

# Tornado Installation and Configuration Guide

Version 2.11  
April 2026



**docmosis**



**Copyright**

© 2026 Docmosis Pty Ltd

**Trademarks**

Docmosis is a registered trademark of Docmosis Pty Ltd.

<https://www.docmosis.com>

Microsoft Word and MS Windows are registered trademarks of the Microsoft Corporation.

<http://office.microsoft.com/en-us/default.aspx>

<http://www.microsoft.com/windows/>

Adobe® PDF is a trademark of the Adobe Corporation.

<http://www.adobe.com/products/acrobat/adobepdf.html>

LibreOffice is a trademark of LibreOffice contributors and/or their affiliates.

<http://www.libreoffice.org>



# TABLE OF CONTENTS

---

- 1. INTRODUCTION.....6**
- 1.1. Using this Guide.....6
- 1.2. Troubleshooting.....6
- 1.3. Related Reading.....6
- 1.4. Support.....7
  
- 2. PLANNING THE TORNADO INSTALLATION.....8**
- 2.1. Hardware Requirements.....8
- 2.2. Software Requirements.....8
- 2.3. Template Source Location.....8
- 2.4. Supplied Software.....9
  - 2.4.1. Docker Installation.....9
  - 2.4.2. Manual Installation.....10
- 2.5. Securing the Tornado Environment.....10
  
- 3. INSTALLING AND STARTING TORNADO.....12**
- 3.1. Installing Tornado.....12
- 3.2. Starting Tornado.....12
  - 3.2.1. Running from the Scripts.....13
- 3.3. Configuring Tornado via the Web Console.....13
- 3.4. Using The Tornado Console.....16
  - 3.4.1. Overview of the Status Tab.....16
  - 3.4.2. Generating a Sample Document.....17



- 4. HOW TORNADO CONFIGURATION WORKS.....21**
  - 4.1. Where Configuration Settings Are Stored.....21
  - 4.2. Changing Configuration from the Java Command Line.....21
  - 4.3. Changing Configuration by Setting Environment Variables.....22
  - 4.4. Changing Configuration Settings on the Tornado Web Console.....22
  - 4.5. Getting Runtime Help for the Configuration Parameters.....23
  - 4.6. Clearing the Configuration.....23
  
- 5. GENERATING DOCUMENTS FROM THE CONSOLE.....24**
  - 5.1. Storing and Editing Templates.....24
  - 5.2. Storing Stock Images.....26
  - 5.3. Supplying Test Data.....26
  - 5.4. The Dev Mode option.....27
  - 5.5. Selecting Output Formats.....27
  
- 6. GENERATING DOCUMENTS FROM AN APPLICATION.....29**
  - 6.1. Invoking the Render Service.....29
  
- 7. MONITORING TORNADO.....30**
  
- 8. COMPARING TORNADO WITH DOCMOSIS CLOUD.....31**
  - 8.1. Service Requests in Tornado.....31
  - 8.2. Store-to Directives.....32
  - 8.3. REST Clients.....32
  - 8.4. Emailing Documents.....32



- 9. ALL CONFIGURATION OPTIONS.....33**
- 9.1. Common Settings.....33**
  - 9.1.1. Templates in a Local Directory.....34
  - 9.1.2. Templates in an AWS S3 Bucket.....34
  - 9.1.3. Templates in an Azure Blob Store.....35
  - 9.1.4. Templates in a Google Bucket.....36
- 9.2. Logging Settings.....37**
- 9.3. Enabling SSL/TLS Encryption.....37**
- 9.4. Other Network Settings.....38**
- 9.5. Enabling Email from Tornado.....39**



# 1. INTRODUCTION

---

Tornado is a stand-alone Document Generation engine that your application calls (using https) to generate documents. This guide provides details on installing, configuring, and starting Tornado ready to generate documents.

## 1.1. Using this Guide

This guide is a summary for software developers and DevOps teams responsible for installing and configuring Tornado.

## 1.2. Troubleshooting

After initial successful launch and configuration of Tornado, when it starts and during operation, it writes detailed log information to files in the configured Working Area. Any time issues occur with startup, configuration, or document generation, examine the logs for helpful information.

## 1.3. Related Reading

All the documentation relating to the latest version of Tornado is available on the Resources website.

<https://resources.docmosis.com/documentation/tornado/latest>

Docmosis document generation provides a large number of features, some of which are controlled by the templates, others controlled by the data.

To get the most out of Docmosis, please read the available Guides.

Tornado Template Guide	This guide provides a complete reference for template authors responsible for designing Docmosis templates.
Tornado Web Services Guide	This guide provides a complete reference for software developers responsible for connecting to the Tornado web service REST-based API.
Tornado – Quick Reference	This document provides a summary of all supported Docmosis Elements, Operators, Expressions, Functions,



	Ranges and Built-in Variables.
Tornado – Release Notes	The Release Notes contain a summary of each Tornado release including: New Features/Changes, web service API changes, new core engine, as well as bug fixes/technical changes.
Tornado Deployment Security Guide	This document discusses the relevant considerations for securing the Tornado server and interfaces.

## 1.4. Support

Please refer to the [Resources](#) site for FAQs, tutorials, code samples, and other articles. For additional support, please refer to the Docmosis [Support](#) page.



## 2. PLANNING THE TORNADO INSTALLATION

---

This chapter provides information about the main features of a Docmosis template.

### 2.1. Hardware Requirements

RAM: minimum 200Mb for Tornado plus 300Mb of RAM per converter

CPU: 1 Core plus 1 Core per converter

Disk Space: 100Mb plus allowance for templates and cached files as required

### 2.2. Software Requirements

Software requirements for the server machine are

- Java version 11 or later
- LibreOffice version 7 or later (download directly from [libreoffice.org](https://libreoffice.org) is recommended)
- Fonts as required

A license key is required to run Tornado. Please go to [Tornado Free Trial](#) to obtain a trial license key.

### 2.3. Template Source Location

Tornado can read templates (and stock images) from four types of location:

Source	Description
Directory	Templates are in a folder on the host running Tornado. This is simplest for testing, but less convenient in production environments. Templates and images are located at the path specified (no specific folder is required unlike the other options below).



Source	Description
AWS Bucket	Templates are located in an AWS Bucket. Tornado requires only read and list permission. Templates are in a “templates” folder and stock images (if any) are in an “images” folder.
Azure Blob Store	Templates are located in an Azure Blob Store. Tornado requires only read and list permission. Templates are in a “templates” folder and stock images (if any) are in an “images” folder.
Google Bucket	Templates are in a Google Bucket. Tornado requires only read and list permission. Templates are in a “templates” folder and stock images (if any) are in an “images” folder.

Using a local directory is simple for evaluation of Tornado and ideal in dev/test environments since it facilitates rapid modification and testing of templates.

See section 9.1 Common Settings for details on configuring the Template Source Location for the location of your choice.

## 2.4. Supplied Software

### 2.4.1. Docker Installation

For a Docker installation, please see our docker project, which includes an example Dockerfile:

<https://github.com/docmosis/tornado-docker>

Docker makes it very easy to get Tornado started without the detailed installation process.



## 2.4.2. Manual Installation

For a manual installation, the following items are included in the Docmosis Tornado bundle:

- **docs:** documentation folder containing all of the Tornado documentation
- **licenses:** license folder containing the Docmosis license agreement, as well as third party licenses
- **docmosisTornado[version number].war:** the installation package
- **placeholder.png:** placeholder image that can be used in templates
- **startTornado:** sample Windows batch file script
- **startTornado.sh:** sample shell script
- **readme.txt:** notes, and welcome message

Download the latest version of the Tornado software from the link below.

