password</b> .....	<b>7</b>
<b>Manage user privileges</b> .....	<b>9</b>
Create user.....	10
Delete user.....	11
Change user password.....	11
Change user role.....	12
<b>Apache Spark Integration</b> .....	<b>13</b>
Advanced Configuration.....	14
Deployment of Spark on a separate machine.....	15
Importing larger data sources.....	15
<b>Set SSL configuration</b> .....	<b>17</b>
Enabling TLS.....	18
Setup of certificates and key material.....	18
Enabling TLS with JKS-based key material.....	19
Enabling TLS with PEM-based key material.....	19
Enabling TLS with PKCS#2-based key material.....	19
<b>Configure SSO with MashZone NextGen</b> .....	<b>21</b>
<b>Configure startup behaviour</b> .....	<b>23</b>
<b>Add custom JAR files</b> .....	<b>27</b>
<b>Configure DES data source</b> .....	<b>29</b>
Configuration of the DesEventBroker microservice.....	30
DesEventBroker microservice API.....	31
Subscribe method.....	31
Unsubscribe method.....	37
Usage.....	38
Configuration of the real-time data source.....	39



---

## Preface

---

The MashZone NextGen Administration Guide includes information for administrators to configure and manage MashZone NextGen Explorer.



# 1 Create user password

Using the MashZone NextGen Explorer Console you are able to create an encrypted password.

You can use the password randomly generated for the MashZone NextGen Explorer user authentication.

## Procedure

1. Click the  **Console** icon in the program bar.
2. Click **User Service** in the **Command** bar.

The subordinated commands are expanded.

3. Click the **hashPassword** command.
4. Enter your password into the **Payload** field.
5. Click the **Invoke** button.
6. Copy the password without the double quotes to clipboard.

The encrypted password is generated and displayed in the **Response** field. The password starts with **\$** and ends with **=** and is set in double quotes (**"**).

## Example

```
"$shiro1$SHA-256$500000$BK02RfNycT5X9+SYFOw==$WVnmao8JwY5+KlOfodA="
```

You can copy the password (without the double quotes) to clipboard and use it for, e.g, ["creating a new " on page 10"](#) ["MashZone NextGen Explorer" on page 10"](#) ["user" on page 10"](#).



## 2 Manage user privileges

---

■ Create user .....	10
■ Delete user .....	11
■ Change user password .....	11
■ Change user role .....	12

MashZone NextGen Explorer enables you to create new users, change their passwords and roles, and delete existing users.

Depending on the user role assigned the user has different access permissions for data sources and analysis.

By default, the following user roles with the corresponding permissions are available.

- administrator:
  - Full control of all data sources and analysis.
- analyst:
  - Full Create, Read, Update and Delete permissions for data sources and analysis owned by the user or if users has an **EDIT** permission.
  - Read only control on data sources and analysis if **VIEW** permission is present.
- guest:
  - Read only permissions for analysis if **VIEW** permission is present.

The MashZone NextGen Explorer users available and roles assigned are listed in the `auth.properties` file stored in the `<MashZone NextGen Explorer installation>\config` folder.

The users are listed with username, password and user role in the form of:

```
user.<username> = <password encrypted>,<user role>,va
```

### Example

```
user.Administrator = $shiro1$SHA-256$500000$BK02RfNycT5X9+SYFOw==$WVnmao8JwY5+KlOfodA=,administrator,va
```

## Create user

You can create new MashZone NextGen Explorer users and assign the relevant permissions.

To create a new MashZone NextGen Explorer user you have to specify the user name, user password, and the user role. Using the MashZone NextGen Explorer Console you are able to create the required user password. See "[Create user password](#)" on page 7 for details.

### Procedure

1. Open the `auth.properties` file in an appropriate text editor.
2. Add a user to the user list in the following form.

```
user.<username> = <password encrypted>,<user role>,va
```

- Replace the `<username>` variable by a unique user name.

- Replace the <password encrypted> variable by the password generated (without quotes).
  - Replace the <user role> variable by the user role with the access permissions required.
3. Save your changes.
  4. Restart the MashZone NextGen Explorer server.

The user is created and can log in to MashZone NextGen Explorer using the appropriate user name and user password.

