

# Package: duckduckr (via r-universe)

August 25, 2024

**Type** Package

**Title** Simple Client for the DuckDuckGo Instant Answer API

**Version** 1.0.0.9000

**Description** Programmatic access to the DuckDuckGo Instant Answer API  
<<https://api.duckduckgo.com/api>>.

**License** MIT + file LICENSE

**URL** <https://github.com/dirkschumacher/duckduckr>

**BugReports** <https://github.com/dirkschumacher/duckduckr/issues>

**Encoding** UTF-8

**LazyData** true

**Imports** crul, jsonlite

**Suggests** testthat, covr

**RoxygenNote** 6.0.1

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

**RemoteUrl** <https://github.com/dirkschumacher/duckduckr>

**RemoteRef** HEAD

**RemoteSha** 59f6589c740ed498c9ac79a3650b29fe57bd36be

## Contents

duckduckr . . . . .	2
duckduck_answer . . . . .	2

<b>Index</b>	<b>4</b>
--------------	----------

---

`duckduckr`*Simple Client for the DuckDuckGo Instant Answer API*

---

**Description**

Programmatic access to the DuckDuckGo's Instant Answer API <<https://api.duckduckgo.com/api>>

**Author(s)**

Dirk Schumacher ([mail@dirk-schumacher.net](mailto:mail@dirk-schumacher.net))

---

`duckduck_answer`*Call DuckDuckGo Instant Answer API*

---

**Description**

Makes a synchronous API call to the DuckDuckGo Instant Answer API. Take a look at DuckDuckGo's terms of use (<https://api.duckduckgo.com/api>) before using it.

**Usage**

```
duckduck_answer(query, no_redirect = FALSE, no_html = FALSE,
  skip_disambig = FALSE, app_name = "duckduckr")
```

**Arguments**

<code>query</code>	the query string
<code>no_redirect</code>	TRUE to skip HTTP redirects (for !bang commands)
<code>no_html</code>	TRUE to remove html from results
<code>skip_disambig</code>	TRUE to to skip disambiguation (D) Type.
<code>app_name</code>	the appname used to identify your application.

**Value**

Always returns a list. If the API call was successful it contains the response of the duckduckgo API as parsed by `fromJSON`. In addition the object's attributes contain additional meta data. Especially the `status` attribute indicates if something went wrong during the HTTP call or parsing of the JSON text.

In case the call was successful the `status` attribute is equal to "OK".

In case something went wrong, the `status` attribute is equal to "error" and in the `error` attribute you will find more information. In particular the `type`, which is either "http\_error" or "json\_parse\_error" depending on the error's source.

In case of a "http\_error", there is an additional message and `http_status` element.

In case of "json\_parse\_error", there is an additional message element.

In addition there is always a `source` element with the URL used to query the data.

**See Also**

<https://api.duckduckgo.com/api> for more information on the API and their terms of use.

**Examples**

```
## Not run:  
tmp <- duckduck_answer("duckduckgo")  
tmp$Abstract  
  
## End(Not run)
```

# Index

[duckduck\\_answer](#), 2  
[duckduckr](#), 2  
[duckduckr-package \(duckduckr\)](#), 2  
[fromJSON](#), 2