<https://resources.docmosis.com/software-downloads/tornado/the-software/latest>



Section 3, details how to install, configure, and run Tornado.

## 2.5. Securing the Tornado Environment

Tornado runs entirely on-premise, so the security and reliability of the service is the responsibility of the appropriate on-site teams. Tornado provides multiple features to assist with security and reliability:

- Configurable administrative password
- Configurable API access key
- https/SSL configuration
- API endpoints for monitoring



Combined with an appropriate environmental configuration, the above features allow the service to be secured as required. Please see the Tornado Deployment Security Guide for detailed consideration aspects of making the Tornado environment secure.



## 3. INSTALLING AND STARTING TORNADO

---

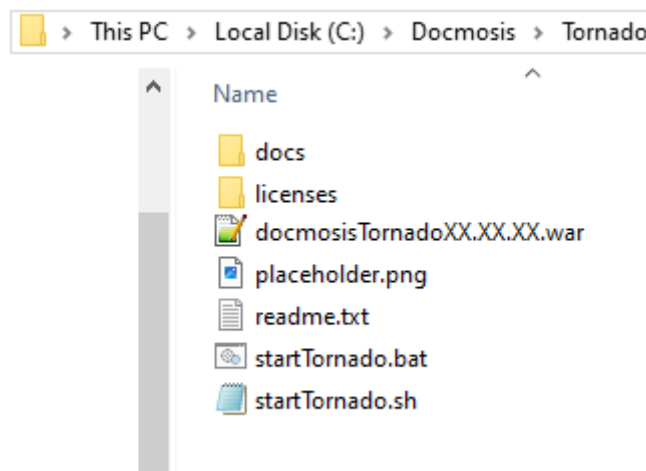
### 3.1. Installing Tornado

Tornado requires an area for program and working files. For the following example, this will be set to C:\Docmosis\Tornado (on Linux use /opt/docmosis/tornado).

Before installing Tornado, prepare a directory structure as follows (in this example, the C drive is used):

1. Create a new folder on the **C** drive, and name it **Tornado**. For example: **C:\Docmosis\Tornado**.
2. Copy the installation package contents into this folder:  
Tornado).

This is how the **Tornado** will look:



### 3.2. Starting Tornado

Tornado can be started using the startTornado.bat (startTornado.sh for Linux). The scripts default to using the default Java installation and listen for requests on port 8080.

The start script contains comments showing how these settings can be adjusted. For example, the highlighted lines:



```

startTornado.bat - Notepad
File Edit Format View Help
@echo off
REM Launch Script for Docmosis Tornado
REM Un-comment lines below to change
REM set PORT=-Dport=8090
REM set DEBUG=-Dlog.level=DEBUG
REM Un-comment the next line to use a specific version of Java (Java folder)
REM "C:\Program Files\Java\jdk1.8.0_51\bin\java.exe" %DEBUG% %*
REM allow blank password
REM set DOCMOSIS_ADMINPWALLOWBLANK=true
REM allow UNC paths
REM set DOCMOSIS_ALLOWUNCPATHS=true
java %DEBUG% %PORT% -jar docmosisTornado2.10.0.war
pause
    
```

Optionally Change the Port or enable DEBUG logging

Optionally specify a specific Java version

are comments (starting with REM) and could be uncommented to take effect.

### 3.2.1. Running from the Scripts

1. run the script by typing startTornado.bat and press enter (on Linux use "chmod +x startTornado.sh && ./startTornado.sh")
2. Tornado will start and log some messages including:

*INFO: Tornado started. Listening on 8080 on all network interfaces (or a similar message), to signify that Tornado is running.*

## 3.3. Configuring Tornado via the Web Console

After Tornado is started, the web console is available and can be accessed from a web browser.



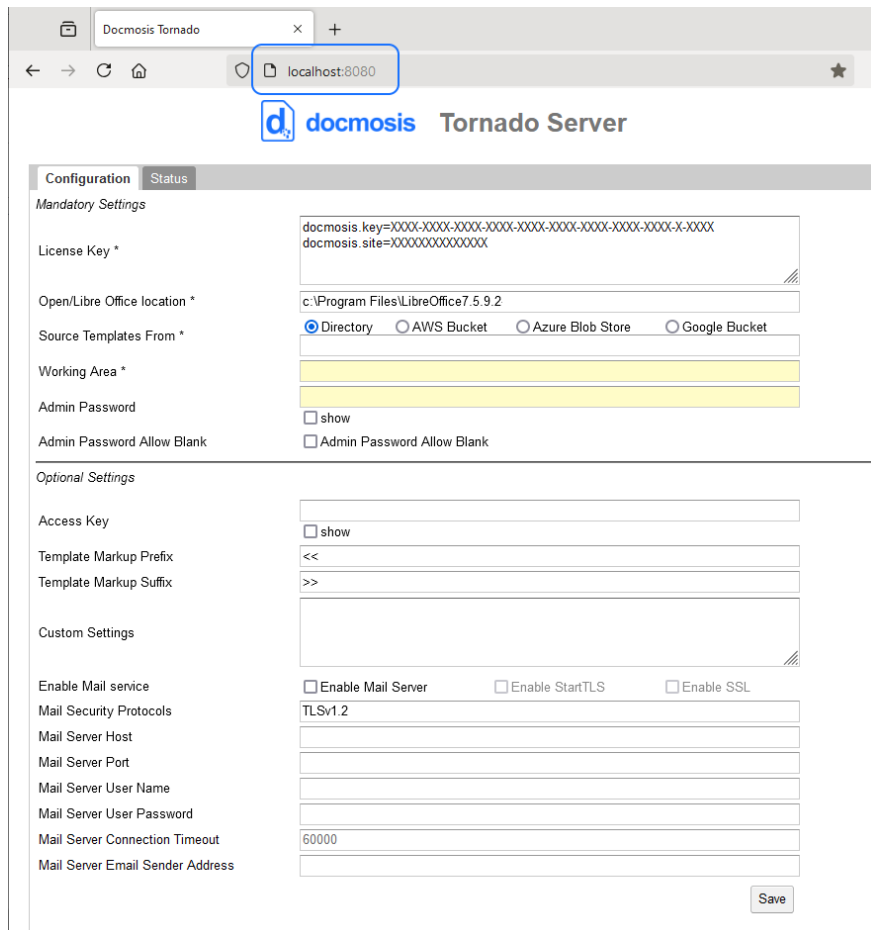
*The following configuration done via the web console could optionally be done via parameters in the launch script (as discussed in section 4 How Tornado Configuration Works).*

1. Open a web browser, such as Chrome or Internet Explorer, and type in the following URL: <http://localhost:8080/>



2. Press Enter.

The Tornado Console opens on the Configuration tab:



3. Set the mandatory values



*Some of the settings are prefilled, but may need adjusting. Edit these if necessary.*