## Delete user

---

You can delete an already existing MashZone NextGen Explorer user.

### Procedure

1. Open the auth.properties file in an appropriate text editor.
2. In the user list, delete the row with the relevant user name.
3. Save your changes.
4. Restart the MashZone NextGen Explorer server.

The user is deleted and can not longer log in to Visual Analytics.

## Change user password

---

You can change the password of an existing MashZone NextGen Explorer user.

Using the MashZone NextGen Explorer Console you are able to create the new user password. See "[Create user password](#)" on page 7 for details.

### Procedure

1. Open the auth.properties file in an appropriate text editor.
2. In the user list, replace the existing password of an user with the new one.

The password is encrypted, starts with \$, and ends with =.

3. Save your changes.
4. Restart the MashZone NextGen Explorer server.

The password is saved and the user can log in to MashZone NextGen Explorer using the appropriate user name and user password.

## Change user role

---

You can change the user role assigned to an existing Visual Analytics user.

### Procedure

1. Open the auth.properties file in an appropriate text editor.
2. In the user list, replace the user role assigned to an user.

You can assign the **administrator**, **analyst** and **guest** user role. See "[Manage user privileges](#)" on page 9.

3. Save your changes.
4. Restart the MashZone NextGen Explorer server.

Your changes are applied and the user can log in to Visual Analytics with the new user role assigned.

# 3 Apache Spark Integration

---

- Advanced Configuration ..... 14
- Deployment of Spark on a separate machine ..... 15

You can use Apache Spark to process large CSV and JSON source files in MashZone NextGen Explorer.

Apache Spark is a fast and general-purpose cluster computing system. MashZone NextGen Explorer version 10.0 ships with Apache Spark and integrates it to process large CSV and JSON source files and execute queries on those data sources. XML source files are currently not supported. When dealing with bigger source files, the query execution time is lower compared to MashZone NextGen Explorer default engine. Thus, the overall performance of MashZone NextGen Explorer will benefit in those cases. As a guidance, data sets with 500,000 lines and above should be processed using the now-integrated Spark engine.

By default, the MashZone NextGen Explorer installation runs the Visual Analytics Server without Apache Spark integration. To enable Apache Spark you have to perform the following steps.

#### Procedure

1. Open the config.json configuration file in a text editor.
2. Add the following code line and save your changes.

```
"spark.enabled" : true,
```

3. To run MashZone NextGen Explorer with Spark open a command line program and run `<MashZone NextGen Explorer installation>\bin\vaserver --embeddedSpark`

The default installation path is `C:\SoftwareAG\MashzoneNG\VisualAnalytics`.

MashZone NextGen Explorer runs with Apache Spark.

## Advanced Configuration

By default, Apache Spark engine utilizes up to four CPU cores and 1 GB of RAM per core. If more resources are available on the machine running MashZone NextGen Explorer, you can configure the maximal number of CPU cores and RAM to be used by Spark.

#### Procedure

1. Open the config.json configuration file in a text editor.
2. Add the following code line and save your changes.

```
"spark.cores.max" : <number of cores>,
"spark.executor.memory" : "<amount of memory>",
```

Your settings are applied.

Thereby, `<number of cores>` is a numeric value, such as 3, 7, 15. It is recommended to not assign all cores of the machine to Spark, i.e., when `N` is the number of cores of the machine, then use `N-1` cores at max for Spark.

<amount of memory> is a numeric value followed by the unit, e.g., 1g for for 1 GiByte, 512m for 512 MiB etc.

**Note:** The amount of memory is defined per CPU core, i.e., 1g with 16 cores equals an allowed total memory consumption of 16 GiB of RAM.

Allowed units are

```
1b (bytes)
1k or 1kb (kibibytes = 1024 bytes)
1m or 1mb (mebibytes = 1024 kibibytes)
1g or 1gb (gibibytes = 1024 mebibytes)
1t or 1tb (tebibytes = 1024 gibibytes)
1p or 1pb (pebibytes = 1024 tebibytes)
```

## Deployment of Spark on a separate machine

MashZone NextGen Explorer integrates Apache Spark via Spark JobServer. This provides the benefit that MashZone NextGen Explorer and Spark do not necessarily have to be deployed on the same machine, i.e., you can install MashZone NextGen Explorer on server A and Spark JobServer on server B (whereby B could be a more capable machine than A).

