

API:Page info i API Discussion results



This page is part of the [Web APIs r](#) Read Edit View history



Use the MediaWiki API to provide more context searching for Wikipedia pages.

[Main page](#)
[Get MediaWiki](#)
[Get extensions](#)
[Tech blog](#)
[Contribute](#)

[Support](#)
[User help](#)
[FAQ](#)
[Technical manual](#)
[Support desk](#)
[Communication](#)

[Development](#)
[Bug tracker](#)
[Code repository](#)
[Code docs](#)
[Statistics](#)
[Wikimedia technology](#)
[Wikimedia audiences](#)

[MediaWiki.org](#)
[Community portal](#)
[Recent changes](#)
[Translate content](#)
[Random page](#)
[Current issues](#)
[Sandbox](#)

[Print/export](#)
[Create a book](#)
[Download as PDF](#)
[Printable version](#)

[Tools](#)
[What links here](#)
[Related changes](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Cite this page](#)

[In other languages](#)
[Add links](#)

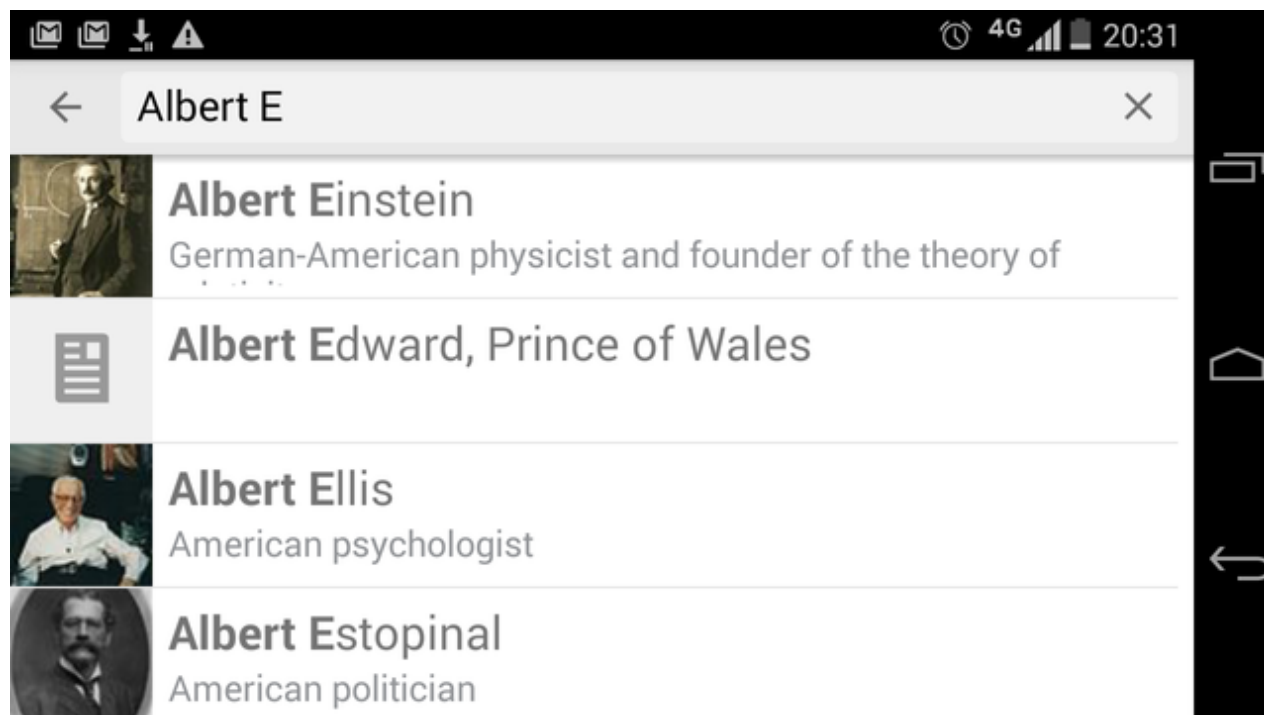
Contents [hide]

- [1 Introduction](#)
- [2 Showing useful page information](#)
- [3 How it works on Wikimedia wikis](#)
 - [3.1 Querying query results in one request](#)
 - [3.2 Further niceties](#)
- [4 Going beyond](#)
- [5 Alternatives](#)
- [6 Next steps](#)
- [7 See also](#)

Introduction [edit]

This shows the API results that search results use to display additional information about articles, including a lead image and a description of the article's subject from Wikidata.

Showing useful page information [edit]



When you search in the Wikipedia mobile apps, as you type they show a drop-down list of matching pages. They also show the lead image for the article and a description of it.

The lead image comes from [Extension:PageImages](#), which adds a `page_image` property to pages giving its guess as to an appropriate image for the page. The description comes from Wikidata, which maintains a localized description of the subject of each wiki page.

A slow way to do this would be to query for pages matching what the user types, then make an API `action=query` request for the `pageimages` property of the set of titles, and make another API query to `wikidata.org` requesting the Wikidata description. This works but involves multiple API queries.

How it works on Wikimedia wikis [\[edit \]](#)

Instead, WMF changed most wikis (but as of May 2015, not `www.mediawiki.org`) to load the [Wikibase client extension](#) for accessing Wikidata. This allows you to query for `prop=pageterms` along with `prop=pageimages` on the source wiki, instead of making a second request to `www.wikidata.org`.

