Package 'abbyyR'

May 30, 2018

Title Access to Abbyy Optical Character Recognition (OCR) API

Version 0.5.4

Maintainer Gaurav Sood <gsood07@gmail.com>

Description Get text from images of text using Abbyy Cloud Optical Character Recognition (OCR) API. Easily OCR images, barcodes, forms, documents with machine readable zones, e.g. passports. Get the results in a variety of formats including plain text and XML. To learn more about the Abbyy OCR API, see <http://ocrsdk.com/>.

URL http://github.com/soodoku/abbyyR

BugReports http://github.com/soodoku/abbyyR/issues

Depends R (>= 3.2.0)

License MIT + file LICENSE

LazyData true

VignetteBuilder knitr

Imports httr, XML, curl, readr, plyr, progress

Suggests testthat, rmarkdown, knitr (>= 1.11), lintr

RoxygenNote 6.0.1.9000

NeedsCompilation no

Author Gaurav Sood [aut, cre]

Repository CRAN

Date/Publication 2018-05-30 13:20:41 UTC

R topics documented:

---

abbyyR-package *abbyyR: R Client for the Abbyy Cloud OCR*

Description

OCR images, barcodes, forms, documents with machine readable zones, e.g. passports, right from R. Get the results in any one of a wide variety of formats, from text to XML. The package provides access to Abbyy Cloud OCR. For more information, see http://ocrsdk.com/. Details about results of calls to the API can be found at http://ocrsdk.com/documentation/specifications/status-codes/.

To learn how to use abbyyR, see this vignette: vignettes/Overview_of_abbyyR.html. Or, see how to scrape text from a folder of static Wisconsin Ads storyboards: vignettes/wiscads.html.

You need to get credentials (application ID and password) to use this application. If you haven't already, get these at http://ocrsdk.com/. And set these using setapp

Author(s)

Gaurav Sood

abbyy_check *Request Response Verification*

Description

Request Response Verification

Usage

```
abbyy_check(req)
```

Arguments

req	request

Value

in case of failure, a message

abbyy_GET *Base POST AND GET functions. Not exported.*

Description

GET

Usage

```
abbyy_GET(path, query, ...)
```

Arguments

path	path to specific API request URL
query	query list
...	Additional arguments passed to GET.

Value

list

abbyy_POST *POST*

Description

POST

Usage

```
abbyy_POST(path, query, body = "", ...)
```

Arguments

path	path to specific API request URL
query	query list
body	passing image through body
...	Additional arguments passed to POST.

Value

list

deleteTask *Delete Task*

Description

Deletes task and associated data. While Abbyy says, "If you try to delete the task that has already been deleted, the successful response is returned," it doesn't appear to hold. Hence, the function now defaults to checking the status of the task via getTaskStatus, and deletes only if it hasn't been deleted.

The function by default prints the status of the task you are trying to delete. It will show up as 'deleted' if successful.

Usage

```
deleteTask(taskId = NULL, ...)
```

Arguments

taskId	Required; ID of the task; String
...	Additional arguments passed to abbyy_GET.

Value

Data frame with all the details of the task you are trying to delete: id (task id), registrationTime, statusChangeTime, status (Submitted, Queued, InProgress, Completed, ProcessingFailed, Deleted, NotEnoughCredits), filesCount (No. of files), credits, resultUrl (URL for the processed file if applicable)

References

http://ocrsdk.com/documentation/apireference/deleteTask/

Examples

```
## Not run:
deleteTask(taskId = "task_id")

## End(Not run)
```

`getAppInfo` *Get Application Info*

Description

Get Information about the Application, such as number of pages (given the money), when the application credits expire etc. The function prints the details, and returns a data.frame with the details.

Usage

```
getAppInfo(...)
```

Arguments

...	Additional arguments passed to `abbyy_GET`.

Value

A data.frame with the following fields: Name of the Application (name), No. of pages remaining (pages), No. of fields remaining (fields), when the application credits expire (expires) and type of application (type).