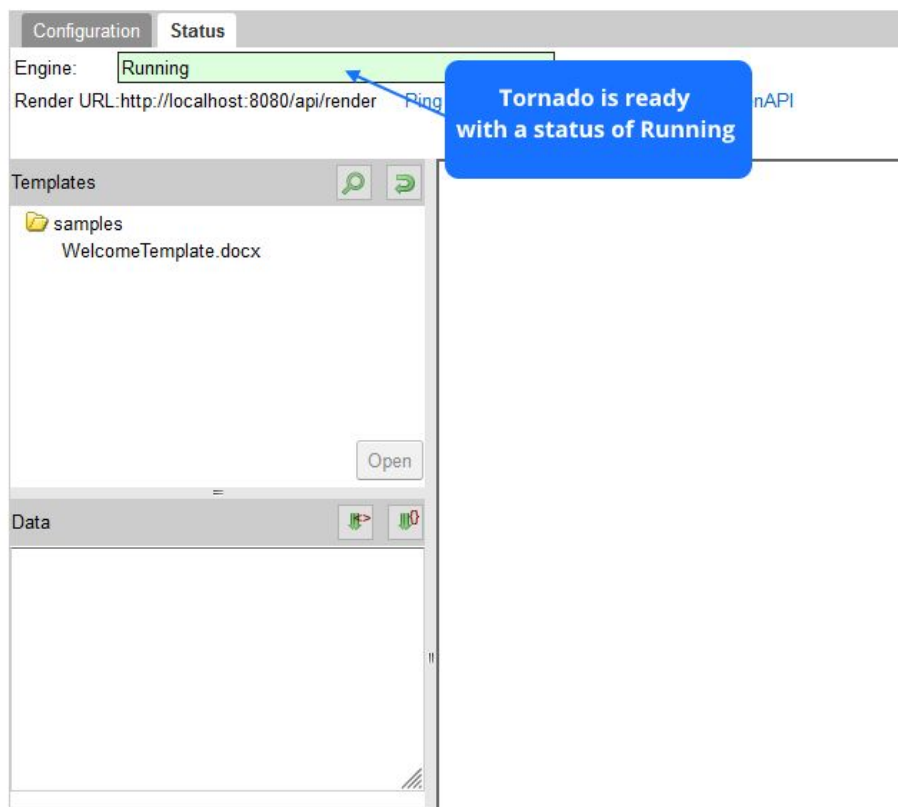
License Key	Copy and paste the complete license key supplied in the email from Docmosis Support. Example:  docmosis.key=XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-X-XXXX docmosis.site=XXXXXXXXXXXXXXXXXX
Open/Libre Office Location	Set the path to the LibreOffice install. For example, "C:\Program Files\LibreOffice", or "/Applications/LibreOffice.app/Contents", or "/opt/libreoffice"



Source Templates From	Specify where to find templates. See section 9.1 Common Settings for details about the options for this setting
Working Area	Specify a working area (e.g. C:\Docmosis\Tornado\Working) for temporary files

- Click **Save**. Docmosis will validate the settings and notify that the save was successful. Tornado must now be restarted (from the script) for the settings to take effect. Return to the command prompt and press control-c to kill Tornado. Then use startTornado.bat (or startTornado.sh) to restart.
- Make sure the startup was successful by refreshing your browser. This time, it should show the Status tab instead.

 docmosis Tornado Server



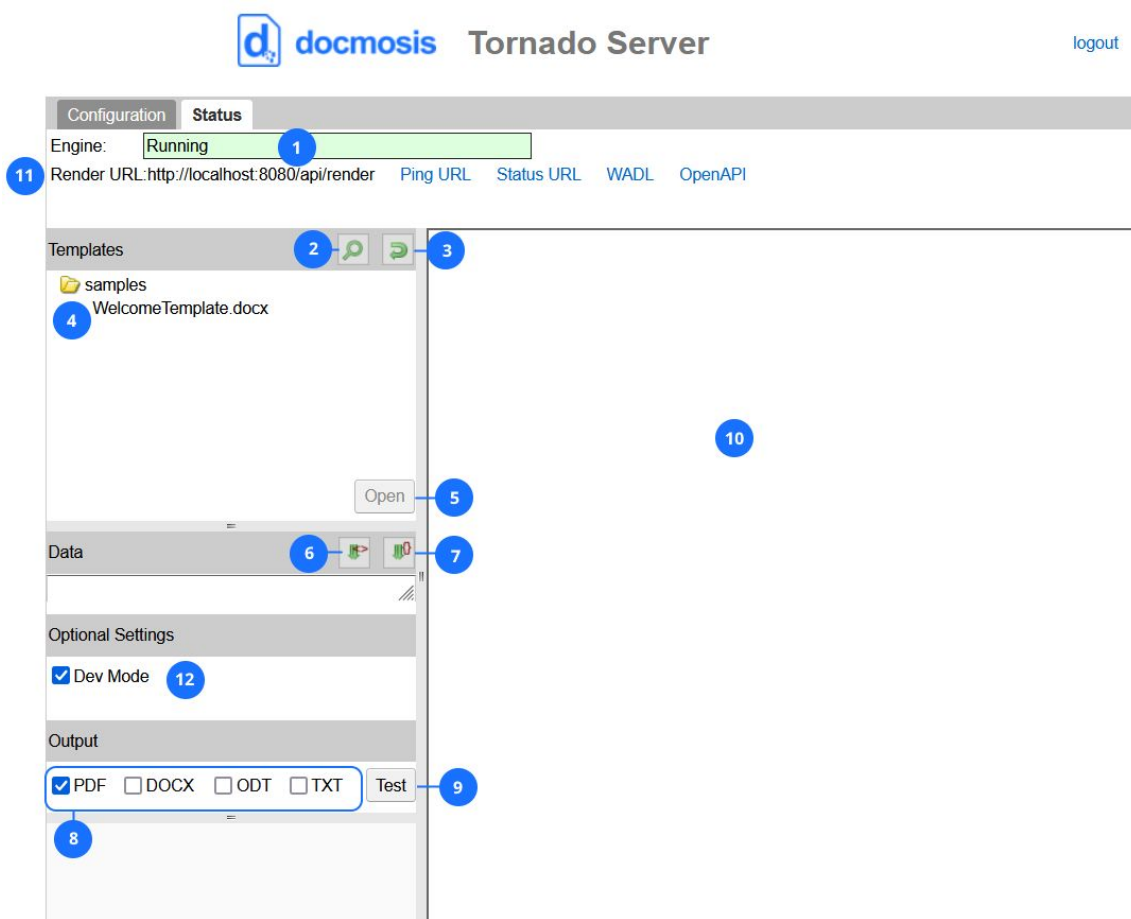
If Tornado has not indicated a running status, examine the log files that will be in your configured Working Area. The logs will provide the diagnostic information about any startup problems.



## 3.4. Using The Tornado Console

### 3.4.1. Overview of the Status Tab

In this screen you can view the templates, view/download templates and execute test renders.



1	Engine status	7	Create dummy JSON data based on template
2	Search in template list	8	Select one or more output formats
3	Refresh template list	9	Generate an output document
4	Select a template to use	10	If PDF format (only) is selected, it appears in this pane
5	Click to open the template	11	Render URL: the URL to use for calling the Render service
6	Create dummy XML data based on template	12	Render in DEV or PROD mode



