

Glasswall Web API

User Guide

# Document Control

|  |  |  |
| --- | --- | --- |
| **Date Issued** | **Version** | **Reason For Change** |
| 26th September 2018 | 0.1 | Initial Version |

Contents

[1. Document Control 2](#_Toc530562806)

[2. Introduction 4](#_Toc530562807)

[2.1 Supported File Types 4](#_Toc530562808)

[2.2 Content Management 4](#_Toc530562809)

[2.3 Embedded Content 4](#_Toc530562810)

[3. Web API Installation 5](#_Toc530562811)

[3.1 Linux 5](#_Toc530562812)

[3.1.1 Prerequisites 5](#_Toc530562813)

[3.1.2 Installation 5](#_Toc530562814)

[3.2 Windows 5](#_Toc530562815)

[3.2.1 Prerequisites 5](#_Toc530562816)

[3.2.2 Installation 5](#_Toc530562817)

[4. Configuration 7](#_Toc530562818)

[5. API Reference 8](#_Toc530562819)

[5.2 Result Types 8](#_Toc530562820)

[5.2.1 Error Messages 8](#_Toc530562821)

[6. Requests 9](#_Toc530562822)

[6.1 Analysis 9](#_Toc530562823)

[6.2 Protect 9](#_Toc530562824)

[6.3 Determine File Type 9](#_Toc530562825)

[6.4 Request Syntax – Analysis Or Protect 10](#_Toc530562826)

[6.4.1 Content Management 11](#_Toc530562827)

[6.4.2 Example Request 13](#_Toc530562828)

[6.5 Response – Analysis Or Protect 14](#_Toc530562829)

[6.6 Request Syntax – Determine File Type 15](#_Toc530562830)

[6.7 Response – Determine File Type 15](#_Toc530562831)

[7. Limitations 16](#_Toc530562832)

[8. Monitoring 17](#_Toc530562833)

# Introduction

The Glasswall Engine inspects files by comparing them to their published specification. It has the following main functions:

* The correction of deviations from the published specification before regenerating a new, reconstructed file.
* The production of reports detailing deep analysis of documents that highlights any deviations from the published specification.
* The management of documents against a configurable content management policy.
* The production of detailed process reporting.

The Glasswall API provides direct access to the Glasswall Engine via a web service.

## Supported File Types

The following file types can be protected by the Glasswall API:

|  |  |  |
| --- | --- | --- |
| **Category** | **File Type** | **Extensions** |
| Microsoft Office (Legacy) | Microsoft Word | doc, dot |
| Microsoft Excel | xls, xlt, xlm |
| Microsoft PowerPoint | ppt, pot, pps |
| Microsoft Office (2007+) | Microsoft Word | docx, dotx, docm, dotm |
| Microsoft Excel | xlsx, xltx, xlsm, xltm |
| Microsoft PowerPoint | pptx, potx, ppsx, pptm, potm,  ppsm |
| PDF | Adobe PDF | pdf |
| Images | JPEG | jpeg, jpg, jpe |
| PNG | png |
| GIF | gif |
|  | EMF | emf |
|  | WMF | wmf |

## Content Management

The Glasswall API contains a powerful Content Management engine that can be utilised by client applications to remove common threat vectors such as macros and JavaScript from files before regenerating them.

## Embedded Content

The Glasswall engine is also able to process embedded content. For example, if a document contains embedded content (such as jpeg images) those files will also be analysed or regenerated and reinserted into the regenerated document.

# Web API Installation

## Linux

The Glasswall API can be installed in Linux Server environments. The application has been tested on the following Linux (64-Bit) distributions:

* Red Hat Enterprise Linux 6.x
* Red Hat Enterprise Linux 7.x
* Amazon Linux 2018.x
* CentOS Linux 6.x
* CentOS Linux 7.x



Note: The application may work on other Linux distributions but testing has been conducted on the operating systems listed above.

### Prerequisites

The following pre-requisites are required:

* glibc-2.12 or later must be installed.

### Installation

Installation of the Glasswall API is completed using the “glasswallapi-X.X-X.x86\_64.rpm” RPM file. The command listed below has to run with **admin privileges** as the **software is installed in system folder and the daemon application has to be configured to launch on start-up of the machine**.