References

http://ocrsdk.com/documentation/apireference/getApplicationInfo/

http://ocrsdk.com/schema/appInfo-1.0.xsd

Examples

```
## Not run:
getAppInfo()

## End(Not run)
```

`getResults` *Get Results*

Description

Get data from all the processed images.

Usage

```
getResults(output = "", save_to_file = TRUE)
```

Arguments

`output`	Optional; folder to which you want to save the data from the processed images; Default is same folder as the script
`save_to_file`	Required; Default is TRUE; If true, it saves to file. Otherwise returns a data frame with results + other attributes from Abbyy

Details

The function calls `listFinishedTasks`, goes through the finishedTasks data frame and downloads all the files in resultsUrl Results can be stored in memory or written to the hard disk. By default, the function writes to the disk. If the user wants the results to be written to disk, a data frame with paths to local file paths is returned. If the user wants to store the results in memory, data frame with a column carrying the results is returned.

Value

data frame returned by `listFinishedTasks` plus either a column that contains paths to local files (when writing to disk), or actual results returned.

References

http://ocrsdk.com/documentation/apireference/getTaskStatus/

Examples

```
## Not run:
getResults(save_to_file = FALSE)

## End(Not run)
```

getTaskStatus *Get Task Status*

Description

This function gets task status for a particular task ID. The function prints the status of the task by default. The function returns a data.frame with all the task details: id (task id), registrationTime, statusChangeTime, status (Submitted, Queued, InProgress, Completed, ProcessingFailed, Deleted, NotEnoughCredits), filesCount (No. of files), credits, resultUrl (URL for the processed file if applicable)

Usage

```
getTaskStatus(taskId = NULL, ...)
```

Arguments

taskId	Required, Id of the task
...	Additional arguments passed to abbyy_GET.

Value

A data.frame with all the available details about the task

References

http://ocrsdk.com/documentation/apireference/getTaskStatus/

Examples

```
## Not run:
getTaskStatus(taskId="task_id")

## End(Not run)
```

listFinishedTasks *List Finished Tasks*

Description

List all the finished tasks in the application.

The tasks are ordered by the time of the end of processing. No more than 100 tasks can be returned at one method call. The function prints number of finished tasks by default.

Usage

```
listFinishedTasks(...)
```

Arguments

...	Additional arguments passed to abbyy_GET.

Value

A data.frame with the following columns: id (task id), registrationTime, statusChangeTime, status (Completed), filesCount (No. of files), credits, resultUrl (URL for the processed file)

References

http://ocrsdk.com/documentation/apireference/listFinishedTasks/

Examples

```
## Not run:
listFinishedTasks()

## End(Not run)
```

---

listTasks *List Tasks*

Description

List all the tasks in the application. You can specify a date range and whether or not you want to include deleted tasks. The function prints total number of tasks and no. of finished tasks by default.

Usage

```
listTasks(fromDate=NULL, toDate=NULL, excludeDeleted = FALSE, ...)
```

Arguments

fromDate	Optional; format: yyyy-mm-ddThh:mm:ssZ
toDate	Optional; format: yyyy-mm-ddThh:mm:ssZ
excludeDeleted	Optional; Boolean, Default=FALSE
...	Additional arguments passed to abbyy_GET.

Value

A data.frame with the following columns: id (task id), registrationTime, statusChangeTime, status (Submitted, Queued, InProgress, Completed, ProcessingFailed, Deleted, NotEnoughCredits), filesCount (No. of files), credits, resultUrl (URL for the processed file)). If no tasks are finished, the last column (resultUrl) isn't returned.

References

http://ocrsdk.com/documentation/apireference/listTasks/

Examples

```
## Not run:
listTasks()
listTasks(fromDate = "2015-11-10T00:00:00Z", toDate = "2016-11-10T00:00:00Z")
listTasks(fromDate = "2015-11-10T00:00:00Z")

## End(Not run)
```

---

ocrFile *OCR File*

Description

Want to quick OCR a local file and get the results? Use this function.