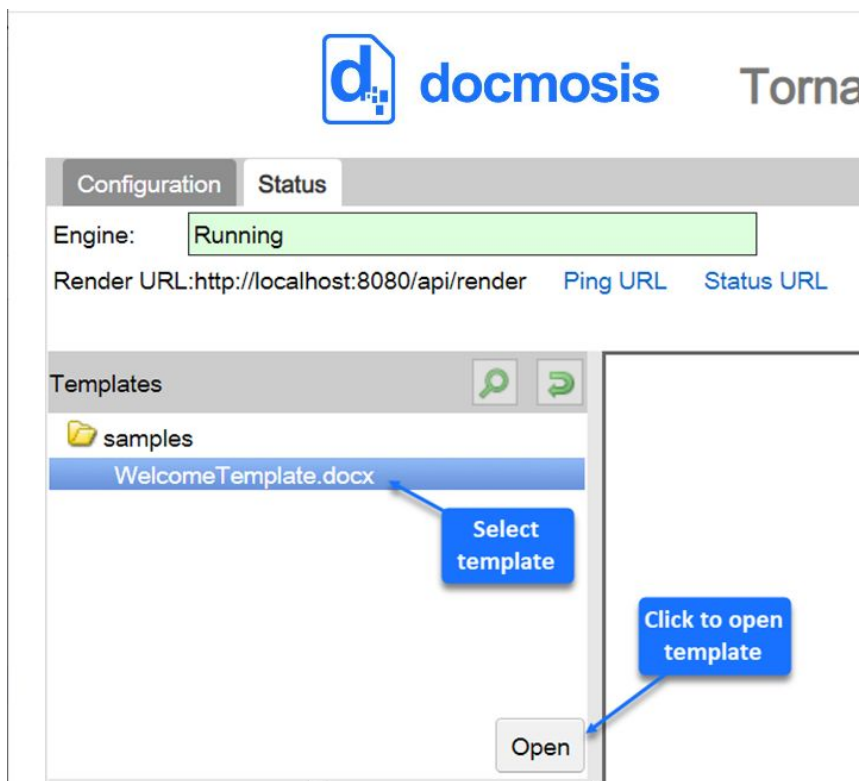
### 3.4.2. Generating a Sample Document

To test the Render service: select a template, supply the required data, choose an output format, then generate the output document. This is detailed below:

1. Select the template that is supplied with the Tornado installation:  
"WelcomeTemplate.docx".



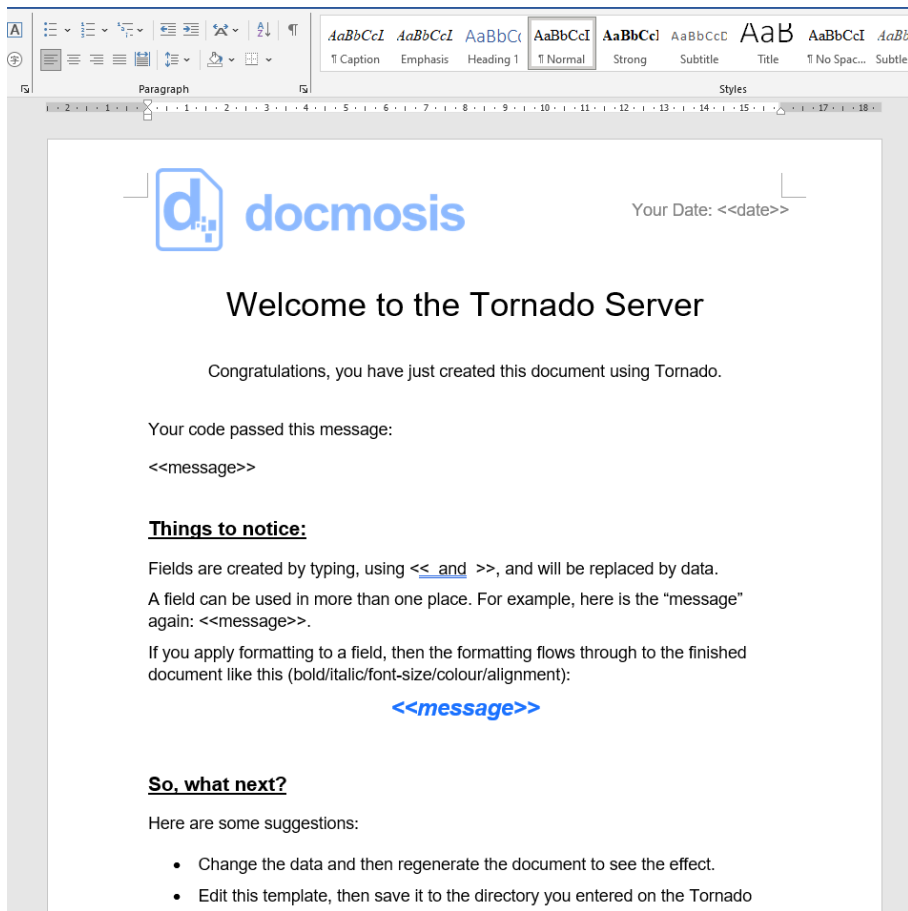
*If you are using a local folder for your configured Template Source, Tornado will automatically copy the WelcomeTemplate.docx template there.*



2. To view the template, click **Open**. This opens the template document in Word in the viewing pane on the right (if the browser is configured for this), or otherwise in a new window.



*The Open button will be replaced with a Download button if the Template Source is not a local folder.*

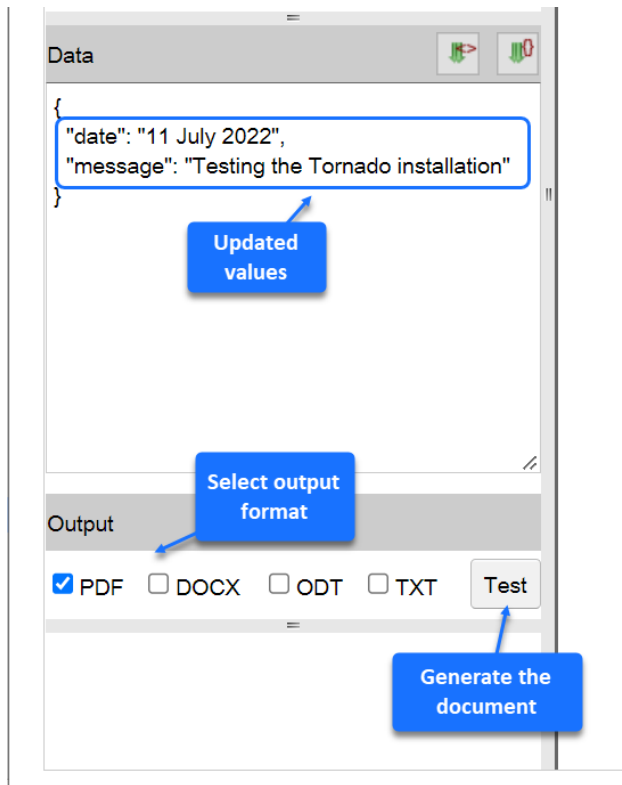


If the file was "Opened" (rather than downloaded), changes you make will affect the actual template being used. This means that rapid changing and testing is possible.

3. Click the "Create dummy JSON data" button (#7 in the earlier diagram), to generate dummy JSON data based on the selected template.
4. Edit the data values in the editor.



5. Select one or more output formats.
6. Click Test to generate the document from the template and data.



The document is generated. If PDF is selected, the generated PDF appears in the viewing pane (if allowed by the browser).



*The supplied data appears in the generated document, replacing the placeholders found in the template.*



docmosis Tornado Server logout

Configuration Status

Engine: Running

Render URL: <http://localhost:8080/api/render> [Ping URL](#) [Status URL](#) [WADL](#) [OpenAPI](#)

Templates

- samples
- WelcomeTemplate.docx

Open

Data

```
{  
  "date": "value1",  
  "message": "value2"  
}
```

Optional Settings

Dev M

**Render Statistics**

Output

PDF  DOCX  ODT  TXT

Render Info

Render Time:	559 millis
Render Size:	89962 bytes
Errors Detected:	false

1 of 1 Automatic Zoom

docmosis Your Date: value1

## Welcome to the Tornado Server

Congratulations, you have just created this document using Tornado.

Your code passed this message: value2

**Things to notice:**

Fields are created by typing, using << and >>, and will be replaced by data.  
A field can be used in more than one place. For example, here is the "message" again: value2.  
If you apply formatting to a field, then the formatting flows through to the finished document like this (bold/italic/font-size/colour/alignment):

*value2*

**So, what next?**

Here are some suggestions:

- Change the data and then regenerate the document to see the effect.
- Edit this template, then save it to the directory you entered on the Tornado Configuration Page under: "Source Templates From". *Note: you will need to refresh the template list on the Status page by hitting the icon.*
- Try out the Example Templates and matching Data here: <https://resources.docmosis.com/example-templates>

Read more about generating a test document in section 5 Generating Documents from the Console.