1. To begin, open a terminal in the location containing the RPM file.
2. Run the following command with admin privileges to install the software:

*>> rpm –i glasswallapirhelX-X.X-X.x86\_64.rpm*

## Windows

The Glasswall API can be installed on Windows environments. The application has been tested on the following Windows (64-Bit) OS versions.

* Windows 7
* Windows 10
* Windows Server 2008 R2
* Windows Server 2012 R2
* Windows Server 2016



Note: The application may work on other Windows OS versions but testing has been conducted on the versions listed above.

### Prerequisites

The following pre-requisites are required:

* Microsoft C++ 2015 Redistributable (x64) must be installed.

### Installation

Installation of the Glasswall API is completed using the “glasswall.core.installer(x64 EN).exe” setup file:

1. To begin, run the setup file and click the “Next” button on the welcome screen.
2. On the End-User Licence Agreement screen, read the agreement and click the “I Agree” button to automatically start the installation.
3. Once the installations complete, the user may click the show details button to display the installation information.
4. On completion, click the close button to close the installation wizard.



Note: Elevated privileges may be required upon executing the installer(s) to permit the installation and start-up of the Glasswall Web Server Service(s).

# Configuration

Glasswall Web API configuration tool must be executed to initialise the necessary settings required for the application to run. The config tool has to run with elevated permissions as it is responsible for the starting of the service/daemon if they are not running. The config tool also is responsible to apply the new changes to the web API config file as requested by the user. The steps to do so, open the command prompt or terminal and execute the following command:

* Windows Command: *glasswall.web.server.config*
* Linux Command: *glasswall.web.server.config.tool*

During execution the user will prompted to enter several settings which are listed and described in the table below. Upon confirmation of the provided settings the server will be launched automatically.



Note: The user can manually stop/restart/start service using windows services or via the use of terminal in Linux to do the same for the daemon.

Service name (Windows): glasswall.web.server.service

Service name (Linux): glasswall.service

The following settings must be applied before the server application can be launched.

|  |  |
| --- | --- |
| **Setting** | **Description** |
| Host IP Address | Host Address (where the application was installed) |
| Listening Port | The TCP Port (where to listen for requests on) |
| Enable SSL | Flag to enable SSL |
| SSL Windows Configuration | If SSL is enabled, on the Windows OS the certificate thumbprint for the certificate that is to be used must be provided. |
| SSL Linux Configuration | If SSL is enabled, on the Linux OS a full file path to the certificate (PEM) file and a full file path for the certificate private key (PEM) file must be provided. |
| Enable Logging | Flag to enable application logging. |
| Change Library Path | Flag for whether to change the library path or use the defaults. (This will require changing if the installation is not made to the default install directory) |

# API Reference

The Glasswall Web API has three URLs that can be called, all URLs can either use the default content management configuration contained in the Glasswall Engine or the specific configuration parameters supplied as part of the API call, if it is present. The URLs are following:

* glasswall/protect
* glasswall/analysis
* glasswall/determine\_filetype

A full URL may look as follows,

*http(s)://<API URL>/glasswall/protect*

All requests to the API are sent in the form of a JSON string using the POST request action. The following HTTP header will need to be included in the request:

*Content-Type: application/json*

## 5.2 Result Types

The analyse, protect and determine file type API calls have an element called “Result” in their response. The value of the Result element can be one of the following:

* Success – The API call was successful, and the supplied file was successfully processed by the Glasswall engine.
* Fail – The API call was successful but the Glasswall engine detected issues in the supplied file.

### 5.2.1 Error Messages

The analyse, protect and determine file type API calls have an element called “ErrorMessage” in their response. This element is only populated if the “Result” element returns the value ‘Fail’ and provides a high-level error description of the failure.