Usage

```
ocrFile(file_path = "", output_dir = "./", exportFormat = c("txt",
  "txtUnstructured", "rtf", "docx", "xlsx", "pptx", "pdfSearchable",
  "pdfTextAndImages", "pdfa", "xml", "xmlForCorrectedImage", "alto"),
  save_to_file = TRUE)
```

Arguments

file_path	path to file containing OCR'd text; required
output_dir	path to output directory. file_name will be same as input file name (except for the extension)
exportFormat	optional, default: txt; options: txt, txtUnstructured, rtf, docx, xlsx, pptx, pdfSearchable, pdfTextAndImages, pdfa, xml, xmlForCorrectedImage, alto
save_to_file	Required, Boolean, Default is TRUE, but if not, returns result to memory

Value

path to output file

Examples

```
## Not run:
ocrFile(file_path = "path_to_ocr_file", output_dir = "path_to_output_dir")

## End(Not run)
```

processBarcodeField *Process Bar Code Field*

Description

Process the bar code field in an image.

Usage

```
processBarcodeField(file_path = "", barcodeType = "autodetect",
  region = "-1,-1,-1,-1", containsBinaryData = "false", pdfPassword = "",
  description = "", ...)
```

Arguments

file_path	path of the document
barcodeType	optional, default: "autodetect"
region	coordinates of region from top left, 4 values: top left bottom right; optional; default: "-1,-1,-1,-1" (entire image)
containsBinaryData	optional, default: "false"
pdfPassword	optional, default: ""
description	optional, default: ""
...	Additional arguments passed to abbyy_POST.

Value

Data frame with details of the task associated with the submitted Image

References

http://ocrsdk.com/documentation/apireference/processBarcodeField/

Examples

```
## Not run:
processBarcodeField(file_path = "file_path")

## End(Not run)
```

processBusinessCard *Process Business Card*

Description

Processes a Business Card

Usage

```
processBusinessCard(file_path = "", language = "English",
  imageSource = "auto", correctOrientation = "true", correctSkew = "true",
  exportFormat = "vCard", description = "", pdfPassword = "", ...)
```

Arguments

file_path	required, path of the document, default: ""
language	optional, default: English
imageSource	optional, default: auto
correctOrientation	optional, default: true
correctSkew	optional, default: true
exportFormat	optional, default: "vCard"
description	optional, default: ""
pdfPassword	optional, default: NULL
...	Additional arguments passed to abbyy_POST.

Value

Data frame with details of the task associated with the submitted Business Card

References

http://ocrsdk.com/documentation/apireference/processBusinessCard/

Examples

```
## Not run:
processBusinessCard(file_path="file_path", language="English")

## End(Not run)
```

processCheckmarkField *processCheckmarkField*

Description

Processes Checkmark Field

Usage

```
processCheckmarkField(file_path = "", checkmarkType = "empty",
  region = "-1,-1,-1,-1", correctionAllowed = "false", pdfPassword = "",
  description = "", ...)
```

Arguments

file_path	required, path of the document, default: ""
checkmarkType	optional, default: "empty"
region	coordinates of region from top left, 4 values: top left bottom right; optional; default: "-1,-1,-1,-1" (entire image)
correctionAllowed	optional, default: "false"
pdfPassword	optional, default: ""
description	optional, default: ""
...	Additional arguments passed to abbyy_POST.

Value

Data frame with details of the task associated with the submitted Image

References

http://ocrsdk.com/documentation/apireference/processCheckmarkField/

For supported image types, see http://ocrsdk.com/documentation/specifications/image-formats/

Examples

```
## Not run:
processCheckmarkField(file_path = "file_path")

## End(Not run)
```

processDocument *Process Document*

Description

This function processes several images for the same task and results in a multi-page document. For instance, upload pages of the book individually via submitImage to the same task. And then process it via ProcessDocument to get a multi-page pdf.

