Robobraille API Guidelines and Test Specifications

Sensus Aps

Author: Paul Cosma
Email: paul@sensus.dk

Summary

This document provides guidelines and test specifications for running the Test Cases against the current release of the Robobraille Web API. Please read through this document before working with the test cases. The test cases can be found in as an appendix to this document: A1-RoboBrailleWebApi.TestCases.xslx

Summary
1.Component controllers
2.Prerequisites
3.General Structure of the Solution
4.Supported request methods
    4.1 Post
    4.2 GetJobStatus
    4.3 GetJobResult
    4.4 Other request methods
5.Workflow for the user tests
6.Valid set of input parameters for each controller
7.The test cases
8.Other appendix files

Please use the following link for a better overview of the current API functionality:
http://{localhost:port}/swagger/ui/index

1.Component controllers

The API component controllers being tested are:

1. AccessibleConversion
2. Audio
3. Braille
4. DaisyConversion
5. EBook
6. Generic (Not relevant for this document)
7. HTMLtoPDF
8. MSOfficeConversion
9. OcrConversion
10. HTMLToText

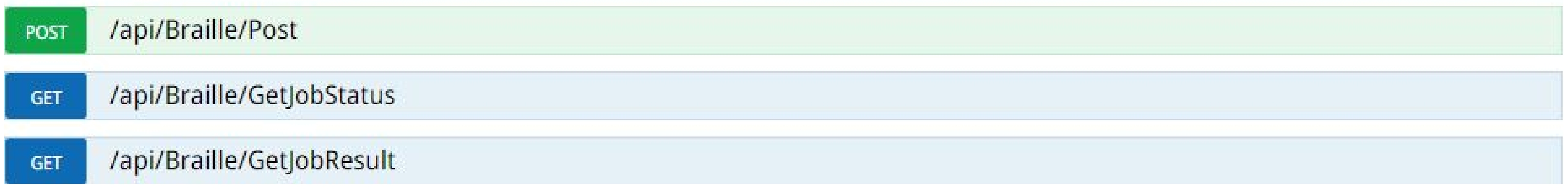
For the purpose of this document the name “Controller” and “Controllers” will be used when providing general information about all of the above components. Information relevant to each component will be provided by referencing the above list of names.

2.Prerequisites

In order to test the API it is recommended to use Postman for creating requests to the Controllers.
https://www.getpostman.com/
Other tools can also be used to access the Web API.

3.General Structure of the Solution

For now all of the Controllers reside under the “http://{localhost:port}/api” namespace.
Each Controller can be referenced by appending the name of the Controller to the namespace (Example: http://{localhost:port}/api/braille), after that simply append the name of the method that will be used (Post, GetJobStatus, GetJobResult).
Please see the next part for more details about the individual API methods.
All controllers have the following generic structure:
http://{localhost:port}/api/{Controller Name}/{Method Name}


POST /api/Braille/Post

GET /api/Braille/GetJobStatus

GET /api/Braille/GetJobResult

4.Supported request methods

These methods are used in all Controllers.

4.1 Post

The Post request contains all the necessary parameters for starting a job in the API. The file content and all the necessary parameters must be placed in the body of the POST request and the POST content type is "multipart/form-data".
The request returns a unique **jobID** in the form of: “ca427b75-bb66-e511-91f0-1c6f65d84158”.
This id will be used in the GetJobStatus and GetJobResult in order to track and retrieve the job.

4.2 GetJobStatus

The GetJobStatus request is used in order to verify when the job is finished. It will return one of the following status codes:

Request url: http://{localhost:port}/api/{Controller Name}/GetJobStatus?jobId={your jobID}		
0 = error	1 = success	2 = processing

When programming against the Web API it is recommended to always check for the Job status before returning the result.

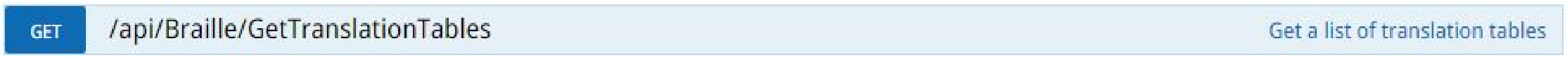
4.3 GetJobResult

The GetJobResult request will return the converted file. Make sure that the GetJobStatus request returns the value 1 before attempting to return the file result.

Request url: http://{localhost:port}/api/{Controller Name}/GetJobResult?jobId={your jobID}

4.4 Other request methods

Other request methods exist to aid the calling of the Post request. They provide relevant input parameters or optional parameters for creating the post request when needed.
For example the request method


GET /api/Braille/GetTranslationTables Get a list of translation tables

is used to get relevant translation tables for the optional input parameter “translationtable” in the post request.

5.Workflow for the user tests

In order to accurately test the Web API controllers, as in a production environment, the following workflow should be followed:

1. POST the job to the Controller under test
2. Save the jobID in your application/computer
3. Repeat: Call GetJobStatus with the saved jobID in predefined time intervals. Repeat until GetJobStatus value equals 1 or 0.
4. Depending on jobstatus from step 3.
    a. If jobstatus = 0 Notify the user that a conversion error has occured.
    b. If jobstatus = 1 then call the GetJobResult with the saved jobID, get the converted file and save it or provide it to the user.
5. Stop the workflow.

In the test cases the above mentioned workflow will define **1 controller request**. In order to achieve more complex conversions it may be necessary to have 2 or more controller requests. Multiple controller requests work by using the job result (the file) from the previous request as the input for the next one.