The table below contains an example of some of the error description that can be returned.

|  |
| --- |
| **Error Description** |
| Unsupported or unrecognized file type. |
| Error detected when processing the file. |
| Content management configuration error. |

# Requests

## Analysis

The Analysis API call provides a detailed report about a file, reporting on the content of the document if it is processed with the ‘glasswall/analysis’ URL. The decoded base64 string provided in the data element of the response JSON is in the form of an XML file. The generated XML report contains the following:

* ‘Content’ Items - structures found in the file.
* ‘Issue’ Items - structures that do not match the specification.
* ‘Sanitisation’ Items - items that would be taken out as per the content management policy.
* ‘Remedy items’ - structures automatically corrected back to standards defined in the public specifications.

This data can be used for reporting purposes or for making a decision about whether to call the ‘glasswall/protect’ URL on the file.

More information on the analysis mode and as well as the output XML schema can be found in the Glasswall SDK documentation that comes installed with this installation and can be found at the following location:

* Location on Window – “C:\Program Files\Glasswall\{version}\documentation”
* Location on Linux – “/opt/Glasswall/{version}/documentation”

## Protect

The Protect API provides the ability to decompose a file as per Analysis Mode but the file is then regenerated. During this process the file is validated and remediation’s are applied where deviation from the file formats specification are identified. As part of this process the user configured content management rules are applied so that certain items are removed (e.g. macros) from the regenerated file so only known good items are present in the new file.

## Determine File Type

The Determine File Type API provides the ability to determine the file type for a given file provided that the file format is supported by the Glasswall engine. This enables the user to determine the correct file type for the file regardless of the presence of a file extension or an incorrect extension.

## Request Syntax – Analysis Or Protect

The request syntax for the Protect File API call is the same as the Analysis API call. The following URLs must be used:

*HTTP POST: http(s)://<API URL>/glasswall/analysis*

*or*

*HTTP POST: https://<API URL>/glasswall/protect*

The JSON for the request should be sent in the following format:

{

"Configuration":

[

{

"FileType": "{File Type}",

"Settings":

[

{

"Switch": "{Content Management Setting}",

"Setting": "{Content Management Action}"

}

]

}

],

"FileProcessing":

{

"FileName": "{File Name}",

"Data": "{File Data as Base64 Encoded String}"

}

}

### Content Management

The following *Content Management Settings* are available for each *File Type*

|  |  |  |
| --- | --- | --- |
| **FileType** | **Setting** | **Description** |
| PDF | watermark | Used to apply a watermark to the regenerated  PDF file. |
| javascript | Embedded JavaScript code. |
| acroform | Forms that accept user input |
| internal\_hyperlinks | Hyperlinks that reference other parts of the  document. |
| external\_hyperlinks | Hyperlinks that reference an external resource  such as a web page. |
| embedded\_files | Files that have been embedded into the  document. |
| metadata | Properties about the document such as the  name of the author and creation date etc. |
| actions\_all | All remaining actions that aren't controlled by  specific content management flags. |
| Word, Excel, PowerPoint | internal\_hyperlinks | Hyperlinks that reference other parts of the  document. |
| external\_hyperlinks | Hyperlinks that reference an external resource  such as a web page. |
| embedded\_files | Files that have been embedded into the  document. |
| metadata | Properties about the document such as the  name of the author and creation date etc. |
| macros | Executable code within the document. |
| review\_comments | Comments added to the document by the  author or a 3rd party reviewer. |
| Word, Excel | dynamic\_data\_exchange | DDE links within the document. |

Each of the individual Content Management Settings can have one of 3 Actions applied:

* + Sanitise – Remove or make the content benign. For example, metadata and macros will be removed whereas hyperlinks will be converted to plain text.
  + Allow – Include the content in the processed document unchanged.
  + Disallow – Add an issue to the document if content of this type is found in the document**. If the ‘glasswall/protect’ URL is called these issues will stop the file from being regenerated**.
* One exception to the above is the PDF Watermark – the Action for this should be the text of the watermark you want applied to the document.



Note: Multiple elements can be included beneath the FileType and Settings elements. These elements are optional, if not specified default settings present in the Glasswall Engine shall be used.

