



API Usage

This API consists of two main data sources:

1. LEGO Catalog - all official LEGO Sets/Parts data is available, including Minifigs.
2. User LEGO Collections - authenticated users can view and modify their LEGO Collection data stored on Rebrickable.

It does not include Rebrickable-specific data such as Sub-Sets, or B-Models. There is no Set/Part pricing data available, as that data is owned by external sites such as BrickLink or BrickOwl. Also, there is no longer any support for MOCs or MOC inventories, other than for alternate builds of Sets.

Authentication

Every call to the Rebrickable API needs to be authenticated in one of two ways (these examples use your key `4f8845c5d9212c179c08fe6f0e0d2d0c`):

1. Authorization Header - add a HTTP header named *Authorization* with the value of *key 4f8845c5d9212c179c08fe6f0e0d2d0c* (Note that your key must be prefixed with the text 'key ').

```
curl --header 'Authorization: key 4f8845c5d9212c179c08fe6f0e0d2d0c' https://rebrickable.com/
```

2. GET/POST parameter - include a GET or POST parameter called *key* with your key value.

```
curl https://rebrickable.com/api/v3/lego/colors/?key=4f8845c5d9212c179c08fe6f0e0d2d0c
```

Standard Parameters

For the GET calls that return a list of items, there are some optional parameters to help process the list results.

1. *page* - Lists are automatically paginated, and currently use a page size of 1000 (might change in the future). By default, it will only return results for the first page of items. You can request additional pages with the 'page' parameter (base 1). Alternatively, there is always a 'next' and 'previous' field in the results which contains the URL for the next and previous page of results.

2. *page_size* - The number of results to display for returned lists. Defaults to 100, but can be increased to a maximum of 1000 if required.
3. *ordering* - supported calls can be ordered by using the 'ordering' parameter. Simply provide the name of one of the top-level result fields to order by, or a comma separated (encoded to %2C) list of fields for nested ordering. If you place a - in front of a field name it will be sorted in reverse order.

Example:

```
curl https://rebrickable.com/api/v3/lego/colors/?key=4f8845c5d9212c179c08fe6f0e0d2d0c&page=2&order
```

REST Methods

This v3 API is based on a REST framework. This means the HTTP method types you use determines what happens with the data. Some REST definitions have differing behaviours, so to make things clear this is what the Rebrickable API uses:

- GET - Used to RETRIEVE items, always a read only operation.
- POST - Used to CREATE new items. If the item already exists, it will fail. Sometimes also used in place of GET calls if there are too many parameters or the data is too sensitive to be placed in a URL.
- PUT - Used to REPLACE existing items, you must specify every field. If the item does not exist it will fail.
- PATCH - Used to partially UPDATE existing items if you only want to change some of the fields. If the item does not exist it will fail.
- DELETE - Used to DELETE an existing item.

Response Codes

200	Success
201	Successfully created item
204	Item deleted successfully
400	Something was wrong with the format of your request
401	Unauthorized - your API key is invalid
403	Forbidden - you do not have access to operate on the requested item(s)
404	Item not found
429	Request was throttled - you are sending too many requests too fast. Normal user accounts are allowed to send on average one request/sec, with some small allowance for burst traffic. Example

response:

```
{ "detail": "Request was throttled. Expected available in 2 seconds." }
```

NOTE: if you continue to ignore these 429 responses and do not slow down your calls, your IP address will be temporarily banned.

Performance Tips

Calling the API for every part item can be slow and trigger throttle errors, or cause slow downs for everyone.

For example, instead of:

```
/lego/parts/3001/  
/lego/parts/3002/  
/lego/parts/3003/
```

It is far more efficient to perform a single call with:

```
/lego/parts/?part_nums=3001,3002,3003&inc_part_details=1
```

You can also reduce the size of the response in some cases, with extra parameters:

- `inc_color_details=0` will disable expansion of color fields in the response

Change Log

While I try my best to keep this API stable, there are inevitable bug fixes and small enhancements added over time. Most of the time they should not break existing functionality.

Date	Change
2020-11-28	Added optional <code>inc_color_details</code> param (defaults true) support for Part List Parts.
2020-07-20	Fixed <code>/lego/parts/{part_num}/colors/</code> to only return colors within sets (was also returning colors with <code>num_sets = 0</code>)
2020-05-29	Added optional <code>inc_minifig_parts</code> param (defaults false) to <code>/sets/{set_num}/parts/</code> endpoint.
2020-05-29	Added optional <code>inc_color_details</code> param (defaults true) to prevent color field expansion and

reduce response sizes.

