

tts performance suite

Creator 2.101.1 Installation Manual - tts performance suite 2024r2



.....

Contents

- 1 General Information..... 4**
 - 1.1 Introduction..... 4
 - 1.1.1 Objective..... 4
 - 1.1.2 Target audience..... 4
 - 1.1.3 Prerequisites..... 4
 - 1.1.4 Contact..... 4
 - 1.1.5 Structure..... 4
 - 1.1.6 Conventions..... 5
 - 1.1.7 Your feedback is highly welcome..... 5
 - 1.2 Application description..... 5
 - 1.3 System overview and deployment scenarios..... 5

- 2 System requirements 6**

- 3 Installation step-by-step 6**
 - 3.1 Checklist..... 6
 - 3.2 Installation environment..... 6
 - 3.2.1 MinIO installation..... 6
 - 3.2.2 Connecting the tts performance suite and the Creator..... 6
 - 3.2.3 Configuring the Creator..... 7
 - 3.2.4 Creator permission to create a S3 lifecycle rule..... 9
 - 3.2.5 Configuring the steplist template..... 9
 - 3.2.6 How to make creator.application.properties accessible?..... 9
 - 3.2.7 Templates Customizing..... 10
 - 3.3 Application server settings..... 10
 - 3.3.1 JVM settings..... 10
 - 3.3.2 Deno settings..... 10
 - 3.4 Deployment..... 11
 - 3.4.1 Tomcat 10..... 11
 - 3.5 Post installation steps..... 13
 - 3.5.1 Test the creator..... 13

- 4 Configuring the First View Note (since Version 2023) 14**
 - 4.1 Configure the Fist Visit Note..... 14
 - 4.1.1 Settings..... 14

- 5 Proxy troubleshooting..... 16**



5.1	Cookies	16
6	Appendix	16
6.1	creator.application.properties reference	16
6.2	Template options reference	18
6.2.1	Example	18
6.3	Cookies	18

1 General Information

1.1 Introduction

1.1.1 Objective

This document describes the requirements of the tts performance suite component named **Creator**, and its installation process.

The goal of this document is to provide an understanding of the server configuration and to guarantee its successful installation. For installation instructions of other tts performance suite components, Curator and WebAccess, refer to the corresponding manuals.

1.1.2 Target audience

System administrators, developers and all interested parties.

1.1.3 Prerequisites

It is expected that system administrators, developers and who else may be interested have good knowledge of

- Windows and/or Unix-based operating systems
- Administration and handling of application server Apache Tomcat
- Deployment of web applications (WAR, external web application)

 **Please read this installation guide completely and carefully!**

1.1.4 Contact

- tts Support
Phone: +49 (0) 2 21 / 17 09 30 - 110
Fax: +49 (0) 2 21 / 17 09 30 - 170
support@tt-s.com
- Application consultant
Application consultants are very experienced with the tts performance suite and can guide you through the installation process, providing professional solutions to match your requirements. If no application consultant has yet been assigned, please contact your key account manager.

1.1.5 Structure

The first chapter sheds light on the functionalities and technologies used in the Creator and finished by presenting the System overview and deployments scenarios.

The next section points to the System requirements concerning hardware and software. They should be checked thoroughly and carefully.

The Installation chapter guides you step-by-step through the deployment of the Creator, including defining server configuration. Post installation steps deals with initial administration tasks.


Due to the fact that the Creator is a complex application, many configuration options are provided. A variety of settings are explained in detail in the Appendix. Each property or parameter of the application's services is listed with its name, a description and the possible values.

1.1.6 Conventions

1.1.6.1 Symbols

To highlight important information on the one hand, and “nice to know” details on the other, the following icons are used:

 Attention

 Hint or note

 Tip

1.1.6.2 Variables

 Variables are marked with a leading \$

Variable	Description
\$TTPS_HOME	installation directory of tts performance suite
\$TOMCAT_HOME	installation directory of Apache Tomcat

1.1.7 Your feedback is highly welcome

tts welcomes your feedback concerning the quality and usefulness of this manual. Your comments and suggestions will be considered as valuable input for future revisions of this manual.

- Found an error? Please let us know where.
- A topic is not described clearly enough? Please let us know which one.
- Need more information? On which topic?
- Something doesn’t work for you? Please let us know, so we can provide additional examples.