### Procedure

1. Copy the folder <MashZone NextGen Explorer installation>/va-sjs to the target machine where you want to run Spark JobServer.
2. Start Spark JobServer on the target machine via  
running va-sjs\bin\va-sjs.bat (Windows)  
running va-sjs/bin/va-sjs (Linux / Unix-derivates)
3. On the MashZone NextGen Explorer machine, open the config.json configuration file in a text editor.
4. Add the following code lines and save your changes.

```
"spark.enabled" : true,
"spark.jobserver.url" : "https://<target machine IP>:8090",
```

Whereby <target machine IP> is the IPv4-address of the machine running Spark JobServer, e.g., "spark.jobserver.url" : "https://172.16.0.1:8090", if the server has the IP 172.16.0.1.

Your settings are applied.

## Importing larger data sources

To deploy data files larger than 1GB, if Spark is deployed on a separate machine, you need to increase the relevant parameters.

Edit the following parameters in the Spark JobServer config file `va.conf`.

The file is located in the `<MashZone NextGen Explorer installation>/va-sjs/config/` directory.

- Set `request-chunk-aggregation-limit = 2000m` in section `spray.can/server`.
- Set `max-chunk-size = 2000m` in section `spray.can/server/parsing`.
- Set `short-timeout = 10 s` in section `spark/jobserver`.

Edit the following parameters in the Spark JobServer startup scripts `va-sjs.bat` for MS Windows and `va-sjs` for Linux.

The files are located in the `<MashZone NextGen Explorer installation>/va-sjs/bin/` directory.

- In `va-sjs.bat`: Change line `set DEFAULT_JVM_OPTS=""` to `set DEFAULT_JVM_OPTS="-Xmx8192m"`.
- In `va-sjs`: Change line `DEFAULT_JVM_OPTS=""` to `DEFAULT_JVM_OPTS="-Xmx8192m"`.

# 4 Set SSL configuration

---

- Enabling TLS ..... 18
- Setup of certificates and key material ..... 18

MashZone NextGen Explorer enables you to setup a secure communication connection between MashZone NextGen Explorer Client and MashZone NextGen Explorer Server.

In most cases, MashZone NextGen Explorer Server will not run on the same machine as MashZone NextGen Explorer Client. To secure the data transfer between client and server, the communication between the client and server should be encrypted.

The default setup after installation of MashZone NextGen Explorer will run the MashZone NextGen Explorer Server without support for encrypted communication (HTTPS / TLS / SSL), i.e., with Transport Layer Security (TLS) disabled. Thus, you cannot connect via https to the server.

## Enabling TLS

MashZone NextGen Explorer allows to enable TLS via settings in the config.json configuration file.

The config.json configuration file is located in the following folder:

*<MashZone NextGen Explorer installation> /config.*

### Procedure

1. Open the config.json configuration file in a text editor.
2. Enter the following code line and save your changes.

```
"httpServer.tls.enabled" : true,
```

TLS is enabled.

**Note:** MashZone NextGen Explorer ships with a self-signed certificate. Thus, when you enable TLS without replacing the key material by your own material ["Setup of certificates and key material" on page 18](#), Web browsers will issue a security warning when connecting to MashZone NextGen Explorer Server.

## Setup of certificates and key material

MashZone NextGen Explorer supports three different kinds of key material / certificates:

- JKS (Java KeyStore)
- PEM (Privacy Enhanced Mail)
- PKCS#12 (from RSA, Inc., based on Microsoft's PFX)

## Enabling TLS with JKS-based key material

MashZone NextGen Explorer allows to use public/private key material stored in a Java Key-Store (JKS) to be used as certificate for TLS communication.

### Procedure

1. Open the config.json configuration file in a text editor.
2. Enter the following code lines and save your changes.

```
"httpServer.tls.certType" : "JKS",  
"httpServer.tls.keyPath" : "<key path>",  
"httpServer.tls.keyPassword" : "<key password>",
```

- The **keyPath** parameter must point to the key file (usually ends with **.jks**). The path may either be relative (to MashZone NextGen Explorer' root folder) or absolute.
- The **keyPassword** parameter is the password required to open the key store.