2020-05-04	Added missing /minifigs/{set_num}/parts/ to see parts within Minifigs.
2020-04-19	Added support for Minifigs. Minifigs are basically a special type of Set and are separate items within Inventories. Users' Minifigs are determined automatically from the Sets they own, but they can also add them individually to Set Lists where they appear as normal Sets.
2020-04-19	Removed support for MOCs. There are too many sites stealing our MOC data, so I've decided to remove this API endpoint.
2019-05-10	Add print_of field to part detail view /lego/parts/{part_num}/.
2018-05-25	Part color details now uses correct color in the part_img_url field.
2018-04-11	Inventory parts list now uses correct color in the part.part_img_url field.
2018-04-11	External system filters in /lego/parts/ no longer return the external part number if an external system mapping is in place
2018-03-10	Added inc_part_details option to endpoints: /lego/parts/, /lego/parts/{part_num}/, /sets/{set_num}/parts/, /users/{user_token}/parts/, /users/{user_token}/partlists/{list_id}/parts/, /users/{user_token}/allparts/ which will return all detailed part fields for each part in the list.
2018-03-10	Added part_nums filter param to /lego/parts/
2018-03-10	Reduce default page_size to 100
2017-09-26	Include disabled Elements in responses
2017-09-14	Added external_ids section to Part listings
2017-05-23	/users/{token}/profile/ now returns additional fields: username, email, avatar_img
2017-05-23	/lego/sets/ search field now looks at both name and set_num
2017-04-21	Add proper headers in CORS responses
2017-04-19	Allow unauthenticated OPTIONS requests to help Ember apps (although it will not check if any authentication header is valid for the following GET/etc call)
2017-04-13	Include CORS header on 40x responses to help Angular apps
2017-03-29	Added element_img_url and part_img_url to /lego/elements/{element_id}/
2017-03-07	Added optional page_size query parameter for list results (default is still 1000 but will be reduced soon)
2017-02-09	Added calls for /users/{user_token}/sets/{set_num}/
2017-02-09	Added designer_name and designer_url to MOC response model

Example Code

There are quite a few [samples of code to be found on GitHub](#), a good example of a Python wrapper can be found at <https://github.com/xxao/rebrick>.

Help!

If you need assistance or think you've found a bug, please post to the [API Forum](#).

Swagger Documentation

Swagger URL = <https://rebrickable.com/api/v3/swagger/?format=openapi>

API Key

4f8845c5d9212c179c08fe6f0e0d2d0c

Set the key to use when testing below

lego

GET /api/v3/lego/colors/ Get a list of all Colors.

Implementation Notes

Get a list of all Colors.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page	<input type="text"/>	A page number within the paginated result set.	query	integer
page_size	<input type="text"/>	Number of results to return per page.	query	integer
ordering	<input type="text"/>	Which field to use when ordering the results.	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
<div>Try it out!</div>			

GET /api/v3/lego/colors/{id}/

Get details about a specific C

Implementation Notes

Get details about a specific Color.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>(required)</div>	A unique value identifying this color.	path	string
ordering	<div></div>	Which field to use when ordering the results.	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
<div>Try it out!</div>			

GET /api/v3/lego/elements/{element_id}/

Get details about a specific Elemer

Implementation Notes

Get details about a specific Element ID.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
element_id	<div>(required)</div>		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
<div>Try it out!</div>			

GET

/api/v3/lego/minifigs/

Get a list of Min

Implementation Notes

Get a list of Minifigs.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page	<input type="text"/>	A page number within the paginated result set.	query	integer
page_size	<input type="text"/>	Number of results to return per page.	query	integer
min_parts	<input type="text"/>		query	double
max_parts	<input type="text"/>		query	double
in_set_num	<input type="text"/>		query	string
in_theme_id	<input type="text"/>		query	string
ordering	<input type="text"/>	Which field to use when ordering the results.	query	string
search	<input type="text"/>	A search term.	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

GET

/api/v3/lego/minifigs/{set_num}/

Get details for a specific Mi

Implementation Notes

Get details for a specific Minifig.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
set_num	<div>(required)</div>		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
------------------	--------	----------------	---------

HTTP Status Code	Reason	Response Model	Headers
200			
<div>Try it out!</div>			

GET	/api/v3/lego/minifigs/{set_num}/parts/	Get a list of all Inventory Parts in this Mi
-----	--	--

Implementation Notes

Get a list of all Inventory Parts in this Minifig.

Optional parameter inc_part_details=1 can be used to return additional part fields, the same as for a single part lookup.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
set_num	<div>(required)</div>		path	string
page	<div></div>	A page number within the	query	integer