Please feel free to send us your feedback: support@tt-s.com. We appreciate your help!

1.2 Application description

The Creator consists of a backend component, written in Java (Spring Boot) and a frontend created with Typescript. The backend is responsible for the connection to the curator and the repository (S3/MinIO), the frontend presents the Creator editor to the user.

1.3 System overview and deployment scenarios

The Creator is a standalone application, which is deployed as a separate war file to an application server. Nevertheless, you’ll need the tts performance suite installed and accessible to use it.

There are two different deployment scenarios.

The first scenario, which is the minimum way in practice, involves one application server hosting the tts performance suite (Curator and WebAccess) and also the Creator.

The second, recommended, scenario would be to deploy the Creator to its own application server. It differs from the first scenario by being more scalable and dynamic, thanks to using separate application servers plus a load balancer that distributes all requests among the reverse proxies.

Both scenarios, minimum and recommended, are field-tested concepts. The usage depends on the customer's requirements. Nevertheless, we recommend the deployment including a reverse proxy in any case. Moreover, if there is a huge number of end users expected, an installation in a clustered environment is advised, since this scenario offers more flexibility, better scalability, and higher performance.

2 System requirements



For the current hardware and software requirements of tts server, please refer to the system requirements.

3 Installation step-by-step

3.1 Checklist

Before you start deploying the Creator, please ensure that the following requirements are met:

- The installation files are available
- The application server is installed properly
- The MinIO Server is installed and configured
- The tts performance suite is installed

3.2 Installation environment

3.2.1 MinIO installation

For full MinIO installation instructions refer to the tts performance suite installation manual. It is required that the Creator use the same MinIO instance as the Curator.

Create a bucket named "bg-creator", which is the default name. To use another bucket name use the corresponding value within the creator.application.properties.

The properties are explained in detail in the appendix.

If you want to encrypt the traffic between MinIO and Creator using TLS with a self-signed root certificate, you need to add your root certificate to your system's trusted root certificates, and set the environment variable `DENO_TLS_CA_STORE` to `system` on your system. Otherwise, the deno processes that are spawned by the Creator will not be able to verify the self-signed certificate used by MinIO.

3.2.2 Connecting the tts performance suite and the Creator

These parameters need to be added to the application-config.properties of the tts performance suite to connect to the Creator:

3.2.2.1 2022

```
1 ttkf.server.businessguidancecreator.endpoint=https://<creator-domain>:<port>/  
↪ creator/v1/session/open
```

3.2.2.2 2022r2

```
1 ttkf.server.businessguidancecreator.endpoint=https://<creator-domain>:<port>/
↪ creator
```

3.2.2.3 2023 and later

```
1 ttkf.server.businessguidancecreator.endpoint=https://<creator-domain>:<port>/
↪ creator
2 ttkf.server.businessguidancecreator.presignedurl.publickey.endpoint=https://<
↪ creator-domain>:<port>/creator/v1/presigned-url/certificate
```

3.2.3 Configuring the Creator

The actual configuration is carried out via a file named *creator.application.properties*. Place it directly in \$TTPS_HOME.

The following example shows a snippet of a *creator.application.properties* file for the minimum deployment scenario with the creator placed on the same application server as the tts performance suite.

For all available parameters check the *creator.application.properties* reference in the appendix.

3.2.3.1 2023r2 and later

```
1 # optional, add customizing specific folder.
2 # The customizing folder has the following structure:
3 #   └─ customizing-root-dir
4 #       └─ template-name
5 #           └─ configuration-id
6 #
7 # e.g:
8 #   └─ customizing-root-dir
9 #       └─ steplist
10 #           └─ eeefffaaa
11 # The configuration id is equals the client configuration id from the producer/curator.
12 creator.templates.customizing.location=file:/<path-to-templates-customizing-directory>
```