6.Valid set of input parameters for each controller

Please consult this table when going through the Test Cases.

Parameter Name	Valid Values	Notes
(AC) AccessibleConversion		
FileContent	The input document can be of any common document format. Please note that the accessible conversion is done through ABBYY and for a local implementation an ABBYY web service license needs to be bought and the web service needs to be setup.	
TargetDocumentFormat	Sensus supports the following output formats: "OFF_MSWord", "OFF_MSExcel", "OFF_RTF", "OFF_XML", "OFF_PDF", "OFF_PDFA", "OFF_Text", "OFF_CSV", "OFF_HTML", "OFF_NoConversion", "OFF_TIFF",	

	"OFF_JPG", "OFF_J2K", "OFF_InternalFormat", "OFF_DOCX", "OFF_XLSX", "OFF_JBIG2", "OFF_ALTO", "OFF_EPUB" Please check your local ABBYY web service configuration for supported formats.	
	(AU) Audio	
FileContent	a simple .txt file containing the text you want to be synthesized to speech	

Language	enUS =0x0409,
	enGB =0x0809	

SpeedOptions	Fastest = 6,
	Faster = 4,
	Fast = 2,
	Normal = 0,
	Slow = -2,
	Slower= -4,
	Slowest = -6	

FormatOptions	Mp3 =1,
	Wav =2,
	Wma =4,
	Aac =8	

	(BR) Braille	
FileContent	a .txt file	

BrailleFormat	sixdot =6,
	eightdot=8	

Language	enUEB =0x1,
	enGB =0x0809,
	enUS =0x0409,
	daDK =0x0406,
	nnNO =0x0814,
	isIS =0x040F,
	ptPT =0x0816,
	itIT =0x0410,
	frFR =0x040C,
	deDE =0x0407,
	roRO =0x0418,
	esES =0x0C0A,
	slSI =0x0424,
		Not all languages may work.

	huHU =0x040E,
	bgBG =0x0402,
	svSE =0x041D,
	elGR =0x0408,
	plPL = 0x0415	

Contraction	full =1,
	small =2,
	large =3,
	large2 =4,
	grade0 =5,
	grade1 =6,
	grade2b =7,
	grade2i =8,
	grade2 =9,
	level0 =10,
	level1 =11,
	level2 =12,
	level3 =13,
	user =14
		Not all contractions work for all languages. Specific contractions work with specific languages (Knowledge of braille contractions is recommended).

OutputFormat	Unicode=215,
	OctoBraille=216,
	Pef=217,
	noContractions = 1,
	compbrlAtCursor = 2,
	dotsIO = 4,
	comp8Dots = 8,
	pass1Only = 16,
	compbrlLeftCursor = 32,
	otherTrans = 64,
	ucBrl = 128,	

ConversionPath	texttobraille = 0,
	brailletotext = 1
		Default is texttobraille = 0

CharactersPerLine	number: 0 = no pagination and greater than (>) 10 for a valid pagination	Default is 0
LinesPerPage	number: 0 = no pagination and greater than (>) 2 for a valid pagination	Default is 0
TranslationTable	a valid translation table from the getTranslationTables API method	(Optional) Note: By using a typed translation table the braille controller will ignore the language and contraction parameters

PageNumbering	none = 0,
	right = 1,
		(Optional)
		Default is none = 0.

	left = 2	
(DA) Daisy		
FileContent	a .docx document that contains only simple text. NO mathematical equations or symbols, pictures.(TODO Edit here)	

DaisyOutput	TalkingBook = 1,
	Epub3WMO = 2	

(EB) EBook		
FileContent	a .pdf file	

format	mobi =1,
	epub =2	

(HT) HTMLtoPDF		
FileContent	a valid .html file with only a table	

size	a1 =1,
	a2 =2,
	a3 =3,
	a4 =4,
	a5 =5,
	a6 =6,
	a7 =7,
	a8 =8,
	a9 =9,
	a10 =10,
	letter =11	

(MS) MSOfficeConversion		
FileContent	Word: - .doc, .dot, .docx, .dotx, .docm, .dotm, .rtf, .odt, .txt, .htm, .html Excel - .xls, .xlsx, .xlsm, .xlt, .xltm, .xltx, .csv, .odc Powerpoint - .ppt, .pptx, .pptm, .pps, .ppsx, .ppsm, .odp Visio - .vsd, .vsdm, .vsdx, .svg [.vsdm, .vsdx & .svg require Visio >= 2013] Publisher - .pub Outlook - .msg, .vcf, .ics Project - .mpp [requires Project >= 2010]	Note: All of the input files can be converted to pdf. Only some files can be converted to .txt .html .rtf

MSOfficeOutput	pdf =1,
	txt=2,
	html=4,
	rtf=8	

(OC) OcrConversion		

FileContent	any common picture format .jpg , .jpeg , .tiff	Note: THE PICTURE MUST CONTAIN TEXT

language	enUS = 0x0409,
	daDK = 0x0406	

HTMLToText		
FileContent	a valid .html file	returns a .txt file

7.The test cases

Please check the Appendix: **A1-RoboBrailleWebApi.TestCases.xslx**

8.Other appendix files

The appendix filex A2 through A13 can be used as test files for the Test Cases. It is also encouraged to use your own files for a wider test coverage. The recommended files for each test case can be found in the A1 spreadsheet under the Input Parameters column in brackets. Example: (A4).