**API keys**

To authenticate a user to Polr, you *must* provide an API key along with each request to Polr API endpoints, as a GET or POST parameter. (e.g ?key=API\_KEY\_HERE)

**Assigning an API key**

To assign an API key, log on from an administrator account, head over to the "Admin" tab, and scroll to the desired user. From there, you can open the API button dropdown to reset, create, or delete the user's API key. You will also be prompted to set a desired API quota. This is defined as requests per minute. You may allow unlimited requests by making the quota negative. Once the user receives an API key, they will be able to see an "API" tab in their user panel, which provides the information necessary to interact with the API.

Alternative method: You can also assign a user an API key by editing their entry in the users database table, editing the api\_key value to the desired API key, api\_active to the correct value (1 for active, 0 for inactive), and api\_quota to the desired API quota (see above).

**Actions**

Actions are passed as a segment in the URL. There are currently two actions implemented:

* shorten - shortens a URL
* lookup - looks up the destination of a shortened URL

Actions take arguments, which are passed as GET or POST parameters. See [API endpoints](https://docs.polrproject.org/en/latest/developer-guide/api/#api-endpoints) for more information on the actions.

**Response Type**

The Polr API will reply in plain\_text or json. The response type can be set by providing the response\_type argument to the request. If this argument is not provided, the response type will default to either plain\_text or json depending on the endpoint.

Example json responses:

{

"action": "shorten",

"result": "https://example.com/5kq"

}

{

"action":"lookup",

"result": {

"long\_url": "https:\/\/google.com",

"created\_at": {

"date":"2016-02-12 15:20:34.000000",

"timezone\_type":3,

"timezone":"UTC"

},

"clicks":"0"

}

}

Example plain\_text responses:

https://example.com/5kq

https://google.com

**API Endpoints**

All API calls will commence with the base URL, /api/v2/.

/api/v2/action/shorten

Arguments:

* url: the URL to shorten (e.g https://google.com)
* is\_secret (optional): whether the URL should be a secret URL or not. Defaults to false. (e.g true or false)
* custom\_ending (optional): a custom ending for the short URL. If left empty, no custom ending will be assigned.

Response: A JSON or plain text representation of the shortened URL.

Example: GET http://example.com/api/v2/action/shorten?key=API\_KEY\_HERE&url=https://google.com&custom\_ending=CUSTOM\_ENDING&is\_secret=false

Response:

{

"action": "shorten",

"result": "https://example.com/5kq"

}

The url argument must be URL encoded.

/api/v2/action/shorten\_bulk

*POST only*

Arguments:

* data: a string containing a JSON-encoded object with an array of links

Example data argument:

{

"links": [

{

"url": "https://polrproject.org/"

},

{

"url": "https://youtube.com",

"is\_secret": true

},

{

"url": "https://github.com/cydrobolt/polr",

"custom\_ending": "polrgithub"

}

]

}

Response: A JSON-encoded object with a list of shortened links.

Example response:

{

"action": "shorten\_bulk",

"result": {

"shortened\_links": [

{

"long\_url": "https://polrproject.org/",

"short\_url": "http://demo.polr.me/81"

},

{

"long\_url": "https://youtube.com",

"short\_url": "http://demo.polr.me/84/b496"

},

{

"long\_url": "https://github.com/cydrobolt/polr",

"short\_url": "http://demo.polr.me/polrgithub"

}

]

}

}

/api/v2/action/lookup

The lookup action takes a single argument: url\_ending. This is the URL to lookup. If it exists, the API will return with the destination of that URL. If it does not exist, the API will return with the status code 404 (Not Found).

Arguments:

* url\_ending: the link ending for the URL to look up. (e.g 5ga)
* url\_key (optional): optional URL ending key for lookups against secret URLs

Remember that the url argument must be URL encoded.

Example: GET http://example.com/api/v2/action/lookup?key=API\_KEY\_HERE&url\_ending=2

Response:

{

"action": "lookup",

"result": "https://google.com"

}

/api/v2/data/link

Arguments:

* url\_ending: the link ending for the URL to look up. (e.g 5ga)
* left\_bound: left date bound (e.g 2017-02-28 22:41:43)
* right\_bound: right date bound (e.g 2017-03-13 22:41:43)
* stats\_type: the type of data to fetch
  + day: click counts for each day from left\_bound to right\_bound
  + country: click counts per country
  + referer: click counts per referer

The dates must be formatted for the strtotime PHP function and must be parsable by Carbon. By default, this API endpoint will only allow users to fetch a maximum of 365 days of data. This setting can be modified in the .env configuration file.

An API key granted to a regular user can only fetch data for their own links. Admins can fetch data for any link.

Response: A JSON representation of the requested analytics data.

Example: GET http://example.com/api/v2/data/link?stats\_type=day&key=API\_KEY\_HERE&url\_ending=5gk&response\_type=json&left\_bound=2017-02-28%2022:41:43&right\_bound=2017-03-13%2022:21:43

Response:

{

"action":"data\_link\_day",

"result": {

"url\_ending":"5gk",

"data": [

{"x":"2017-03-10","y":42},

{"x":"2017-03-11","y":1},

{"x":"2017-03-12","y":5}

]

}

}

Example: GET http://example.com/api/v2/data/link?stats\_type=country&key=API\_KEY\_HERE&url\_ending=5gk&response\_type=json&left\_bound=2017-02-28%2022:41:43&right\_bound=2017-03-13%2022:21:43

Response:

{

"action":"data\_link\_day",

"result": {

"url\_ending":"5gk",

"data": [

{"label":"FR","clicks":1},

{"label":"US","clicks":6},

{"label":"CA","clicks":41}

]

}

}

Example: GET http://example.com/api/v2/data/link?stats\_type=country&key=API\_KEY\_HERE&url\_ending=5gk&response\_type=json&left\_bound=2017-02-28%2022:41:43&right\_bound=2017-03-13%2022:21:43

Response:

{

"action":"data\_link\_day",

"result": {

"url\_ending":"5gk",

"data": [

{"label":"Direct","clicks":6},

{"label":"reddit.com","clicks":12},

{"label":"facebook.com","clicks":30}

]

}

}

**HTTP Error Codes**

The API will return an error code if your request was malformed or another error occured while processing your request.

HTTP 400 Bad Request

This status code is returned in the following circumstances:

* By the shorten endpoint
  + In the event that the custom ending provided is already in use, a 400 error code will be returned and the message custom ending already in use will be returned as an error.
* By any endpoint
  + Your request will return a 400 if it is malformed or the contents of your arguments do not fit the required data type.

HTTP 500 Internal Server Error

* By any endpoint
  + The server has encountered an unhandled error. This is most likely due to a problem with your configuration or your server is unable to handle the request due to a bug.

HTTP 401 Unauthorized

* By any endpoint
  + You are unauthorized to make the transaction. This is most likely due to an API token mismatch, or your API token has not be set to active.
* By the lookup endpoint
  + You have not provided the valid url\_key for a secret URL lookup.

HTTP 404 Not Found

* By the lookup endpoint
  + Returned in the circumstance that the short URL to look up was not found in the database.

HTTP 403 Forbidden

* By the shorten endpoint
  + Your request was understood, but you have exceeded your quota.

**Error Responses**

Example json error response:

{

"status\_code":429,

"error\_code":"QUOTA\_EXCEEDED",

"error":"Quota exceeded."

}

Example plain\_text error response:

429 Quota exceeded.