## 4. HOW TORNADO CONFIGURATION WORKS

---

There are three different ways to set the configuration settings of Tornado.

They are evaluated and used by Tornado in the following order of preference (first to last):

- Via the **Java command line**
- Setting **Environment variables** of the format of Docmosis\_xxx=yyy
- **Previously saved configuration settings**

In addition, many of the configuration settings can be updated in the Web Console's Configuration tab for convenience (particularly when first using Tornado).

When an option is set using the environment or command line parameter, it is applied and automatically saved. This means you don't need to provide the configuration settings every time.

### 4.1. Where Configuration Settings Are Stored

Configuration settings are saved against the account of the user running the server. The configuration is persisted in the Windows registry on a Windows platform and in the user's home directory on Linux/Unix based systems.

**Windows:** HKEY\_CURRENT\_USER\Software\JavaSoft\Prefs\com\docmosis\webserver

**Linux:** ~/.java/.userPrefs/com/docmosis/webserver

**Mac:** /Library/Preferences/com.docmosis.webserver.plist

### 4.2. Changing Configuration from the Java Command Line

This configuration method overrides other previously saved configurations, and also any configuration settings in the Environment Variables.

Use the format **-Dxxx=yyy** to change parameters in the Java Command Line.

**Example:** To set the templates directory templatesDir to /projects/tornado/templates, type the following in the command line:

```
java -DtemplatesDir=/projects/tornado/templates -jar
docmosisTornado[version number].war
```



## 4.3. Changing Configuration by Setting Environment Variables

This overrides previously saved configurations, but is overridden by configuration settings in the Java Command line.

The environment variables are named using the format `Docmosis_xxx=yyy`.

**Example:** To set the templates directory **templatesDir** to **/projects/tornado/templates** use the following command:

In a Linux shell:

```
export DOCMOSIS_TEMPLATESDIR=/projects/tornado/templates
java -jar docmosisTornado[version number].war
```

In a Windows environment:

```
set DOCMOSIS_TEMPLATESDIR=/projects/tornado/templates
java -jar docmosisTornado.war
```



*Environment variables may all be specified in lower or mixed case also (e.g. `Docmosis_templatesDir`).*

## 4.4. Changing Configuration Settings on the Tornado Web Console



*You can only do this after starting Tornado.*

Previously saved settings can be overridden by Setting Environment Variables, or changing configuration in the Java Command Line.

To change the configuration on the Tornado Web Console, update settings and click **Save**. Restart Tornado for the new settings to apply.

**Example:** To set the templates directory **templatesDir** to **/projects/tornado/templates**:

1. Open the Configuration tab in the Tornado Web Console.
2. Select **Directory**, update the directory path for **Source Templates From**.





## 5. GENERATING DOCUMENTS FROM THE CONSOLE

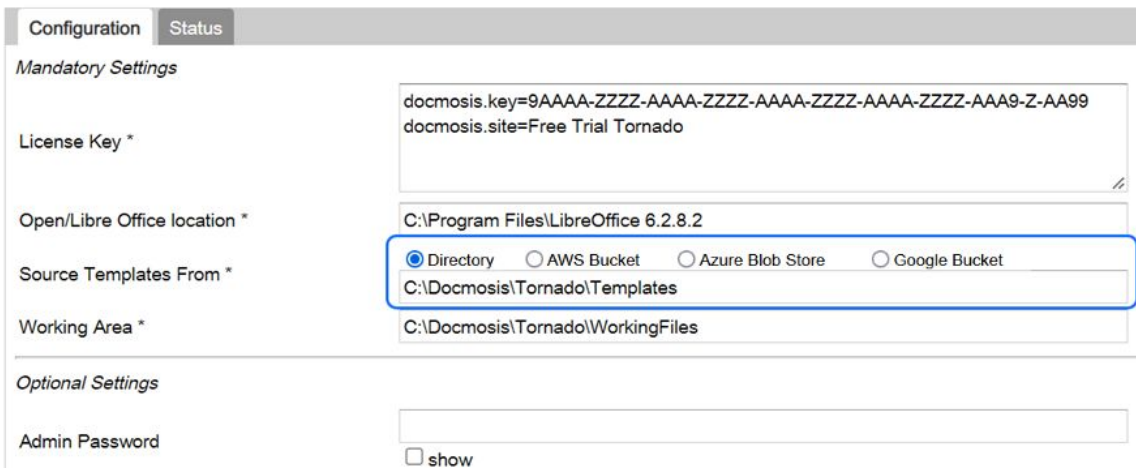
This section provides more detail about rendering documents from the console.

### 5.1. Storing and Editing Templates

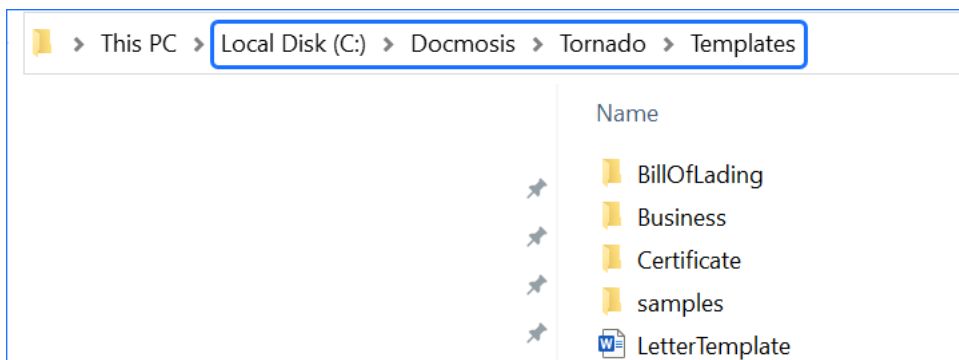
Add template to your configured Template Source location however required. Tornado itself does not provide any method to upload templates – the template source location is considered read only.

To add templates, copy them into a folder using Windows Explorer or upload into your configured Amazon, Azure or Google storage. Tornado will attempt to read templates from the template location.

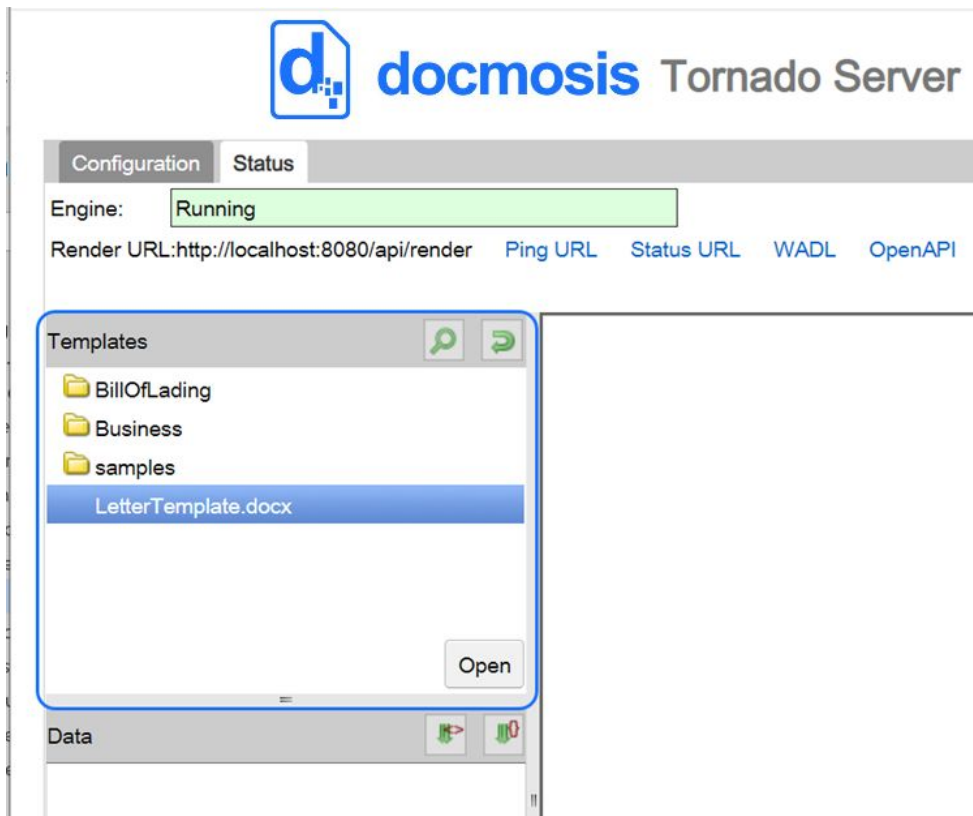
The following example describes working with a local folder to add templates.