TLS with JKS is enabled.

## Enabling TLS with PEM-based key material

### Procedure

1. Open the config.json configuration file in a text editor.
2. Enter the following code lines and save your changes.

```
"httpServer.tls.certType" : "PEM",  
"httpServer.tls.keyPath" : "<key path>",  
"httpServer.tls.certPath" : "<key password>",
```

- The **keyPath** parameter must point to the key file (usually ends with **.pem**). The path may either be relative (to MashZone NextGen Explorer' root folder) or absolute.
- The **certPath** parameter must point to the public key file (usually ends with **.pem**). The path may either be relative (to MashZone NextGen Explorer' root folder) or absolute.

TLS with PEM is enabled.

## Enabling TLS with PKCS#2-based key material

### Procedure

1. Open the config.json configuration file in a text editor.
2. Enter the following code lines and save your changes.

```
"httpServer.tls.certType" : "PKCS12",  
"httpServer.tls.keyPath" : "<key path>",
```

```
"httpServer.tls.keyPassword" : "<key password>",
```

- The **keyPath** parameter must point to the key file (usually ends with **.p12** ). The path may either be relative (to MashZone NextGen Explorer' root folder) or absolute.
- The **keyPassword** parameter is the password required to open the **PKCS#12** file.

## 5 Configure SSO with MashZone NextGen

You can configure Single Sign-On (SSO) between MashZone NextGen Explorer and MashZone NextGen.

By default, SSO is automatically set up if MashZone NextGen Explorer and MashZone NextGen are installed on the same machine. Where MashZone NextGen acts as Identity Provider (IdP) for MashZone NextGen Explorer (Service Provider). If MashZone NextGen Explorer and MashZone NextGen are installed on different machines, you need to configure SSO manually.

The relevant config.json file is located in the following directory.

<MashZone NextGen installation> \VisualAnalytics\config\

### Procedure

1. Open the config.json file in a text editor.
2. Change the setting of the *mashzone.hostname* parameter to `http://<my-mashzone-hostname>:<my-mashzone-port>`

Default is `http://localhost:8080`.

3. Save your settings.

Your settings are applied.



## 6 Configure startup behaviour

You can configure the startup behaviour of MashZone NextGen Explorer.

MashZone NextGen Explorer version 10.1 is delivered with an embedded Apache Kafka instance additionally to the embedded Spark instance and the startup and shutdown processes have been changed.

The start script supports the following options.

Options	Description
embeddedSpark	<p>Enables the embedded Apache Spark (Spark Job Server) instance. When enabled, this instance is automatically started when starting MashZone NextGen Explorer and is available via <code>https://localhost:19002</code>.</p> <p>The configuration can be found at <code>&lt;MashZone NextGen installation&gt;/VisualAnalytics/va-sjs/config</code>.</p> <p>When using another Spark Job Server instance, configure MashZone NextGen Explorer accordingly (<code>"spark.enabled"=true</code> and <code>"spark.jobserver.url"="&lt;URL of the running Spark Job Server&gt;"</code>) and start it without the <b>--embeddedSpark</b> option.</p>
disableEmbeddedKafka	<p>Disables the embedded Apache Kafka instance. When not disabled, this instance (and also the required Apache Zookeeper instance) is automatically started when starting MashZone NextGen Explorer and is available via <code>localhost:19004</code>.</p> <p>The configuration can be found at <code>&lt;MashZone NextGen installation&gt;/VisualAnalytics/va-kafka/config</code>.</p> <p>When using another Apache Kafka instance, configure MashZone NextGen Explorer accordingly (<code>"zookeeper.urls"="&lt;comma separated list of Zookeeper URLs&gt;"</code> and <code>"kafka.bootstrap.servers"="&lt;comma separated list of Kafka URLs&gt;"</code>) and start it with the <b>--disableEmbeddedKafka</b> option.</p>

The MashZone NextGen Explorer startup process is as follows:

- If explicitly enabled via the **--embeddedSpark** option
  - Apache Spark (Spark Job Server) is started on `https://localhost:19002` by calling the start script `<MashZone NextGen installation>/VisualAnalytics/va-sjs/bin/va-sjs[.bat]`
- If not explicitly disabled via the **--disableEmbeddedKafka** option
  - Apache Zookeeper is started on **localhost:19003** by calling the start script `<MashZone NextGen installation>/VisualAnalytics/va-kafka/bin/va-zookeeper[.bat]`
  - Apache Kafka is started on **localhost:19004** by calling the start script `<MashZone NextGen installation>/VisualAnalytics/va-kafka/bin/va-kafka[.bat]`
- MashZone NextGen Explorer is started

**Note:** Starting MashZone NextGen Explorer issue on Mac OS X 10.12 (Sierra)

There is a Java issue with respect to DNS lookup on Mac OS X 10.12 (Sierra) which causes the DNS lookup to block for 5 seconds (see <https://stackoverflow.com/questions/39636792/jvm-takes-a-long-time-to-resolve-ip-address-for-localhost/39698914#39698914>). As result thereof the MashZone NextGen Explorer startup process does not work correctly.

There are two workarounds for Mac OS X 10.12 (Sierra):

- Modify `/etc/hosts` by adding your Mac hostname to the entries for `127.0.0.1` and `::1`,
- or
- Increase the timeouts in the script accordingly.

The MashZone NextGen Explorer shutdown process is as follows:

- MashZone NextGen Explorer is shut down
- If Apache Spark (Spark Job Server) was started using the **--embeddedSpark** option
  - Apache Spark (Spark Job Server) is shut down by calling the shutdown script `<MashZone NextGen installation>/VisualAnalytics/va-sjs/bin/shutdown[.bat]`
- If Apache Kafka was started using the **--disableEmbeddedKafka** option
  - Apache Kafka is shut down by calling the shutdown script `<MashZone NextGen installation>/VisualAnalytics/va-kafka/bin/va-kafka-shutdown[.bat|.sh]`
  - Apache Zookeeper is shut down by calling the shutdown script `<MashZone NextGen installation>/VisualAnalytics/va-kafka/bin/va-zookeeper-shutdown[.bat|.sh]`

**Note:** When shutting down MashZone NextGen Explorer by pressing **Ctrl+C** on Windows, do not terminate the batch job since this will break the shutdown process otherwise.



## 7 Add custom JAR files

You can add custom JAR files to MashZone NextGen Explorer.

When working with cache-based data sources it is often necessary to add a custom JAR file to the class path of a MashZone NextGen Explorer instance in order to load custom classes when instances of these classes are used as key or value in a BigMemory cache or to load custom attribute extractors.

**Note:** It is not possible to replace classes delivered by MashZone NextGen Explorer via custom JAR files.

### Procedure

1. Copy the custom JAR files and all required libraries in the following folder.  
`<MashZone NextGen installation>/VisualAnalytics/lib/custom`
2. Start or restart MashZone NextGen Explorer.

The cache-based data sources using the custom classes can be deployed to and analyzed with MashZone NextGen Explorer.



---

# 8 Configure DES data source

---

■ Configuration of the DesEventBroker microservice .....	30
■ DesEventBroker microservice API .....	31
■ Configuration of the real-time data source .....	39

You can use Digital Event Services (DES) as data sources for MashZone NextGen Explorer. DES data sources can be made available to MashZone NextGen Explorer by a two-stage process.

1. A microservice (DesEventBroker) usually hosted by the MashZone NextGen Explorer instance creates a DES instance, subscribes for events of a configured type and writes them to a topic hosted by a Kafka instance.
2. A JSON file describing the real-time data source is deployed to the MashZone NextGen Explorer instance.

## Configuration of the DesEventBroker microservice

The first stage of deploying a streaming data source to MashZone NextGen Explorer is to configure the DesEventBroker such that it subscribes for the desired event type and writes them to a topic hosted by a Kafka instance.

