

# Package: rmistral (via r-universe)

May 13, 2026

**Type** Package

**Title** Experimental R interface to Mistral AI API

**Version** 0.1.0

**Author** Sebastian Kranz

**Maintainer** Sebastian Kranz <sebastian.kranz@uni-ulm.de>

**Description** Currently implement only OCR capabilities for PDF conversion.

**License** GPL >= 2.0

**Encoding** UTF-8

**LazyData** true

**Depends** httr, stringi, restorepoint, base64enc

**RoxygenNote** 7.2.3

**Config/pak/sysreqs** libicu-dev libssl-dev

**Repository** <https://repboxr.r-universe.dev>

**Date/Publication** 2026-03-27 06:16:39 UTC

**RemoteUrl** <https://github.com/skranz/rmistral>

**RemoteRef** main

**RemoteSha** fc800312b80367d3460cb623bc658e9987cd919f

## Contents

mistral_api_key . . . . .	2
mistral_get_file_url . . . . .	2
mistral_ocr . . . . .	3
mistral_ocr_save_images . . . . .	3
mistral_ocr_save_md . . . . .	4
mistral_upload_file . . . . .	5
set_mistral_api_key . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

<code>mistral_api_key</code>	<i>Retrieve the Mistral API key</i>
------------------------------	-------------------------------------

---

**Description**

This function retrieves the Mistral API key stored in the R options.

**Usage**

```
mistral_api_key()
```

**Value**

The stored API key as a string.

---

<code>mistral_get_file_url</code>	<i>Get a signed URL for file access</i>
-----------------------------------	---

---

**Description**

Get a signed URL for file access

**Usage**

```
mistral_get_file_url(file_id, expiry = 24, api_key = mistral_api_key())
```

**Arguments**

<code>file_id</code>	ID of the file to get URL for
<code>expiry</code>	Expiry time in hours, default is 24
<code>api_key</code>	Your MistralAI API key

**Value**

The signed URL information as a list

---

mistral_ocr	<i>Perform OCR using the Mistral API</i>
-------------	--

---

### Description

This function sends a document (either via URL or file upload) to the Mistral OCR API for text extraction.

### Usage

```
mistral_ocr(
    url = NULL,
    file = NULL,
    model = "mistral-ocr-latest",
    include_images = TRUE,
    timeout_sec = 60 * 5,
    api_key = mistral_api_key()
)
```

### Arguments

url	Optional. The URL of the document to be processed.
file	Optional. A local file path or an uploaded file object.
model	The OCR model to use. Defaults to "mistral-ocr-latest".
include_images	Logical. Whether to include images in the response. Defaults to 'TRUE'.
timeout_sec	The timeout for the API request in seconds. Defaults to 300 (5 minutes).
api_key	Character string. Your Mistral API key. Best set globally via calling <code>set_mistral_api_key</code> .

### Value

List containing the result. The field `is_ok` should be TRUE if everything worked nicely. The element `pages` contains the extracted pages.

---

mistral_ocr_save_images	<i>Save extracted images from OCR results</i>
-------------------------	---

---

### Description

This function extracts and saves images from OCR results to a specified directory.

### Usage

```
mistral_ocr_save_images(ocr, img_dir, overwrite = FALSE)
```

**Arguments**

ocr	A list containing OCR results, including extracted images.
img_dir	Directory where images should be saved.
overwrite	Logical. If 'TRUE', overwrites existing image files. Defaults to 'FALSE'.

**Value**

The number of images saved.

---

mistral\_ocr\_save\_md     *Save OCR results as a Markdown file*

---

**Description**

This function saves the extracted OCR text in Markdown format, either as a single file or split by pages.

**Usage**

```
mistral_ocr_save_md(
  ocr,
  file,
  by_page = FALSE,
  save_images = TRUE,
  overwrite = FALSE,
  img_dir = dirname(file)
)
```

**Arguments**

ocr	A list containing OCR results, including extracted text and images.
file	The file path where the Markdown content should be saved.
by_page	Logical. If 'TRUE', saves each page as a separate file. Defaults to 'FALSE'.
save_images	Logical. If 'TRUE', also saves extracted images. Defaults to 'TRUE'.
overwrite	Logical. If 'TRUE', overwrites existing files. Defaults to 'FALSE'.
img_dir	Directory where images should be saved. Defaults to the directory of 'file'.

---

`mistral_upload_file`     *Upload a file for OCR processing*

---

**Description**

Upload a file for OCR processing

**Usage**

```
mistral_upload_file(file_path, purpose = "ocr", api_key = mistral_api_key())
```

**Arguments**

<code>file_path</code>	Path to the local file to upload
<code>purpose</code>	Purpose of the upload, default is "ocr"
<code>api_key</code>	Your MistralAI API key

**Value**

The file metadata as a list

---

`set_mistral_api_key`     *Set the Mistral API key*

---

**Description**

This function sets the Mistral API key, either from a provided string or by reading from a file.

**Usage**

```
set_mistral_api_key(key = NULL, file = NULL)
```

**Arguments**

<code>key</code>	Optional. The API key as a string.
<code>file</code>	Optional. A file path containing the API key.

# Index

mistral\_api\_key, 2  
mistral\_get\_file\_url, 2  
mistral\_ocr, 3  
mistral\_ocr\_save\_images, 3  
mistral\_ocr\_save\_md, 4  
mistral\_upload\_file, 5  
set\_mistral\_api\_key, 5