3.2.3.2 2023 and later

```
1 # Internal URL of the Curator
2 creator.curator.internal-address=https://<curator-domain>:<port>/curator
3 # External URL of the Curator
4 creator.curator.external-address=https://<curator-domain>:<port>/curator
5 # Version of the Curator where creator.curator.internal-address points to
6 creator.curator.version=<curator-version>
7 # Internal address of minio server.
8 # Must be exactly the same value as the
9 #   ttkf.server.repository.endpoint.internal
10 # parameter in the Curator configuration.
11 # E.g.
```

```

12 # if
13 #   ttkf.server.repository.endpoint.internal=http://127.0.0.1:9000
14 # is used in the Curator, use
15 #   creator.s3.minio.internal-address= http://127.0.0.1:9000
16 # if
17 #   ttkf.server.repository.endpoint.internal=http://localhost:9000
18 # is used in the Curator, use
19 #   creator.s3.minio.internal-address= http://localhost:9000
20 creator.s3.minio.internal-address=https://<minio-domain>:9000
21 # External address of minio server
22 creator.s3.minio.external-address=https://<minio-domain>:9000
23 # MionIO authentication
24 creator.s3.access-key={MinIO credentials}
25 creator.s3.secret-key={MinIO credentials}
26 # Path to the templates directory
27 creator.templates.location=file://<path-to-templates-directory>

```

3.2.3.3 2022r2 and before

Contains old mandatory parameter `creator.jwt-max-age`, `creator.keep-alive-interval` and `creator.keep-alive-retry-delay`

```

1 # Internal URL of the Curator
2 creator.curator.internal-address=https://<curator-domain>:<port>/curator
3 # External URL of the Curator
4 creator.curator.external-address=https://<curator-domain>:<port>/curator
5 # Version of the Curator where creator.curator.internal-address points to
6 creator.curator.version=<curator-version>
7 # Seconds until the jwt expires. Must be 0 < jwt-max-age <= Curator JSESSIONID Max
↳ -Age.
8 creator.jwt-max-age=300
9 # Send a keep-alive request every keep-alive-interval seconds.
10 creator.keep-alive-interval=150
11 # Time in seconds to wait before retrying a failed keep-alive
12 creator.keep-alive-retry-delay=10
13 # Internal address of minio server.
14 # Must be exactly the same value as the
15 #   ttkf.server.repository.endpoint.internal
16 # parameter in the Curator configuration.
17 # E.g.
18 # if
19 #   ttkf.server.repository.endpoint.internal=http://127.0.0.1:9000
20 # is used in the Curator, use
21 #   creator.s3.minio.internal-address= http://127.0.0.1:9000
22 # if
23 #   ttkf.server.repository.endpoint.internal=http://localhost:9000
24 # is used in the Curator, use
25 #   creator.s3.minio.internal-address= http://localhost:9000
26 creator.s3.minio.internal-address=https://<minio-domain>:9000
27 # External address of minio server
28 creator.s3.minio.external-address=https://<minio-domain>:9000
29 # MionIO authentication
30 creator.s3.access-key={MinIO credentials}
31 creator.s3.secret-key={MinIO credentials}
32 # Path to the templates directory

```



```
33 creator.templates.location=file:///<path-to-templates-directory>
```

3.2.4 Creator permission to create a S3 lifecycle rule

The creator has a preview for documents.

To do this, files are stored on S3 `s3://<defined_bucket_in_application_config.properties>/<defined_prefix_in_application_config.properties>/previews/<preview_files>` that are no longer needed after the preview.

Therefore, a lifecycle rule is created that says that after one day, everything with the key prefix `<defined_prefix_in_application_config.properties>/previews/*` will be deleted.

If the creator does not have permissions or something else goes wrong when setting the lifecycle rule is set, it will appear as an info in the log and the creator will continue as normal. So it is optional to give the creator the permission. If he does not have it, some else is responsible to clean up the files himself.

3.2.4.1 Disable s3 lifecycle rule

If you want to disable the complete s3 lifecycle mechanism use following `application-config.properties`.

```
1 creator.preview.s3.lifecycle.skip=true
```

3.2.5 Configuring the steplist template

The steplist template is configured using the file referenced by `creator.templates.options` in `creator.application.properties`. The following shows a example configuration.

The settings are described in the *Appendix*.

```
1 steplist:
2   use-informal-auto-texts: true|false
3   recorder:
4     port: <recorderPort>
```

3.2.6 How to make creator.application.properties accessible?

There exist two options to provide the application with the configuration file externally. Either you extend the class path so the application will find that properties file; or you directly define its path within the context file as a parameter.

For better maintenance, we suggest to configure the path to the location of `creator.application.properties` directly in the corresponding Creator context file.