The DesEventBroker microservice provides the following configuration properties:

Property name	Type	Description	Default value
com.softwareag.events.using encryption.configuration.location	String	The property for specifying the Digital Event Services encryption configuration location.  Mandatory: No	"../common/DigitalEventServices/security/passman"
com.softwareag.events.license.location	String	The property pointing to the location of the Digital Event Services license file.  Mandatory: No	"../common/DigitalEventServices/license/license.xml"
com.softwareag.events.repository.location	String	The property pointing to the common root of the annotation repository and the type repository.  Mandatory: No	"../common/DigitalEventServices/TypeRepository"
com.softwareag.events.routing.configuration.directory	String	The property for specifying the Digital Event Services configuration directory.  Mandatory: No	"../profiles/VAN/configuration"

The default values of the configuration properties are sufficient for an installation of DES and MashZone NextGen Explorer in the same SoftwareAG directory. In this case, no additional configuration is required and the DesEventBroker can be activated by using the following module specification in *<MashZone NextGen installation>/VisualAnalytics/config/modules.json*.

```
{
  "level"      : 90,
  "impl"       : "java:com.softwareag.va.des.impl.DesEventBrokerImpl"
},
```

Due to the provided default values this module specification is equivalent to the following module specification.

```
{
  "level"      : 90,
  "impl"       : "java:com.softwareag.va.des.impl.DesEventBrokerImpl",
  "options"    : {
    "config"   : {
      "com.softwareag.events.repository.location"
        : "../../common/DigitalEventServices/TypeRepository",
      "com.softwareag.events.license.location"
        : "../../common/DigitalEventServices/license/license.xml",
      "com.softwareag.events.encryption.configuration.location"
        : "../../common/DigitalEventServices/security/passman",
      "com.softwareag.events.routing.configuration.directory"
        : "../../profiles/VAN/configuration"
    }
  }
},
```

If DES and MashZone NextGen Explorer are not installed in the same SoftwareAG directory adapt the module specification accordingly.

## DesEventBroker microservice API

The DesEventBroker microservice provides subscribe and unsubscribe methods for subscribing for events of a configured event type and append them to a topic hosted by a Kafka instance and for deleting the subscription again.

### Subscribe method

The subscribe method takes a SubscriptionRequest specifying the desired subscription and returns an according SubscriptionResponse.

#### SubscriptionRequest

A SubscriptionRequest is a JSON object having the following properties.

Property Name	Type	Description	Default Value
<b>SubscriptionRequest</b>			
kafkaConfiguration	JSON Object	The property for configuring the Kafka topic the received events should be appended to. Mandatory: No	see KafkaConfiguration
transformer Configuration	JSON Object	The property for configuring the transformation of received events to records that can be written to Kafka. Mandatory: No	see TransformerConfiguration
universal MessagingUrl	String	The property for specifying the URL of the Universal Messaging instance used for sending events. Mandatory: No	"nsp://localhost:9000"
eventName	String	The property for specifying the event type that should be subscribed. Mandatory: Yes	
<b>KafkaConfiguration</b>			
bootstrapServers	String	For detailed information please refer the <a href="#">Apache Kafka documentation</a> . Mandatory: No	"localhost:19004"
acks	String		"1"
retries	Integer		0

Property Name	Type	Description	Default Value
batchSize	Integer		16 * 1024
lingerMs	Integer		0
bufferMemory	Integer		32 * 1024 * 1024
keySerializerClass Name	String		"org.apache.kafka. common.serialization. StringSerializer"
valueSerializerClass Name	String		"org.apache.kafka. common.serialization. ByteArraySerializer"
<b>TransformerConfiguration</b>			
topic	String	<p>Specifies the topic a received and transformed event should be appended to.</p> <p>The configuration property <b>topic</b> specifies the name of the Kafka topic.</p> <p>The configuration property <b>topicFactoryClassName</b> specifies a class that maps events to topic names and implements the <b>com.softwareag.va.des.factories.EventFunction</b> interface.</p> <p>The configuration of both <b>topic</b> and <b>topicFactoryClassName</b> is not allowed.</p> <p>If neither <b>topic</b> nor <b>topicFactoryClassName</b> is configured a topic with the configured <b>eventTyeName</b> is used</p>	
topicFactoryClass Name	String		

Property Name	Type	Description	Default Value
		for appending received and transformed events. Mandatory: No	