### Example Request

An example request is shown below:

{

"Configuration":

[

{

"FileType": "DOCX",

"Settings":

[

{

"Switch": "embedded\_files",

"Setting": "sanitise"

},

{

"Switch": "internal\_hyperlinks",

"Setting": "sanitise"

},

{

"Switch": "external\_hyperlinks",

"Setting": "sanitise"

}

]

},

{

"FileType": "XLS",

"Settings":

[

{

"Switch": "embedded\_files",

"Setting": "sanitise"

},

{

"Switch": "internal\_hyperlinks",

"Setting": "allow"

},

{

"Switch": "external\_hyperlinks",

"Setting": "disallow"

}

]

}

],

"FileProcessing":

{

"FileName": "Clean.png", "Data":"iVBORw0KGgoAAAANSUhEUgAAAAsAAAALBAMAAABbgmoVAAAAGFBMVEX///8AUq0AY8Y5nP+MjIyUzv/W7////87upOv2AAAAAXRSTlMAQObYZgAAAAFiS0dEAIgFHUgAAAAJcEhZcwAACxIAAAsSAdLdfvwAAAAHdElNRQfTAQICGAfQ7PyOAAAANElEQVR42mNggAJGQUEQJSgoKADiKCkpAimhEqVAEOVoDKaUTUGUknIYUI5BKDRIAKEPDAC1oAUVPU2kXQAAAABJRU5ErkJggg=="

}

}

## Response – Analysis Or Protect

The response will be in the following format:

{

"ErrorMessage": "{Error Message}",

"File": {

"Data": "{Output File Data Or Report As Base64 Encoded String}",

"FileName": "{File Name}",

"FileType": "{File Type}"

},

"Result": "{Result}"

}

* **FileName** – The filename of the file that was submitted to the API.
* **Result** – The result of the API call. See the Result Types section for more information.
* **ErrorMessage** – The error message if an error occurred, otherwise blank.
* **Data** – The output file from the Glasswall engine as a base64 encoded string. If the protect API call is made this will be the regenerated file, if the analysis call is made this will be the analysis XML report.
* **FileType** – The file type of the file processed by the Glasswall engine.

The following is an example response for a file that was sanitised, in this case the Glasswall Engine.

{

"ErrorMessage": null,

"File": {

"Data": "0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAABAAAAEwAAAAAAAAAAEAAA/v///wAAAAD+////AAAAABQAAAD///////...",

"FileName": "Test.doc",

"FileType": "doc"

},

"Result": "Success"

}

In analysis the Data element of the JSON would contain a base64 encoded XML report.

## Request Syntax – Determine File Type

*HTTP POST: https://<API URL>/glasswall/determine\_filetype*

The JSON for the request should be sent in the following format:

{

"DetermineFileType":

{

"FileName": "{File Name}",

"Data": "{File Data as Base64 Encoded String}"

}

}

## Response – Determine File Type

The response will be in the format:

{

"ErrorMessage": {Error Message},

"FileName": "{File Name}",

"FileType": "{File Type}",

"Result": "{Result}"

}

* + **FileName** – The filename of the file that was submitted to the API.
  + **ErrorMessage –** The error message if an error occurred, otherwise blank.
  + **FileType –** The file type of the file processed by the Glasswall engine.
  + **Result –** The result of the API call. See the [Result Types](#_5.2_Result_Types) section for more information.

# Limitations

Due to the nature of the product, whilst the Web API can handle concurrent connections there is a limit on the number of concurrently processed requests, as such when multiple request come in they are queued and processed in turn if the limit is reached. Whilst the API can safely handle high loads, an increase in load can result in a linear increase in response time. We recommend if a customer is expecting a very high volume of files, they install multiple Web API instances across several servers and load balance the requests accordingly.

# Monitoring

The Glasswall API logs events to both the file system and the Windows Event Log for the windows OS or the Syslog for Linux OS. Service related issues/information is logged to the Windows event logs or the Linux system logs. Application level issues and information is logged to the file system log file.

There are several different levels that the API can log at within the log file stored on the file system:

* **Error** – Events that cause an error but do not cause the API to become inoperable. These events should be investigated as soon as possible.
* **Warn** – Events that an administrator should be aware of but are not error conditions.
* **Info** – Informational events that do not require any intervention from an administrator.