To do so, add the `spring.config.location` context parameter in the context file during deployment (see the deployment chapter):

```
1 ...
2 <Parameter name="spring.config.location" override="false" value="file:/C:/tts/
↪ creator.application.properties"/>
```

3 ...

3.2.7 Templates Customizing

Add following configuration parameter to the application server settings:

```
1 creator.templates.customizing.location=file:/<
↳ absolute_path_to_templates_customizing_directory>
```

The customizing folder has the following structure:

```
1 customizing-root-dir
2   └─ template-name
3   └─ configuration-id
4
5 e.g:
6 customizing-root-dir
7   └─ steplist
8   └─ eeefffaaa
```

The configuration id is equals the client configuration id from the producer/curator. In this folder, customizing files could be can be deposited.

Which files are applied/supported is template specific. Please look into the template specific manual. As template developer see the "Template Developer Guide".

3.3 Application server settings

3.3.1 JVM settings

The JVM of each server instance must meet at least the following memory settings and might be adapted to the environment's requirements:

- Maximum Java heap size: 1024 MB (-Xmx1024m)
- Initial Java heap size: 256 MB (-Xms256m)
- Maximum thread stack size: (-Xss256k)

Depending on the Java distribution UTF-8 might not be the default encoding. Thus, the following JVM parameter has to be set:

```
1 -Dfile.encoding=utf-8
```

3.3.2 Deno settings

The template backend scripts will be executed in a javascript runtime called deno. The creator provides a bundled deno version for following platform and arch.

platform	arch
windows	amd64 (x86)
linux	amd64 (x86)

platform	arch
linux	aarch64

If another deno executable is required, it can be set via following application config property: `creator.deno.executable`. This could be the case if a creator update is not possible, but the bundled deno version has a security issue.

3.3.2.1 Deno performance settings

If the creator runs on low performance compute unit following application config properties can be adjusted to ensure better operation.

`creator.deno.timeout` Maximum number of seconds a single deno process is allowed to run. Increase this, if backend scripts runs regularly in some timeouts. On low performance machines, the execution could take some more time.

`creator.deno.v8.max-heap-size` Maximum amount of memory (in Mbytes) used by a single deno process. Reduce this, if the server runs in out of memory issues. On low performance machines, the default settings could consume too much memory.

3.4 Deployment

Usually, tts provides pre-configured .WAR files for the server components.

Starting with Creator, release 2022r2, there's a `templates.zip` which has to be deployed additionally to the server.

1. Stop Tomcat if it is running.
2. Unzip the `templates.zip` and copy its content to `$TOMCAT_HOME/`
3. Copy `creator.war` to `$TOMCAT_HOME/webapps`.
4. Create the `creator.xml` file in the `$TOMCAT_HOME/conf/Catalina/localhost` directory.
5. Configure the location of the `creator.application.properties` within this context file.