Property Name	Type	Description	Default Value
partition	Integer	Specifies the partition to which a received and transformed event should be sent.  The configuration property <b>partition</b> specifies the partition.  The configuration property <b>partitionFactoryClassName</b> specifies a class that maps events to partitions and implements the <b>com.softwareag.va.des.factories.EventFunction</b> interface.  The configuration of both <b>partition</b> and <b>partitionFactoryClassName</b> is not allowed.  If neither <b>partition</b> nor <b>partitionFactoryClassName</b> is configured but a key is present a partition will be chosen using a hash of the key. If neither key nor partition is present a partition will be assigned in a round-robin fashion.  Mandatory: No	
partitionFactoryClassName	String		

Property Name	Type	Description	Default Value
timestampField	String	Specifies the timestamp of a received and transformed event.	
timestampFactory ClassName	String	<p>The configuration property <b>timestampField</b> specifies the name of the event type's field holding the timestamp information. The field must be of type <b>TIME</b>.</p> <p>The configuration property <b>timestampFactoryClassName</b> specifies a class that maps events to Java epoch timestamps and implements the <b>com.softwareag.va.des.factories.EventFunction</b> interface.</p> <p>The configuration of both <b>timestampField</b> and <b>timestampFactoryClassName</b> is not allowed.</p> <p>If neither <b>timestampField</b> nor <b>timestampFactoryClassName</b> is configured, the received and transformed events are stamped with current system time.</p> <p>Mandatory: No</p>	
keyField	String	Specifies the key of a received and transformed event.	
keyFactory ClassName	String	The configuration property <b>keyField</b> specifies the name of	

Property Name	Type	Description	Default Value
		<p>the event type's field holding the key.</p> <p>The configuration property <b>keyFactoryClassName</b> specifies a class that maps events to keys and implements the <b>com.softwareag.va.des.factories.EventFunction</b> interface.</p> <p>The configuration of both <b>keyField</b> and <b>keyFactoryClassName</b> is not allowed.</p> <p>If neither <b>keyField</b> nor <b>keyFactoryClassName</b> is configured, the received and transformed events will have no key.</p> <p>Mandatory: No</p>	
valueFactoryClassName	String	<p>Specifies a class that maps events to values and implements the <b>com.softwareag.va.des.factories.EventFunction</b> interface.</p> <p>If not configured, the value of the received and transformed events will be the serialized event itself.</p> <p>Mandatory: No</p>	"com.softwareag.va.des.factories.SerializedBytesValueFactory"

The default values of SubscriptionRequest are sufficient if events are received via the default UniversalMessaging and written to embedded Kafka. In this case only the name of the event type that should be subscribed has to be specified.

```
{
  "eventName" : "<fully qualified name of the event type
                to be subscribed>"
}
```

Due to the provided default values this SubscriptionRequest is equivalent to the following SubscriptionRequest.

```
{
  "kafkaConfiguration"      : {
    "bootstrapServers"      : "localhost:19004",
    "acks"                   : "1",
    "retries"                : 0,
    "batchSize"              : 16384,
    "lingerMs"               : 0,
    "bufferMemory"           : 33554432,
    "keySerializerClassName" : "org.apache.kafka.common.serialization.StringSerializer",
    "valueSerializerClassName" : "org.apache.kafka.common.serialization.ByteArraySerializer"
  },
  "transformerConfiguration" : {
    "topic"                  : "<fully qualified name of the event type to be subscribed>",
    "valueFactoryClassName" : "com.softwareag.va.des.factories.SerializedBytesValueFactory"
  },
  "universalMessagingUrl"    : "nsp://localhost:9000",
  "eventName"                : "<fully qualified name of the event type to be subscribed>"
}
```

If the events are not received via the default UniversalMessaging or written to a Kafka instance not running on localhost:19004 or the transformation needs to be reconfigured adapt the SubscriptionRequest accordingly.

### SubscriptionResponse

A SubscriptionResponse is an empty JSON object.

## Unsubscribe method

The **unsubscribe** method takes an UnsubscriptionRequest specifying the subscription to be removed and returns an according UnsubscriptionResponse.

### UnsubscriptionRequest

An UnsubscriptionRequest is a JSON object having the following properties.