For example: in the **Templates** folder, or under a sub-folder of **Templates**.



You can locate, open, and edit templates from the **Templates** panel in the **Tornado Status** tab.



- To view the available templates list, click the **Refresh**  button. All subfolders containing templates are displayed in the templates list.
- To search for a template, click the **Search**  button, then enter text search edit box. Type your search query and press “enter” to execute.
- To open a folder, click on the closed folder  icon. Templates within open folders are displayed. Click on the open folder  icon to close the folder.
- To open a template, click the **Open** button. The template opens in Word. From here you can edit and save changes to the template (only if working with a local folder as the template source).

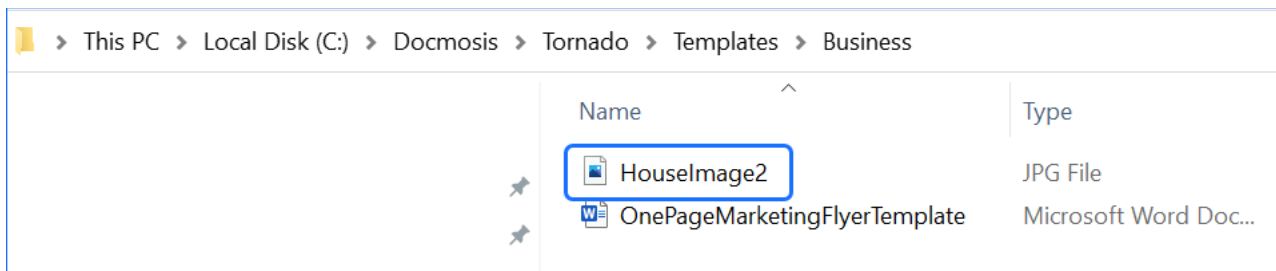


*With **AWS S3**, **Azure Blob** and **Google** storage template locations, Tornado expects a “templates” and an “images” folder to be able to locate templates and images, respectively. If templates are not in a “templates” folder, no templates will be displayed. If images are not in the “images” folder, no stock images will be available.*



## 5.2. Storing Stock Images



To add stock images, copy them into a folder using Windows Explorer. This should be directly in the source templates folder (C:\Docmosis\Tornado\Templates in this example) or one of its sub-folders.



When referencing a stock image in the data, name the path if the image is stored in a sub-folder of **Templates**. Refer to the **Tornado Web Services Guide** for more information on storing and retrieving images in Tornado.

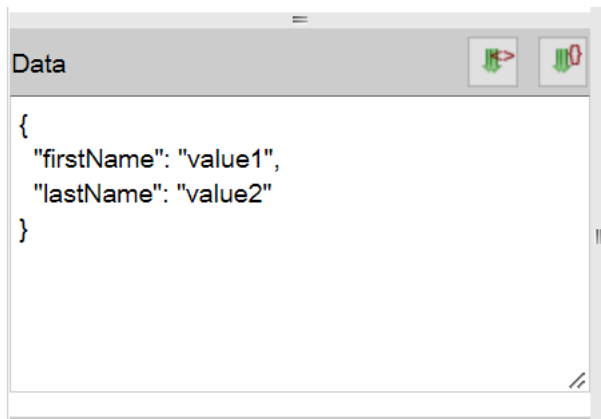
## 5.3. Supplying Test Data

To test the template thoroughly, provide test data to replace placeholders in the template with values.

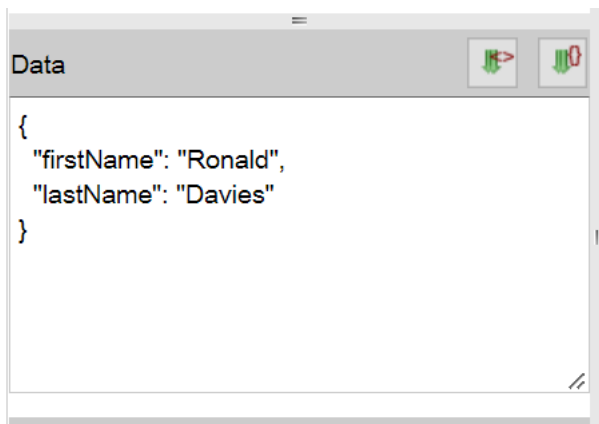
- To generate test data, first select a template. Then click the XML data  button to generate XML test data, or click the JSON data  generator to generate JSON test data. Tornado queries the template for placeholder fields, and attempts to create data key/value pairs to match the template, presenting this in either XML or JSON format.



*Templates can contain complex data structures, such as arrays, so you may need to adjust the generated data structure.*



- Once there is data in the edit box, you can edit the values to look more like the data you would like to see in the test document.



Alternatively, edit a data file to match the data requirements, and copy and paste the contents into the **Data** edit box.

## 5.4. The Dev Mode option

Docmosis can generate documents in two different modes:

- Dev Mode (Development): useful for development, testing, and diagnosing issues
- Prod Mode (Production): errors are reported but no document is produced

See the Web Services Guide for more information about this setting.

## 5.5. Selecting Output Formats

If you render a PDF only and your browser is configured with a PDF viewer, the output file is displayed in the viewing panel on the right.



If you choose any other formats, or combinations of formats, you will receive the rendered document as a download.

1. Select one or more output formats for your document.
2. Click **Test** to generate the document.

The document is rendered, in one or more output formats, and the render statistics are displayed in the **Output** panel.