Example: Simple query for `pageimages` and `pageterms` properties of [Albert Einstein](#):

`api.php ? action=query & formatversion=2 & prop=pageimages|pageterms & titles=Albert%20Einstein` [\[try in ApiSandbox\]](#)

Result	[Expand]
--------	--------------------------



Note: You should always use `formatversion=2` in API requests if you are requesting results in JSON format. It returns results in a structure that is easier to process, and uses UTF8 encoding by default.}}

If you have a set of page titles, you can request their information all at once. Set `pilimit` to the number of titles you're querying, otherwise it will only return one thumbnail, from the first article in the set that has a plausible image. Also you should reduce the API response size by specifying only the properties you want the API modules to supply, in this case only the the thumbnail and Wikidata description. Finally, you may want the query to handle pages that are redirects.

Example: Query for `pageterms` and `pageimage` thumbnails of several pages

`api.php ? action=query & formatversion=2 & prop=pageimages|pageterms & titles=Albert%20Einstein|Albert%20Ellis|Albert%20Estopinal & pilimit=3 & piprop=thumbnail & wbpptterms=description & redirects=` [\[try in ApiSandbox\]](#)

Result	[Expand]
--------	--------------------------

Querying query results in one request [\[edit \]](#)

The above example is incomplete, since the set of page titles whose properties we're querying – `Albert Einstein|Albert Ellis|Albert Estopinal` – must have come from another query.

In many situations you can combine getting the page properties you want with the initial query for a set of pages, using the MediaWiki API's [generator](#) feature. The list of pages from the generator become the set of pages for the other part of the query, all in a single API request. The MediaWiki API's query module has a `prefixsearch` submodule that queries for a list of pages starting with the prefix you specify ("`Albert Ei`"), and list queries can act as a generator. The [MobileFrontend extension](#) and mobile apps do this, If you look at MobileFrontend's API query in [SearchApi.js](#), you

can see it combines the `generator=prefixsearch` with a query for the `pageimages` property. We can do the same, asking for the Wikidata description as well with `prop=pageimages|pageterms`.

Example: A query feeding a prefixsearch generator into useful property queries.

api.php ? action=query & formatversion=2 & generator=prefixsearch & gpssearch=Albert%20Ei&gpslimit=10 & prop=pageimages|pageterms & piprop=thumbnail&pithumbsize=50&pilimit=10 & redirects= & wbptterms=description [try in ApiSandbox]

The `prefixsearch` generator provides an `index` for each page in the `pages` array in the result; you can use this to sort the page titles, each with its thumbnail image and description, in the correct order.



Note: You used to have to append a separate `list=prefixsearch` query to get the titles in the correct order (phab:T98125).

Further niceties [\[edit \]](#)

If the set of articles that start with what the user types does not fill the search results list, the Wikimedia mobile apps go on to search for in-page matches that you would get from [Special:Search](#). The Wikipedia Android and iOS mobile apps combines `generator=prefixsearch` with querying for the `pageterms` and `pageimages` properties *and* getting a list of search terms. From its implementation file ,

```
@ "action": @ "query",
@ "generator": @ "prefixsearch",
@ "gpssearch": self.searchTerm,
@ "gpsnamespace": @ 0,
@ "gpslimit": @ (SEARCH_MAX_RESULTS),
@ "prop": @ "pageterms|pageimages",
@ "piprop": @ "thumbnail",
@ "wbptterms": @ "description",
@ "pithumbsize" : @ (SEARCH_THUMBNAIL_WIDTH),
@ "pilimit": @ (SEARCH_MAX_RESULTS),
// -- Parameters causing prefix search to efficiently return
suggestion.
@ "list": @ "search",
@ "srsearch": self.searchTerm,
@ "srnamespace": @ 0,
@ "srwhat": @ "text",
@ "srinfo": @ "suggestion",
@ "srprop": @ "",
@ "sroffset": @ 0,
@ "srlimit": @ 1,
```

Going beyond [\[edit \]](#)

The Wikipedia iOS mobile app uses face detection to detect the focal region of the image!

Alternatives [\[edit \]](#)

As an alternative, the [Popups](#) extension behind the [Hovercards beta feature](#) uses the `extracts` query submodule of the [TextExtracts extension](#) to show two sentences from the lead text of an article from the local wiki (together with its image, "last edited", etc.) when you hover over a link. This text from the local wiki is usually longer and less definitive than the Wikidata description. Its API request is in [resources/ext.popups.renderer.article.js](#)

Example: Sample Hovercards API request including textextracts properties of [Albert Einstein](#):

`api.php ? action=query & formatversion=2 & prop=extractslpageimageslrevisions & titles=Albert%20Einstein & redirects= & exintro=true&exsentences=2&explaintext=true & piprop=thumbnail&pithumbsize=300 & rvprop=timestamp` [\[try in ApiSandbox\]](#)

Result	[Expand]
--------	--------------------------

The query returns an array of pages, if successful this will have one element, the single matching page. This also requests the last-changed timestamp (``prop=revisions&rvprop=timestamp``) to display "Edited *N* days/hours ago."

Next steps [\[edit \]](#)

Try these API requests in the [Special:ApiSandbox](#) page, then make the same API requests from your own applications.

See also [\[edit \]](#)

- [Introducing lead images to Wikipedia's Android beta app](#) – a blog post on lead images

Category: [Web APIs hub](#)

This page was last edited on 21 September 2017, at 11:39.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About MediaWiki.org](#) [Disclaimers](#) [Code of Conduct](#) [Developers](#) [Cookie statement](#) [Mobile view](#)