Property Name	Type	Description	Default Value
eventName	String	The property for specifying the event type that should be unsubscribed. Mandatory: Yes	

An UnsubscriptionRequest looks as follows.

```
{
```

```

"eventName" : "<fully qualified name of the event type
                to be unsubscribed>"
}

```

## UnsubscriptionResponse

An UnsubscriptionResponse is an empty JSON object.

## Usage

There are two ways to use the DesEventBroker microservice API.

- The MashZone NextGen Explorer Console contains a DES Event Broker entry that provides access to the DesEventBroker microservice API. The MashZone NextGen Explorer Console can be used for subscribing to and unsubscribing from event types at runtime.

**Note:** The DesEventBroker microservice has no persisted state, i.e., when restarting MashZone NextGen Explorer all subscriptions are gone.

- The DesEventBrokerConfigurationUtility microservice can be used for subscribing to event types at startup time of the MashZone NextGen Explorer server. The module specification of the DesEventBrokerConfigurationUtility microservice provides property subscriptions holding a JSON array of SubscriptionRequests. These SubscriptionRequests are sent to the DesEventBroker microservice at startup time of the MashZone NextGen Explorer server. The module specification looks as follows.

```

{
  "level"      : 91,
  "impl"      : "java:com.softwareag.va.des
                .DesEventBrokerConfigurationUtility",
  "options"   : {
    "config"  : {
      "subscriptions" : [
        {
          "eventName"
            : "<fully qualified name of the event type
              to be subscribed>"
        },
        {
          "eventName"
            : "<fully qualified name of another event type
              to be subscribed>"
        }
      ]
    }
  }
},

```

The number of the SubscriptionRequests is not limited and they can be configured as described above.

**Note:** The level of the DesEventBrokerConfigurationUtility must be greater than the level of the DesEventBroker. This guarantees that the DesEventBroker

microservice is started before the DesEventBrokerConfigurationUtility microservice.

## Configuration of the real-time data source

After subscribing for events of configured event types corresponding real-time data sources can be deployed to MashZone NextGen Explorer. In order to achieve this goal JSON files describing the real-time data sources have to be placed in the *<MashZone NextGen installation>/VisualAnalytics/data/auto-deploy/stream* directory for each subscribed DES event type. The JSON file must contain a single JSON object which supports the following properties.

Property Name	Type	Description	Default Value
zookeeperInstances	String	A comma-separated list of Zookeeper servers, e.g., localhost:2181,localhost:2182. Mandatory: No	zookeeper.urls configured in MashZone NextGen Explorer server or "127.0.0.1:19003" if not configured
brokerInstances	String	For detailed information please refer the <a href="#">Apache Kafka documentation</a> . Mandatory: No	kafka.bootstrap.servers configured in MashZone NextGen Explorer server or "127.0.0.1:19004" if not configured
topicName	String	The property specifying the name of the topic used for polling events. Mandatory: No	messageType
repositoryRoot	String	The property pointing to the common root of the annotation repository and the type repository Mandatory: No	"../common/DigitalEventServices/TypeRepository"
messageType	String	The property for specifying the name	

Property Name	Type	Description	Default Value
		of the event type that should be analyzed. Mandatory: Yes	
windowSize	Long	The size of the temporal window used for aggregating in milliseconds. Mandatory: Yes	

The supported default values are sufficient if events are written to MashZone NextGen Explorer's embedded Kafka and DES and MashZone NextGen Explorer are installed in the same **SoftwareAG** directory. In this case only the name of the event type that should be analyzed and the size of the temporal window has to be specified.

```
{
  "messageType" : "<fully qualified name of the subscribed
                  event type>",
  "windowSize"  : 60000
}
```

Due to the provided default values this JSON object is equivalent to the following JSON object.

```
{
  "zookeeperInstances" : "127.0.0.1:19003",
  "brokerInstances"    : "127.0.0.1:19004",
  "topicName"          : "<fully qualified name of the subscribed
                          event type>",
  "repositoryRoot"     : "../..../common/DigitalEventServices/TypeRepository",
  "messageType"        : "<fully qualified name of the subscribed
                          event type>",
  "windowSize"         : 60000
}
```

If the events are not written to a Kafka instance running on localhost:19004 or DES and MashZone NextGen Explorer are not installed in the same **SoftwareAG** directory adapt the JSON object accordingly.