Usage

```
processDocument(taskId = NULL, language = "English",
  profile = c("documentConversion", "documentArchiving", "textExtraction",
  "fieldLevelRecognition", "barcodeRecognition"), textType = c("normal",
  "typewriter", "matrix", "index", "ocrA", "ocrB", "e13b", "cmc7", "gothic"),
  imageSource = c("auto", "photo", "scanner"),
  correctOrientation = c("true", "false"), correctSkew = c("true", "false"),
  readBarcodes = c("false", "true"), exportFormat = c("txt",
  "txtUnstructured", "rtf", "docx", "xlsx", "pptx", "pdfSearchable",
  "pdfTextAndImages", "pdfa", "xml", "xmlForCorrectedImage", "alto"),
  description = NULL, pdfPassword = NULL, ...)
```

Arguments

taskId	Only tasks with Submitted, Completed or NotEnoughCredits status can be processed using this function.
language	String. Optional; default: English
profile	String. Optional; default: documentConversion Options: documentConversion, documentArchiving,
textType	String. Optional; default: normal Options: normal, typewriter, matrix, index, ocrA,ocrB, e13b
imageSource	String. Optional; default: auto Options: auto, photo, scanner
correctOrientation	String. Optional; default: true. Options: true or false
correctSkew	String. Optional; default: true. Options: true or false
readBarcodes	Optional; Options: true or false
exportFormat	optional, default: txt options: txt, txtUnstructured, rtf, docx, xlsx, pptx, pdfSearchable, p
description	Optional; default: ""
pdfPassword	Optional; default: NULL
...	Additional arguments passed to abbyy_GET.

Value

data.frame with details of the task associated with the submitted Document

References

http://ocrsdk.com/documentation/apireference/processDocument/

Examples

```
## Not run:
processDocument(taskId = "task_id")

## End(Not run)
```

`processFields` *Process Fields*

Description

This function gets Information about a particular application

Usage

```
processFields(file_path = "", taskId = NULL, description = "", ...)
```

Arguments

`file_path`	path of the document
`taskId`	Only tasks with Submitted, Completed or NotEnoughCredits status can be processed using this function.
`description`	optional, default: ""
`...`	Additional arguments passed to `abbyy_POST`.

Value

`data.frame` with details of the task associated with the submitted Image

References

http://ocrsdk.com/documentation/apireference/processFields/

Examples

```
## Not run:
processFields(file_path = "file_path", taskId = "task_id", description = "")

## End(Not run)
```

processImage *Process Image*

Description

This function processes an image

Usage

```
processImage(file_path = "", language = "English",
  profile = c("documentConversion", "documentArchiving", "textExtraction",
  "fieldLevelRecognition", "barcodeRecognition"), textType = c("normal",
  "typewriter", "matrix", "index", "ocrA", "ocrB", "e13b", "cmc7", "gothic"),
  imageSource = c("auto", "photo", "scanner"),
  correctOrientation = c("true", "false"), correctSkew = c("true", "false"),
  region = "-1,-1,-1,-1", readBarcodes = c("false", "true"),
  exportFormat = c("txt", "txtUnstructured", "rtf", "docx", "xlsx", "pptx",
  "pdfSearchable", "pdfTextAndImages", "pdfa", "xml", "xmlForCorrectedImage",
  "alto"), description = "", pdfPassword = "", ...)
```

Arguments

file_path	path to the document
language	optional, default: English
profile	String. Optional; default: documentConversion Options: documentConversion, documentArchiving,
textType	String. Optional; default: normal Options: normal, typewriter, matrix, index, ocrA,ocrB, e13b
imageSource	String. Optional; default: auto Options: auto, photo, scanner
correctOrientation	String. Optional; default: true. Options: true or false
correctSkew	String. Optional; default: true. Options: true or false
region	String. Optional. Default: "-1,-1,-1,-1". Region of the image.
readBarcodes	Optional; Options: true or false
exportFormat	optional, default: txt options: txt, txtUnstructured, rtf, docx, xlsx, pptx, pdfSearchable, p
description	optional, default: ""
pdfPassword	optional, default: NULL
...	Additional arguments passed to abbyy_POST.