The screenshot displays the Tornado web application interface. At the top, there are tabs for 'Configuration' and 'Status'. The 'Engine' is set to 'Running'. Below this, the 'Render URL' is 'http://localhost:8080/api/render', with links for 'Ping URL', 'Status URL', 'WADL', and 'OpenAPI'. The 'Templates' section lists 'BillOfLading', 'Business', and 'samples', with 'LetterTemplate.docx' selected. An 'Open' button is visible. The 'Data' section shows a JSON object: 

```
{ "firstName": "Ronald", "lastName": "Davies" }
```

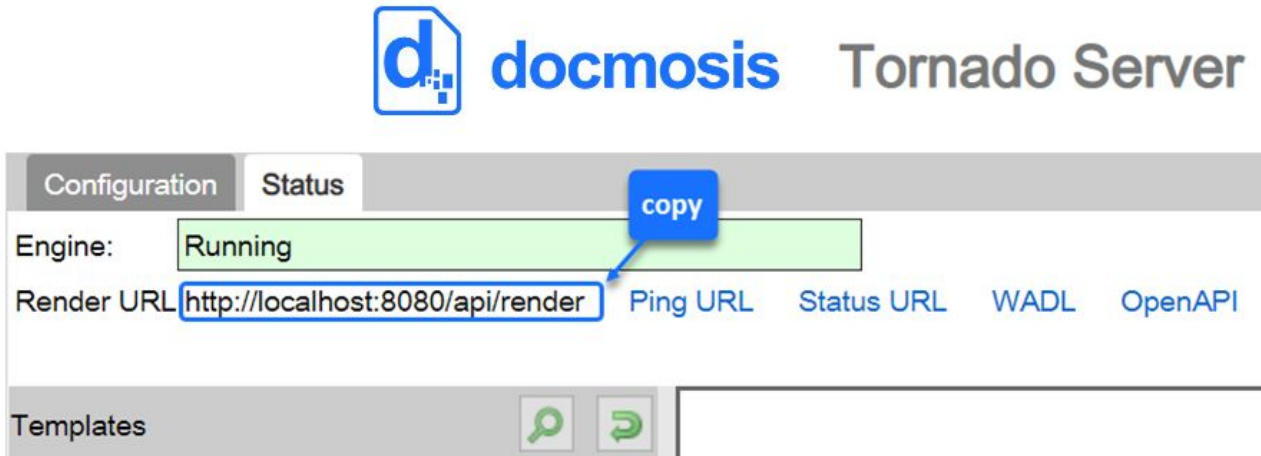
. The 'Optional Settings' section has 'Dev Mode' checked. The 'Output' section has 'PDF' selected, with 'DOCX', 'ODT', and 'TXT' unselected, and a 'Test' button. The 'Render Stats' section shows 'Render Time: 1136 millis' and 'Render Size: 96215 bytes'. On the right, a 'fetch' button is visible. A blue callout box points to the 'fetch' button with the text 'Output PDF displayed in viewing panel'. Another blue callout box points to the 'Test' button with the text 'Render statistics'. A third blue callout box points to the 'PDF' checkbox with the text 'PDF only selected'. The rendered document is displayed on the right, featuring a logo for 'ME LEGAL & FI' and the text: 'Ronald D', 'Dear Ron', 'Thank yo', 'We will b', 'invite for', 'We will c', 'Yours Sin', and a signature 'J.P.' followed by 'Julie Part Human R'.



## 6. GENERATING DOCUMENTS FROM AN APPLICATION

Use the Render URL with your client code/libraries to call the Tornado Render service to render documents.

- Copy the Render URL from the Tornado Status tab.



### 6.1. Invoking the Render Service

Refer to the **Tornado Web Services Guide** for details on invoking the render service. The render service has many more options for the documents being produced than those visible in the Tornado console.



## 7. MONITORING TORNADO

---

Tornado includes two web service end-points to support automated monitoring:

- “ping”, for example: `http://localhost:8080/api/ping`
- “status”, for example: `http://localhost:8080/api/status`

Refer to the **Tornado Web Services Guide** for details about these monitoring end-points.



## 8. COMPARING TORNADO WITH DOCMOSIS CLOUD

---

The primary role of Tornado is to provide a REST API for document generation, the same primary purpose as the Docmosis Cloud Service. Tornado:

- provides a subset of the API functionality available in Docmosis Cloud, but the same core document generation features
- requires installation, configuration and management to meet security, stability and performance requirements. This is an ongoing staffing, process and equipment requirement.

The most important API service (/render) is almost identical between the two products (see section 8.2 below regarding the store-to directives). The documents produced will be the same as via the Cloud, allowing for differences in system configuration (such as software versions and fonts installed).

Tornado does not provide a template-management API. It is the customer's responsibility to manage the templates in their chosen Template Source Location. Tornado treats Template Source Locations as read only, except for the local file system, where default sample templates may be installed and templates opened from the console may be edited directly.

### 8.1. Service Requests in Tornado

Docmosis Tornado provides a limited set of the full set of services available in the Docmosis Cloud Service. Note that the URL used is for directing requests to your local Docmosis Tornado server, instead of the public Cloud service.

The following services are available in Tornado. Refer to the **Tornado Web Services Guide** for the full details and syntax.

- **Render Service:** The primary service used. Render is called to render the document.
- **Get Template Structure Service:** Retrieves the structure of a template that has been uploaded.
- **Convert Document Service:** Allows files to be converted between formats.
- **Ping Service:** Allows monitoring systems to detect whether the Tornado service has been started.
- **Status Service:** Allows monitoring systems to determine the status of the Tornado service, including whether it is ready to render documents.
- **Get Sample Data Service:** Allows data to be generated for a template based on the current structures in the template.



## 8.2. Store-to Directives

Store-to directives for Cloud and AWS (Amazon S3) storage are **not** available in Tornado.

## 8.3. REST Clients

REST clients communicating with Tornado only need to supply an access key if it has been configured in the Tornado server. In Docmosis Cloud, the access key is always required.

## 8.4. Emailing Documents

Emailing of documents is supported via a customer-provided email gateway. When the diagnosing gateway settings have been configured, documents can be delivered by email by setting the applicable render parameters in calls to the render service. See [Enabling Email from Tornado](#).



## 9. ALL CONFIGURATION OPTIONS

### 9.1. Common Settings

The following settings can be set at the command line, and some may be set in the Configuration tab.

Setting	Description
port	Specify the port on which the console and the web services will listen.
license	Specify the Tornado license all as one string. This includes the key and the site and overrides the key and site parameters below. “\n” is used to provide separate lines.  Example: <code>license="docmosis.key=XXXX-XXXX-XXXX-X-XXXX\ndocmosis.site=Free Trial Tornado"</code>
key	Specify the key part of the Tornado license (if the above multiline setting is difficult). This requires the site parameter to also be set.  Example: <code>key="docmosis.key=XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-X-XXXX"</code>
site	Specify the site part of the Tornado license (if the above multiline setting is difficult). This requires the key parameter to also be set.  <code>site="docmosis.site=Free Trial Tornado"</code>
officeDir	Specify the office install location for LibreOffice.
templatesDir	Specify where templates will be sourced from (original templates) If specifying an AWS S3 Bucket, Azure Blob Store or Google Bucket, Tornado will expect folders “templates” and “images” to be able to locate templates and images. If these folders are not present, no templates or images will be found. See sub-sections below for details about configuring each.
workingDir	Specify where logs and working caches are to be stored.
adminPw	Specify the admin password for access the web console (optional).
adminPwAllowBlank	Allow the admin password to be blank (assumes deployed in otherwise secured environment). Optional. Default <b>false</b> .
accessKey	Specify the access key for calling the web service end points (optional).



Setting	Description
<code>customSettings</code>	Specify any custom settings using the format <code>key=value</code> and separating settings by <code>"\n"</code> .  Example: <code>customSettings="docmosis.xyz=abc\ndocmosis.xyz.2=def"</code>
<code>templatePrefix</code>	Specify the template field prefix. Default: <code>&lt;&lt;</code> (must be at least 2 chars).
<code>templateSuffix</code>	Specify the template field suffix. Default: <code>&gt;&gt;</code> (must be at least 2 chars).
<code>installSamples</code>	Specify whether to install sample templates at startup. This only works if the templates are configured to be in a local file system. All other sources of templates are considered read-only to Tornado. Default: <b>true</b> .

### 9.1.1. Templates in a Local Directory

When `templatesDir` is configured to read templates from a local directory, Tornado will find and load templates from the named directory on the host running Tornado.

To configure a local directory, value is simply a path on the local file system. For example:

```
/projects/tornado/templates
```

Templates are sourced from this directory and used for document generation.

