

**Aspose for Cloud**

REST APIs for Word

Version 1.0

***Design Guide***

**Revision History**

|  |  |  |
| --- | --- | --- |
| Revision | Date | Comments |
| 1.0 | 07-Aug-2016 | Consists of the design/implementation details of Word REST APIs for Cloud version 1.0 |

Contents

[1. Introduction 4](#_Toc458399780)

[2. Purpose 4](#_Toc458399781)

[3. Scope 4](#_Toc458399782)

[4. Implementation 4](#_Toc458399783)

[4.1 Convert Controller 4](#_Toc458399784)

[4.2 Protection Controller 5](#_Toc458399785)

[4.2.1 GET Request 5](#_Toc458399786)

[4.2.2 POST Request 5](#_Toc458399787)

[4.3 Storage Controller 6](#_Toc458399788)

[4.4 Common Project 6](#_Toc458399789)

[4.5 Authentication Header 6](#_Toc458399790)

[4.6 Models 6](#_Toc458399791)

# Introduction

These cloud services are provided through standard HTTP REST Web APIs which makes it more user-friendly to consume them. Moreover this provides the leverage of platform, device and language independence.

# Purpose

Objective of writing this guide is to let developers and other technical guys explain all the bits participating in the solution. It elaborates the implementation details of core APIs and the helping classes so they should be able to provide technical assistance in fixing any issue or enhancing them.

# Scope

Scope of this guide is limited to providing the design and implementation details of these APIs. Installation and deployment details can be found in the ***Installation Guide*** which is provided separately.

# Implementation

In this version we are providing two APIs to convert a document into other formats and to apply certain protections on them. We have developed these APIs using ASP.Net. Currently there are three controllers to achieve this functionality:

## Convert Controller

This controller is used to implement the functionality of Convert API which provides the option of conversion of documents into multiple formats like DOC, DOCX, RTF, TIF, JPEG, PDF, XPS, EPUB, etc. For this purpose a PUT request is available. Find below is the detail of parameters it takes to perform its function:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Parameter Type | Data Type | Comments |
| inputfile  (Mandatory) | Query String | String | This parameter accepts the source document name which needs to be converted. |
| format  (Mandatory) | Query String | String | Desired format needs be provided against this parameter like bmp, pdf, jpeg, etc. |
| outpath  (Optional) | Query String | String | This is an optional parameter which can be supplied to save the converted document with a different name and to a sub folder e.g. /MyFolder/MyFile.pdf |

**Example:**

*PUT /Word/Convert/?inputfile=abc.docx&amp;format=pdf&amp;outpath=abc.pdf*

*Host: localhost:50221*

*API\_KEY: 1744faaa-e5bf-47f4-a955-dc164b1ad5d1*

*APP\_SID: 0a4252a0-7e70-11d0-a5d6-28db04c10000*

## Protection Controller

This controller contains the implementation of two requests i.e. GET and POST related to protection mechanism.

### GET Request

GET request returns the protection currently set on the selected document.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Parameter Type | Data Type | Comments |
| DocName  (Mandatory) | Resource Name | String | Document name needs to be specified as resource identifier/name i.e. Word/{Document Name}/Protection. |
| Folder  (Optional) | Query String | String | This is an optional parameter which needs be supplied incase the document resides in subfolder. |

**Example**

*GET /Word/abc.docx/Protection/*

*Host: localhost:50221*

*API\_KEY: 1744faaa-e5bf-47f4-a955-dc164b1ad5d1*

*APP\_SID: 0a4252a0-7e70-11d0-a5d6-28db04c10000*

### POST Request

POST request is used when certain protection is desired to be set on the document.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Parameter Type | Data Type | Comments |
| DocName  (Mandatory) | Resource Name | String | Document name needs to be specified as resource identifier/name i.e. Word/{Document Name}/Protection. |
| Protection Type  (Mandatory) | Request Body | String | This parameter is used to specify the protection type needs to be applied on the document i.e. AllowOnlyComments, AllowOnlyRevisions, ReadOnly and NoProtection. |
| Password  (Mandatory) | Request Body | String | This parameter accepts the current password applied on the document. |
| New Password  (Optional) | Request Body | String | New password needs to be provided in case a password is desire to be applied otherwise a blank string is supplied to clear the password. |
| Folder  (Optional) | Query String | String | This is an optional parameter which needs be supplied incase the document resides in subfolder. |

**Example**

*POST /Word/abc.docx/Protection HTTP/1.1*

*Host: localhost:50221*

*API\_KEY: 1744faaa-e5bf-47f4-a955-dc164b1ad5d1*

*APP\_SID: 0a4252a0-7e70-11d0-a5d6-28db04c10000*

*Content-Type: application/json*

Note: Both of the Convert and Protection controllers use the corresponding methods of Aspose.Words DLLs.

## Storage Controller

This controller comprises of only one POST request which is used to upload the document to the server. Documents are stored in a subfolder which is mentioned against a key “DOCUMENTS\_FOLDER” in web.config under appSettings.

## Common Project

There is a project named as Common in this solution which contains couple of utility classes. Header class is designed to get the value against a key in request header. There is also a class App which is used to get the name of documents folder from web.config.

## Authentication Header

To prevent the usage of APIs from unwanted consumers an authentication mechanism is also implemented. For this purpose a pair of keys is required to be passed in HTTP request header. These are API Key and Application Security ID (named as APP\_KEY and APP\_SID respectively) and are saved in web.config file. Whenever a request is made to an API, system checks that if the request header contains the valid keys? And proceed otherwise an exception is thrown. Note: Currently the verification of keys is implemented inside every request so separate sets of keys can be defined and used for every request.

## Models

Different models are defined to send data to requests and to get data returned from them. ResponseMessage is used to get the response from PUT request of Convert and Convert requests return data via ProtectionResponse. ProtectionInfo is used to send the protection information to POST request of Protection.

END