Value

A data.frame with details of the task associated with the submitted Image

References

http://ocrsdk.com/documentation/specifications/image-formats/

http://ocrsdk.com/documentation/apireference/processImage/

Examples

```
## Not run:
processImage(file_path = "file_path", language = "English", exportFormat = "txtUnstructured")

## End(Not run)
```

`processMRZ` *Process MRZ: Extract data from Machine Readable Zone*

Description

Extract data from Machine Readable Zone in an Image

Usage

```
processMRZ(file_path = "", ...)
```

Arguments

`file_path`	path to the document
`...`	Additional arguments passed to `abbyy_POST`.

Value

Data frame with details of the task associated with the submitted MRZ document

References

http://ocrsdk.com/documentation/apireference/processMRZ/

Examples

```
## Not run:
processMRZ(file_path = "file_path")

## End(Not run)
```

processPhotoId *Process Photo ID*

Description

Get data from a Photo ID. The function is under testing and may not work fully.

Usage

```
processPhotoId(file_path = "", idType = "auto", imageSource = "auto",
  correctOrientation = "true", correctSkew = "true", description = "",
  pdfPassword = "", ...)
```

Arguments

file_path	path to file; required
idType	optional; default = "auto"
imageSource	optional; default = "auto"
correctOrientation	String. Optional; default: true. Options: true or false
correctSkew	String. Optional; default: true. Options: true or false
description	optional; default = ""
pdfPassword	optional; default = ""
...	Additional arguments passed to abbyy_POST.

Value

Data frame with details of the task associated with the submitted Photo ID image

References

http://ocrsdk.com/documentation/apireference/processPhotoId/

Examples

```
## Not run:
processPhotoId(file_path = "file_path", idType = "auto", imageSource = "auto")

## End(Not run)
```

`processRemoteImage` *Process Remote Image*

Description

This function gets Information about a particular application

Usage

```
processRemoteImage(img_url = NULL, language = "English",
  profile = c("documentConversion", "documentArchiving", "textExtraction",
  "fieldLevelRecognition", "barcodeRecognition"), textType = c("normal",
  "typewriter", "matrix", "index", "ocrA", "ocrB", "e13b", "cmc7", "gothic"),
  imageSource = c("auto", "photo", "scanner"),
  correctOrientation = c("true", "false"), correctSkew = c("true", "false"),
  readBarcodes = c("false", "true"), exportFormat = c("txt",
  "txtUnstructured", "rtf", "docx", "xlsx", "pptx", "pdfSearchable",
  "pdfTextAndImages", "pdfa", "xml", "xmlForCorrectedImage", "alto"),
  description = NULL, pdfPassword = NULL, ...)
```

Arguments

`img_url`	Required; url to remote image
`language`	Optional; default: English
`profile`	String. Optional; default: `documentConversion` Options: `documentConversion, documentArchiving,`
`textType`	String. Optional; default: `normal`. Specifies the type of the text on a page. Options: `normal, typewriter, matrix, index, ocrA,ocrB, e13b,cmc7, gothic`
`imageSource`	String. Optional; default: `auto` Options: `auto, photo, scanner`
`correctOrientation`	String. Optional; default: `true`. Options: `true` or `false`
`correctSkew`	String. Optional; default: `true`. Options: `true` or `false`
`readBarcodes`	Optional; default:
`exportFormat`	Optional; default: txt. Must be one of the following: txt, rtf, docx, xlsx, pptx, pdfSearchable, pdfTextAndImages, pdfa, xml, alto
`description`	Optional; default: ""
`pdfPassword`	Optional; default: NULL
`...`	Additional arguments passed to `abbyy_GET`.