Example:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3   <Context reloadable="false">
4
5     <Parameter name="spring.config.location" override="false" value="file:/C:/
↪ tts/creator.application.properties" />
6
7   </Context>

```

6. Restart Tomcat.

3.4.1 Tomcat 10

There is a significant breaking change between Tomcat 9.0.x and Tomcat 10.0.x. The Java package used by the specification APIs has changed from `javax...` to `jakarta...`

The Creator supports only the javax API.

Tomcat can convert an existing web application from Java EE 8 to Jakarta EE 9 at deployment time using the Apache Tomcat migration tool for Jakarta EE. To make use of the feature, the web application should be placed in the Host legacyAppBase folder (by default named webapps-javaee) and they will be converted to an equivalent Jakarta EE web application in the Host appBase folder (by default named webapps).

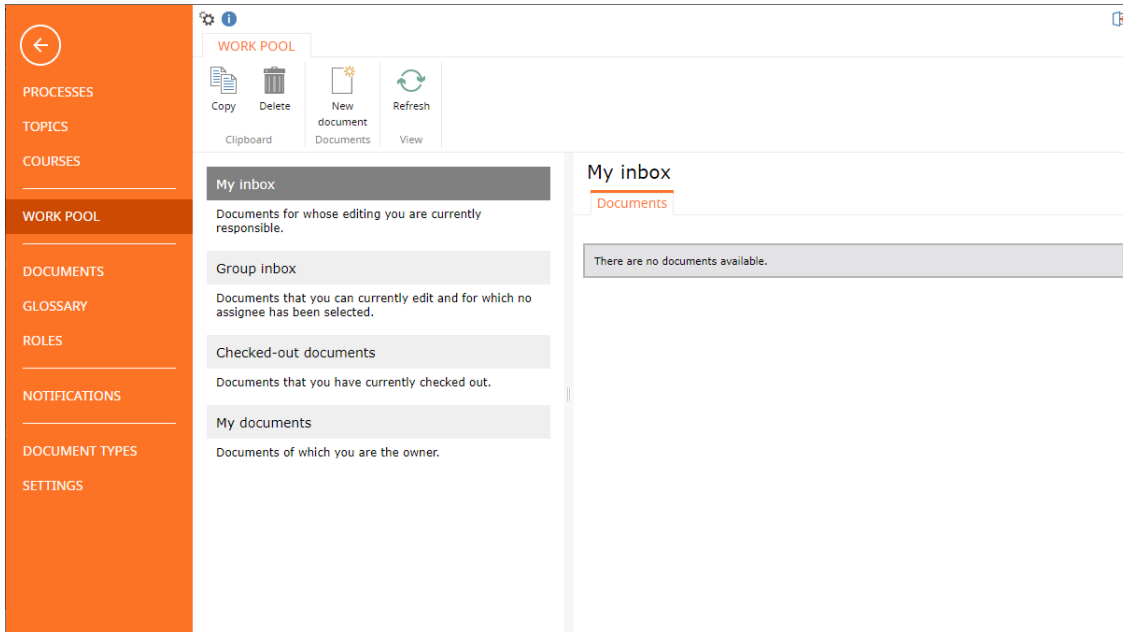
Example:

```
1 <Host name="localhost" appBase="webapps-javaee" unpackWARs="true" autoDeploy="
↳ true">
2   <Valve className="org.apache.tomcat.jakartaee.TomcatMigrationToolValve"
↳ jakartaEeForward="true" />
3 </Host>
```

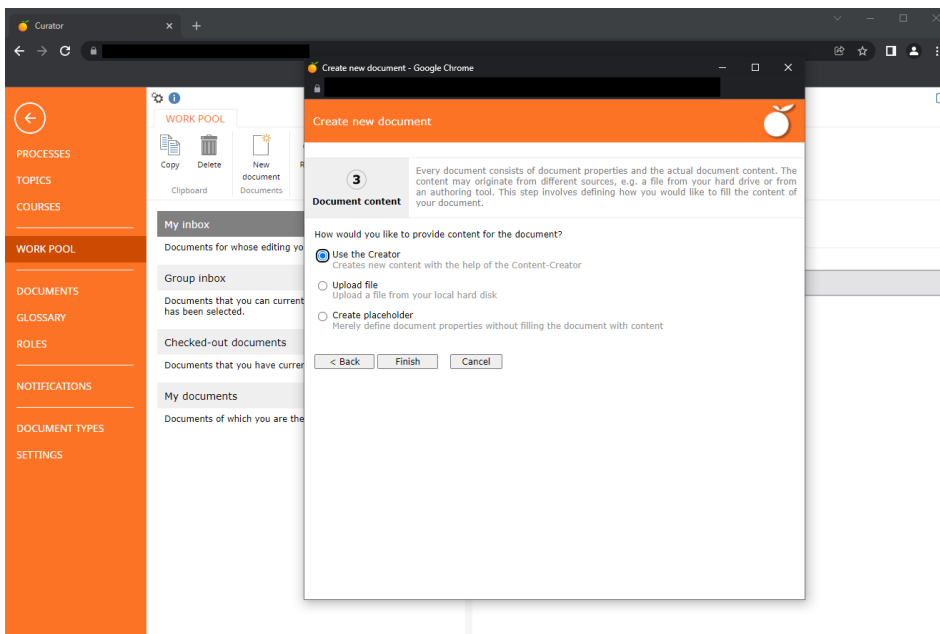
3.5 Post installation steps

3.5.1 Test the creator

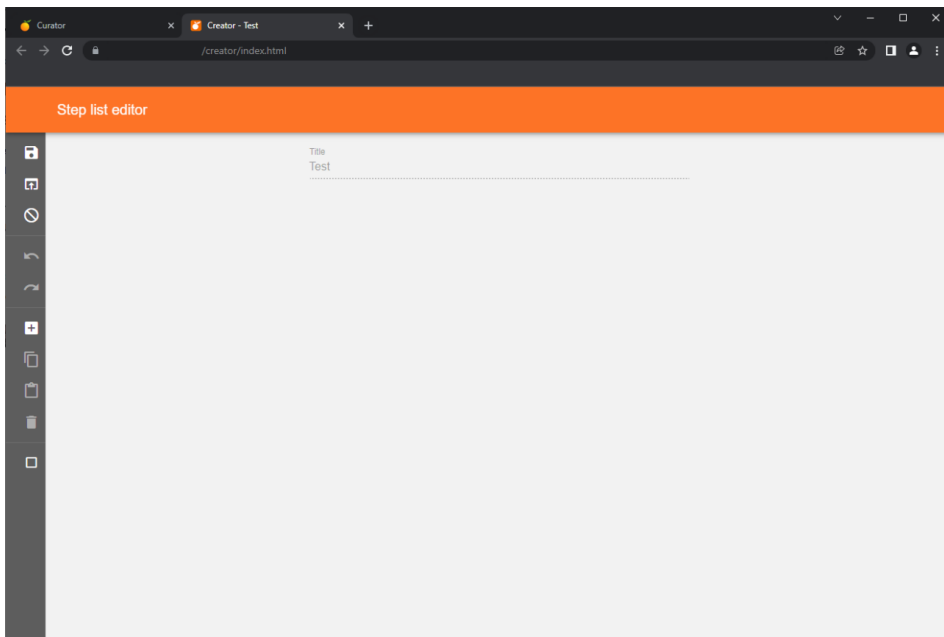
Log in to the Curator



When creating a new document, on step three of the wizard, there should be the option "Use the Creator"




After clicking "Finish" the Creator editor should open with a new document in a separate browser window.



4 Configuring the First View Note (since Version 2023)

Since Version 2023 a note can be configured to be shown the first time one visits the template selection.

 Important: Even if the feature is enabled and configured. If only one template is installed, no template selection appears and therefore the note is **not** displayed. The one template will be opened directly.

4.1 Configure the First Visit Note

To turn this feature on one has to configure it like follows.

1. Stop the creator
2. Create a Settings file in yml format. e.g. `first-visit-note-options.yml`
3. Open your `creator.application.properties` file and add `creator.first-visit-note.options=file:/path/to/file/From/step/1.yml`
4. Start the creator

4.1.1 Settings

General configuration for the note.

Option	required/optional	Description
<code>fallbackLanguage</code>	required	The message can be displayed in different languages. If a message is not found in a specific language, this language shall be used. References to a message in a specified language. Possible values are en, de, fr, es.

Option	required/optional	Description
showWhenOpenedFrom	required	The template selection can be opened both from the curator and from the QA. One can configure if the note will be shown when the template selection was opened from the QA, or the curator.
showAgainInDays	optional	If set this option defines when to show the note again after one has confirmed it. Will only expect positiv integer values such as e.g. 30 for having the note reappear after 30 days.
firstVisitNote	required	Array of elements defining the messages in different languages. See chapter <i>firstVisitNote</i> for details.

4.1.1.1 firstVisitNote

Option	required/optional	Description
en	optional	Message defined in English. See chapter <i>Message</i> .
de	optional	Message defined in German. See chapter <i>Message</i> .
fr	optional	Message defined in French. See chapter <i>Message</i> .
es	optional	Message defined in Spanish. See chapter <i>Message</i> .

4.1.1.2 Message

Option	required/optional	Description
message	required	Message to be displayed in the given language. One may add HTML elements to the message.
buttonText	required	Text for the confirmation Button in the given language

4.1.1.3 Sample configuration

Message is configured in english and in german. The fallback is english and the message will be repeatedly shown every 60 days. The message will only be shown if the template selection was opened from the QA and not if it was opened from the Curator

```

1 fallbackLanguage: "en"
2 showNoteWhenOpenedFrom:
3   - qa
4 showAgainInDays: 60
5 firstVisitNote:
6   en:
7     message: |
8       Good morning.<br />
9       We are <a href="https://tt-s.com" target="_blank">TTS</a>
10    buttonText: "Confirm"
11   de:

```