One particular feature of using a local file system is that if Tornado is running on the same machine as the user running the browser, templates can be opened, modified and tested rapidly. In this case, Tornado facilitates the user modifying the actual files in the local file system. This makes the Tornado console a very efficient way to develop and maintain templates.

Unlike the other locations below, images and templates are mixed together in the same folders under the configured location.

### 9.1.2. Templates in an AWS S3 Bucket

When `templatesDir` is configured to read templates from ASW S3, Tornado will find and load templates from the named ASW S3 bucket.



The format for the configuration parameter is:

```
s3:bucket;prefix[;accessKey;privateKey]
```

where:

Element	Description
bucket	The name of the bucket
prefix	An optional prefix for accessing items in the bucket. Typically, this would only be required if the bucket also holds other files/folders not intended for Tornado.
accessKey	The access key providing access. This does not need to be provided if Tornado is running in an environment where access to the bucket can be determined from the environment in which Tornado is running.
privateKey	The private key/secret matching the accessKey

For example:

```
s3:my.aws.bucket;;
```

configures access to templates in my.aws.bucket. Templates will be in the “templates” folder and stock images (if any) in the “images” folder. Permissions are expected to be available from the environment.

```
s3:my.aws.bucket;projectA;xxx;yyy
```

configures access to templates in my.aws.bucket. Templates will be in the “projectA/templates” folder and images in “projectA/images”. Permissions are obtained using the xxx and yyy public and secret keys.

If credentials can be obtained from the environment, the access key and private key do not need to be provided in the configuration.

Tornado only requires list and read permissions.

### 9.1.3. Templates in an Azure Blob Store

When *templatesDir* is configured to a read templates from Azure Blob Storage, Tornado will find and load templates from the named Azure Blob Store.

The format for the configuration parameter is:



```
azureblob:containerName;prefix;accountName[;accountKey]
```

where:

Element	Description
containerName	The name of the bucket
prefix	An optional prefix for accessing items in the bucket. Typically, this would only be required if the bucket also holds other files/folders not intended for Tornado.
accountName	The access key providing access. This does not need to be provided if Tornado is running in an environment where access to the bucket can be determined from the environment in which Tornado is running.
accountKey	The private key/secret matching the accountName if required.

For example:

```
azureblob:myAzureBlobstore;;myAccount;myKey
```

configures access to templates in myAzureBlobstore. Templates will be in the “templates” folder and stock images (if any) in the “images” folder. Permissions are obtained using myAccount and myKey credentials.

Tornado only requires list and read permissions.

### 9.1.4. Templates in a Google Bucket

When *templatesDir* is configured to a read templates from a Google Bucket, Tornado will find and load templates from the named Google Bucket.

The format for the configuration parameter is:

```
google:bucketName;prefix;accessKey;secret[;region=us]
```

where:

Element	Description
bucketName	The name of the bucket
prefix	An optional prefix for accessing items in the bucket. Typically, this would only be required if the bucket also holds other files/folders not intended for Tornado.
accessKey	The HMAC access id string providing access. This does not need to



Element	Description
	be provided if Tornado is running in an environment where access to the bucket can be determined from the environment in which Tornado is running.
secret	The private key/secret matching the accessKey.
region	Specify a region if required using region=xxx

For example:

```
google:myGoogleBucket;;myAccessKey;mySecret
```

configures access to templates in myGoogleBucket. Templates will be in the “templates” folder and stock images (if any) in the “images” folder. Permissions are obtained using myAccessKey and mySecret credentials (generated by creating a Google HMAC key in your Google account).

Tornado only requires list and read permissions.

## 9.2. Logging Settings

Logging of information by Tornado can be controlled by several command line settings.

Setting	Description
log.level=debug info error	Specify the level of logging to the console and log files. E.g. java -Dlog.level=debug -jar docmosisTornado.war
java.util.logging.config.file=path	Specify the Java Util logging configuration file. Overrides log.level.  Example: java -Djava.util.logging.config.file=c:/logging.properties ..
log.dir.override	Override the location where logs are to be written.  Default: <b>&lt;working area&gt;/logs</b>

## 9.3. Enabling SSL/TLS Encryption

The following settings can be added to the command line or the Custom Settings on the Configuration page to enable SSL/TLS Encryption.



Setting	Description
<code>ssl.port=port</code>	The port to listen for secured connections.
<code>http.disable=true false</code>	Determines whether the non-secure listener should be disabled. Default: <b>false</b> .

The following settings are built-in Java settings and are applicable for specifying a key store or trust store other than the default in the Java installation. When implementing HTTPS security custom certificates may be required and expertise in setting up and configuring the following is required.

Setting	Description
<code>javax.net.ssl.keyStore=path</code>	The path to the key store file.
<code>javax.net.ssl.keyStorePassword=password</code>	The key store file password.
<code>javax.net.ssl.trustStore=path</code>	The path to the trust store file.
<code>javax.net.ssl.trustStorePassword=password</code>	The trust store file password.

For more details on these settings, refer to your Java documentation.

## 9.4. Other Network Settings

Setting	Description
<code>address.listen</code>	Specify the address on which to listen for connections. This is useful when multiple networks are present and only a specified network should accept connections. By default, all networks are used.
<code>keepAlive</code>	Specify whether to set KeepAlive (SO_KEEPALIVE) on network connections. Values are <b>true/false</b> . Default: <b>false</b> .
<code>httpContext</code>	Allows the api context to be customized in Tornado URLs. For example, a Tornado URL by default might be: <code>http://localhost/</code> and setting <code>httpContext=tornado1</code> would change the URL to: <code>http://localhost/tornado1</code>



Setting	Description
	This facilitates running Tornado behind a load balancer that also services other applications. The context can be used to route only some requests to Tornado. The default is no context.

## 9.5. Enabling Email from Tornado

Set these parameters to enable emailing of documents in Tornado.

Setting	Description
<code>mailEnabled</code>	Enable the mail server. Default: <b>false</b> .
<code>mailHost</code>	The mail server hostname.
<code>mailPort</code>	The mail server port.
<code>mailUser</code>	The mail server username.
<code>mailPw</code>	The mail server password.
<code>mailFrom</code>	The from email address.
<code>mailTimeout</code>	The mail server connect-timeout in milliseconds.
<code>mailUseTls</code>	Enable TLS security on the connection to the mail server. Default: <b>false</b> .
<code>mailUseSsl</code>	Enable SSL security on the connection to the mail server. Default: <b>false</b> .
<code>mailConnectRetryMaxTimes</code>	Set the maximum number of attempts to connect to the mail server. Default: <b>2</b> .
<code>mailConnectRetryMinWaitMillis</code>	Set the minimum wait time to get a connection to the mail server. Default: <b>5000ms</b> .
<code>mailConnectRetryMaxWaitMillis</code>	Set the maximum wait time to get a connection to the mail server. Default: <b>20000ms</b> .
<code>mailConnectRetryRebuildTransport</code>	Rebuild the Message Transport object on failure to connect. Default: <b>true</b> .
<code>mailSendRetryMaxTimes</code>	Set the maximum number of attempts to send email. Default: <b>2</b> .
<code>mailCustomHeadersAdd</code>	Add a custom mail header. Default: <b>true</b> .



Setting	Description
mailCustomHeadersName	Set the custom header name. Default: <b>X-DWS-Tag-1</b>

## Docmosis Pty Ltd

### **Address**

Suite 8 / 5 Hasler Road,  
Osborne Park,  
WA 6017 Australia

### **Website**

<https://www.docmosis.com>

### **Resources**

<https://resources.docmosis.com>