Value

Data frame with details of the task associated with the submitted Remote Image

References

http://ocrsdk.com/documentation/apireference/processRemoteImage/

Examples

```
## Not run:
processRemoteImage(img_url = "img_url")

## End(Not run)
```

`processTextField` *Process Text Field*

Description

This function gets Information about a particular application

Usage

```
processTextField(file_path = "", region = "-1,-1,-1,-1",
  language = "English", letterSet = "", regExp = "",
  textType = "normal", oneTextLine = "false",
  oneWordPerTextLine = "false", markingType = "simpleText",
  placeholdersCount = "1", writingStyle = "default", description = "",
  pdfPassword = "", ...)
```

Arguments

`file_path`	path of the document
`region`	coordinates of region from top left, 4 values: top left bottom right; optional; default: "-1,-1,-1,-1" (entire image)
`language`	optional; default: "English"
`letterSet`	letterset to be used for recognition, set by language but can be customized; optional; default: ""
`regExp`	which words are allowed in the field. see regular expression documentation; optional; default: ""
`textType`	type of the text in the field including typewriter, handprinted; optional; default: "normal"
`oneTextLine`	field contains only one text line or more; optional; default: "false"
`oneWordPerTextLine`	field contains one word per line or not; optional; default: "false"
`markingType`	only for handprint recognition, includes underlinedText etc.; optional; default: "simpleText"
`placeholdersCount`	No. of character cells for the field; optional; default: "1"
`writingStyle`	handprint writing style, see Abbyy FineReader documentation for values; optional; default: "default"
`description`	Description of processing task; optional; default: ""
`pdfPassword`	Password for pdf; optional; default: ""
`...`	Additional arguments passed to `abbyy_POST`.

Value

Data frame with details of the task associated with the submitted Image

References

http://ocrsdk.com/documentation/apireference/processTextField/

http://ocrsdk.com/documentation/specifications/regular-expressions/

Examples

```
## Not run:
processTextField(file_path="file_path")

## End(Not run)
```

---

setapp *Sets Application ID and Password*

Description

Set Application ID and Password. Needed for interfacing with Abbyy FineReader Cloud OCR SDK. Run this before anything else.

Usage

```
setapp(appdetails = NULL, force = FALSE)
```

Arguments

appdetails	A vector of app_id, app_password. Get these from http://ocrsdk.com/. Set them before you use other functions.
force	Force change the app_id and app_password stored in the environment

Details

The function looks for AbbyyAppId and AbbyyAppPassword in the environment. If it doesn't find them and if we don't want to force change in them, it looks for arguments. And if no arguments are passed, it asks for user to input the values.

References

http://ocrsdk.com/documentation/apireference/getApplicationInfo/

Examples

```
## Not run:
setapp(c("app_id", "app_password"))

## End(Not run)
```

submitImage *Submit Image*

Description

Adds image to the existing task or creates a new task for the uploaded image. The new task isn't processed till processDocument or processFields is called.

Usage

```
submitImage(file_path = "", taskId = "", pdfPassword = "", ...)
```

Arguments

file_path	Required; Path to the document
taskId	Optional; Assigns image to the task ID specified. If an empty string is passed, a new task is created.
pdfPassword	Optional; If the pdf is password protected, put the password here.
...	Additional arguments passed to abbyy_POST.

Value

Data frame with all the details of the submitted image: id (task id), registrationTime, statusChangeTime, status (Submitted, Queued, InProgress, Completed, ProcessingFailed, Deleted, NotEnoughCredits), filesCount (No. of files), credits

References

http://ocrsdk.com/documentation/apireference/submitImage/

Examples

```
## Not run:
submitImage(file_path="/images/image1.png", taskId="task_id", pdfPassword="pdf_password")

## End(Not run)
```

Index