```

12     message: "Guten Morgen.<br>Wir sind <a href='https://tt-s.com' target='
↳ _blank'>TTS</a>"
13     buttonText: "Bestätigen"
    
```

5 Proxy troubleshooting

5.1 Cookies

The following cookies must not set to http only

```

1 jwtMaxAge
2 jwtKeepAliveInterval
3 jwtKeepAliveRetryDelay
    
```

6 Appendix

6.1 creator.application.properties reference

Name	Description	Default
creator.curator.external-address	External URL of the Curator.	-
creator.curator.internal-address	Internal URL of the Curator.	-
creator.curator.version	Version of the used Curator. For Example "22.1".	-
creator.deno.executable	Path to the deno executable. If not configured, the bundled deno executable will be used.	-
creator.deno.timeout	Maximum number of seconds a single deno process is allowed to run.	60
creator.deno.v8.max-heap-size	Maximum amount of memory (in MBytes) used by a single deno process.	128
creator.s3.access-key	Access key ID to access s3/minio bucket. If creator.s3.accessKey or creator.s3.secretKey is omitted DefaultCredentialProvider is used to find credentials.	-
creator.s3.arn-kms-key	If an encrypted bucket is used this needs to contain the ARN of the KMS key.	-

Name	Description	Default
creator.s3.arn-role	To grant users access to the s3 bucket, temporary credentials are being created using STS AssumeRole. This property must contain the ARN of the parent role that limits the permissions of the temporary credentials. The temporary credentials cannot assume broader permissions than the parent role. The parent role needs to allow s3:ListBucket access on the bucket, and s3:GetObject and s3:PutObject on the prefixed key. If an encrypted bucket is used, it needs to allow kms:Encrypt, kms:Decrypt and kms:GenerateDataKey with the KMS key ARN. arn-role can be omitted if minio is used.	
creator.s3.bucket-name	-	bg-creator
creator.s3.key-prefix	Path used by the creator inside the bucket. Must be empty or end with a '/'.	
creator.s3.minio	Minio options. Must be omitted if AWS S3 buckets are used.	-
creator.s3.minio.external-address	External address of minio server.	-
creator.s3.minio.internal-address	Internal address of minio server.	-
creator.s3.region	-	us-east-1
creator.s3.secret-key	Secret access key to access s3/minio bucket. If creator.s3.accessKey or creator.s3.secretKey is omitted DefaultCredentialProvider is used to find credentials.	-
creator.preview.s3.lifecycle.skip	Prevent that the creator tries to add a s3 lifecycle rule for the preview path rule	true
creator.templates.external-upload	Enable the external document Upload in the template selection if mode is 'create'. Only 'true' is true.	true
creator.templates.location	Path to the given templates. e.g. file:/	-
creator.templates.options	Path to the one and only templates options yaml file. e.g. file:/	-
creator.templates.customizingPublication	Path to the customizing directory. e.g. file:/	-
creator.web-log-level	Only log entries with a level equal or 'above' will be logged on the server. Must be debug, info, warn or error.	info
logging.config	Path to custom log4j2.xml configuration file. If not set a default log4j2.xml is used.	-
server.servlet.context-path	-	/creator
Deprecated since 2023 creator.jwt-max-age	Seconds until the jwt expires. Must be 0 < jwt-max-age <= Curator JSESSIONID Max-Age.	0
Deprecated since 2023 creator.keep-alive-interval	Send a keep-alive request every keep-alive-interval seconds. Must be 0 < keep-alive-interval < Curator JSESSIONID Max-Age.	0

Name	Description	Default
Deprecated since 2023 creator.keep-alive-retry-delay	Time in seconds to wait before retrying a failed keep-alive request. Must be 0 < keep-alive-retry-delay < Curator JSESSIONID Max-Age.	0

6.2 Template options reference

6.2.1 Example

```

1 steplist:
2   use-informal-auto-texts: true|false
3   recorder:
4     port: <recorderPort>
    
```

6.3 Cookies

Name	Description	HTTP-ONLY
jwt	Used for authentication	true
jwtMaxAge	Used to retrieve new jwts	false
jwtKeepAliveInterval	Used to retrieve new jwts	false
jwtKeepAliveRetryDelay	Used to retrieve new jwts	false