
shorten-url

Release 1.0

Alexander

Mar 14, 2024

CONTENTS

1	Contents	3
1.1	Data Dictionary	3
1.2	API (version 1)	3

shorten-url is a professional URL shortener service with lot of features, link editing, detailed click statistics, traffic filters, redirect rules, beautiful QR codes, custom domains, password protected links, API access and much more...

Website: <https://shorten-url.com>

CONTENTS

1.1 Data Dictionary

1.1.1 Group status

Possible values for group status:

Status	Description
1	Group enabled
2	Group disabled

1.1.2 Link status

Possible values for short link status:

Status	Description
1	Link enabled
2	Link disabled
3	Link expired (cannot be set via API)
4	Link inactive (cannot be set via API)

1.2 API (version 1)

Base URL: `https://my.shorten-url.com/api/v1`

API uses Bearer authorisation tokens to authorise the user. In order to use the API you will need to generate your API key and use it as a token. This can be done in your account, API Keys section.

API uses GET, POST, PATCH and DELETE methods. If you are unable to use PATCH and DELETE methods due to network restrictions then send request using POST method and add `_method` field with required method name to root of each request.

Postman desktop application json collection with samples of all API requests is available for download [here](#). Download, unzip and import to Postman. You will need to generate your own API key and set it as a Bearer authorisation token on the collection level. Some of the requests uses collection level variables for certain object (group, link, etc) IDs. Replace them with your own object IDs.

1.2.1 System API

System API contains requests to do a system related functions e.g. ping API host and retrieve miscellaneous data dictionaries.

system: echo

System echo request. Returns API key user details. Can be used to test the connection and API key setup.

Sub-URL: /system/echo

Method: GET

system: link-types

Get supported short link types e.g. link to a web page, phone number, email address, social profile, etc. Link type ID should be used to create new short links. Variable {data} in link type format will be replaced with actual user data when creating a new link.

Sub-URL: /system/link-types

Method: GET

system: domains

Get available public and user custom domains for short URLs. Variable {key} in domain format will be replaced with generated short URL key when creating a new link.

Sub-URL: /system/domains

Method: GET

system: countries

Get available countries and their codes. Country codes can be used in redirect rules and filters.

Sub-URL: /system/countries

Method: GET

system: languages

Get available browser languages and their codes. Language codes can be used in redirect rules and filters.

Sub-URL: /system/languages

Method: GET

system: os-families

Get available operating system names. Can be used in redirect rules and filters.

Sub-URL: `/system/os-families`

Method: GET

system: browser-families

Get available browser names. Can be used in redirect rules and filters.

Sub-URL: `/system/browser-families`

Method: GET

1.2.2 Groups API

Link group actions, e.g. create, update and delete link groups.

groups: list

Get list of current user link groups. Group ID should be used when creating new short link.

Sub-URL: `/groups`

Method: GET

group: add

Add new link group.

Sub-URL: `/groups`

Method: POST

Request fields:

name

Name of the new group.

group: show

Fetch link group details. Replace `{group_id}` in URL with group ID you want to fetch. Group IDs can be taken from `groups: list` response.

Sub-URL: `/groups/{group_id}`

Method: GET

group: update

Update link group details. Replace {group_id} in URL with group ID you want to update.

Sub-URL: /groups/{group_id}

Method: PATCH

Request fields:

name

New name of the group.

status

New status of the group. All status values are described in data dictionary.

color

(optional) HEX color of the group in #RRGGBB format.

group: delete

Delete existing link group and all its links. Replace {group_id} in URL with group ID you want to delete.

Sub-URL: /groups/{group_id}

Method: DELETE

1.2.3 Links API

Short link (short URL) actions, e.g. create, update and delete links.

links: list

List all short links in the group. Replace {group_id} in URL with group ID you want to list links for.

Sub-URL: /groups/{group_id}/links

Method: GET

link: add

Add new short link to the group.

Sub-URL: /groups/{group_id}/links

Method: POST

Request fields:

link_type_id

Type ID of the new link. Link types can be retrieved using `system: link-types` request.

domain_id

Domain ID of the new link. Domains can be retrieved using `system: domains` request.

data

New short link data. Will be stored in {data} portion of link type format. Data will be different for each type e.g. a full destination URL for a web link, email address for email type links, etc.

key

(optional) Key portion of new short link. Will be random generated if not present.

descr

(optional) Description of the new short link.

password

(optional) Short link access password. If set, visitors clicked the link will be prompted for a password before redirecting to the link destination.

valid_till

(optional) Short link expiry date in YYYY-MM-DD format.

link: show

Fetch short link details. Replace {link_id} in URL with link ID you want to fetch. Link IDs can be taken from `links: list` response.

Sub-URL: `/links/{link_id}`

Method: GET

link: update

Update link group details. Replace {link_id} in URL with link ID you want to update.

Sub-URL: `/links/{link_id}`

Method: PATCH

Request fields:

link_group_id

New group ID of the link. Groups can be retrieved using `groups: list` request.

status

New status of the link. All status values are described in data dictionary.

descr

(optional) Description of the new short link.

password

(optional) Short link access password. If set, visitors clicked the link will be prompted for a password before redirecting to the link destination.

valid_till

(optional) Short link expiry date in YYYY-MM-DD format.

link: delete

Delete existing link and all its data. Replace {link_id} in URL with link ID you want to delete.

Sub-URL: `/links/{link_id}`

Method: DELETE

link: set rules

Set short link redirect rules. Different alternate short link destinations are possible depending on visitor browser language, origin country and device software. Rule becomes true if all configured conditions are met. Rules are inspected using priority (from bigger to smaller) and first true rule is used. Default link destination is used if no rules are true.

Sub-URL: /links/{link_id}/rules

Method: PATCH

Request fields:

rules

Array of redirect rules. Operation completely replaces any existing rules defined for the link. Send empty array if you wish to clear all existing rules. Each rule is an object of following fields:

link_type_id

Type ID of the alternate redirect destination. Link types can be retrieved using `system: link-types` request.

data

Data of the alternate redirect destination. Will be stored in `{data}` portion of link type format. Data will be different for each type e.g. a full destination URL for a web link, email address for email type links, etc.

priority

Rule priority.

rule_browser

Array of browser families. Can be retrieved using `system: browser-families` request.

rule_country

Array of country codes. Can be retrieved using `system: countries` request.

rule_device

Array of traffic source types: `bot` - search engine bots, `desktop` - computers and notebooks, `mobile` - mobile devices, `tv` - TV and consoles, `wear` - wearable and peripheral devices.

rule_lang

Array of language codes. Can be retrieved using `system: languages` request.

rule_os

Array of OS families. Can be retrieved using `system: os-families` request.

link: get stats

Get short link statistics for time period. Replace `{link_id}` in URL with link ID.

Sub-URL: /links/{link_id}/stats

Method: POST

Request fields:

from

From datetime in `YYYY-MM-DDTHH:MM:SS.000Z` format.

to

To datetime in `YYYY-MM-DDTHH:MM:SS.000Z` format.

granula

Breakdown granula: 1 - per hour, 2 - per day.

tz

Timezone offset in minutes. Required for correct breakdown. Use 0 for UTC breakdown.

link: get logs

Get short link logs for time period. Replace {link_id} in URL with link ID.

Sub-URL: /links/{link_id}/logs

Method: POST

Request fields:

from

From datetime in YYYY-MM-DDTHH:MM:SS.000Z format.

to

To datetime in YYYY-MM-DDTHH:MM:SS.000Z format.

link: get log

Get full log details for specified log entry ID. List of log entries can be retrieved using **link: get logs** request.

Sub-URL: /links/{link_id}/logs/{log_id}

Method: GET

link: qr

Get short link QR code data. Replace {link_id} in URL with link ID.

Sub-URL: /links/{link_id}/qr

Method: POST

Request fields:

ecc

Error correction level: 1 - Low, 2 - Medium, 3 - Quartile, 4 - High.

style

Style: 1 - Classic, 2 - Rounded dots, 3 - Rounded all.

color

Color: 1 - Black, 2 - Solid color, 3 - 2-color gradient, 4 - 3-color gradient.

color1

Color 1. Used in color modes 2, 3, 4.

color2

Color 2. Used in color modes 3, 4.

color3

Color 3. Used in color mode 4.