Nextcloud Bookmarks Documentation

Marvin Thomas Rabe, Stefan Klemm, Arthur Schiwon, Marcel Klel

Oct 22, 2021

Contents:

The Nextcloud Bookmarks app provides you with a web interface for collecting, organizing and sharing bookmarks to the sites on the web that are precious to you. You can browse and filter your bookmarks via tags, folders and by using the built-in search feature and you can share folders with users and groups and using public links. Furthermore, in order to access your bookmarks anywhere, it also allows you to synchronize third-party clients via a built-in REST API.

This documentation describes how to use this API.

CHAPTER 1

Authentication

1.1 User-based authentication

In order to access the REST API you will need to provide credentials for the user on behalf of which you'd like to access the bookmarks app. This should be done using Basic Auth and must happen for every request.

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark HTTP/1.1
Host: example.com
Accept: application/json
Authorization: basic 345678ikmnbvcdewsdfgzuiolkmnbvfr==
```

1.2 Token-based authentication

If a user has shared one of their folders publicly, you can access its contents via the token as part of the public link. You may bass this token to the various endpoints using the `token` GET-parameter or by setting it as part of the `Authorization header`.

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark?token=j5KJr7c HTTP/1.1
Host: example.com
Accept: application/json
```

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark HTTP/1.1
Host: example.com
Accept: application/json
Authorization: bearer j5KJr7c
```

CHAPTER 2

Bookmarks

Contents

2.1 Bookmark model

A bookmark has at least the following properties

`Bookmark`

Param int id The bookmarks unique id

Param string url The Uniform Resource Locator that this bookmark represents

Param string title A short humanly readable label for the bookmark

Param string description A longer description or note on the bookmark

Param array tags A list of tags this bookmark is tagged with

Param array folders The folders this bookmark has been added to

2.2 Query bookmarks

GET /public/rest/v2/bookmark

Synopsis Filter and query all bookmarks by the authenticated user.

New in version 0.11.0.

Query Parameters

- **tags[]** – An array of tags that bookmarks returned by the endpoint should have
- **page** – if this is non-negative, results will be paginated by `limit` bookmarks a page. Default: `0`.
- **limit** – Results will be paginated by this amount of bookmarks per page. Default: `10`.
- **sortby** – The column to sort the results by; one of `url`, `title`, `description`, `public`, `lastmodified`, `clickcount`. Default: `lastmodified`.
- **search[]** – An array of words to search for in the following columns `url`, `title`, `description`, `tags`
- **conjunction** – Set to `and` to require all search terms to be present, `or` if one should suffice. Default: `or`
- **folder** – Only return bookmarks that are direct children of the folder with the passed ID. The root folder has id `-1`.
- **url** – Only return bookmarks with this URL. With this parameter you can test whether a URL exists in the user's bookmarks.
- **unavailable** – Only return bookmarks that are dead links, i.e. return 404 status codes or similar.
- **archive** – Only return bookmarks that whose contents have been archived.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **data** (*array*) – The list of resulting bookmarks

Note that before v3.4.0 You couldn't mix `folder`, `url`, `unavailable` and `archive`.

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark?tags[]=firsttag&
↪tags[]=secondtag&page=-1 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```
  "status": "success",
  "data": [{ "id": 7, "title": "Google", "tags": ["firsttag"] }]
}
```

2.3 Create a bookmark

POST /public/rest/v2/bookmark

Synopsis Create a bookmark

New in version 0.11.0.

Parameters

- **id** – the url of the new bookmark
- **tags** (*array*) – Array of tags for this bookmark (these needn't exist and are created on-the-fly; this used to be *item[tags][]*, which is now deprecated)
- **title** (*string*) – the title of the bookmark. If absent the title of the html site referenced by *url* is used
- **description** (*string*) – A description for this bookmark
- **folders** (*array*) – An array of IDs of the folders this bookmark should reside in.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*object*) – The created bookmark

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/bookmark?tags[]=firsttag&
↪tags[]=secondtag&page=-1 HTTP/1.1
Host: example.com
Accept: application/json

{
  "url": "http://google.com",
  "title": "Google",
  "description":"in case i forget",
  "tags": ["search-engines", "uselessbookmark"]
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": {
    "id": 7,
    "url": "http://google.com",
    "title": "Google",
    "description":"in case i forget",
```

(continues on next page)

(continued from previous page)

```
    "tags": ["search-engines", "uselessbookmark"],
    "folders": [-1]
  }
}
```

2.4 Get a bookmark

GET /public/rest/v2/bookmark/(**int:** *id*)

Synopsis Retrieve a bookmark

New in version 0.11.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*object*) – The retrieved bookmark

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark/7 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": {
    "id": 7,
    "url": "http://google.com",
    "title": "Google",
    "description":"in case i forget",
    "tags": ["search-engines", "uselessbookmark"],
    "folders": [-1]
  }
}
```

2.5 Edit a bookmark

PUT /public/rest/v2/bookmark/(**int:** *id*)

Synopsis Edit a bookmark

New in version 0.11.0.

Parameters

- **id** – the url of the new bookmark
- **tags** (*array*) – Array of tags for this bookmark (these needn't exist and are created on-the-fly; this used to be *item[tags][]*, which is now deprecated)

- **title** (*string*) – the title of the bookmark. If absent the title of the html site referenced by *url* is used
- **description** (*string*) – A description for this bookmark
- **folders** (*array*) – An array of IDs of the folders this bookmark should reside in.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*object*) – The new bookmark after editing

Example:

```
PUT /index.php/apps/bookmarks/public/rest/v2/bookmark/7 HTTP/1.1
Host: example.com
Accept: application/json

{ "title": "Boogle" }
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": {
    "id": 7,
    "url": "http://google.com",
    "title": "Boogle",
    "description":"in case i forget",
    "tags": ["search-engines", "uselessbookmark"],
    "folders": [-1]
  }
}
```

2.6 Delete a bookmark

DELETE /public/rest/v2/bookmark/(**int:** *id*)

Synopsis Delete a bookmark

New in version 0.11.0.

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
DELETE /index.php/apps/bookmarks/public/rest/v2/bookmark/7 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

2.7 Get a preview image

GET /public/rest/v2/bookmark/(int: *id*)/image

Synopsis Retrieve the preview image of a bookmark

New in version 1.0.0.

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark/7/image HTTP/1.1
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: image/png

... binary data ...
```

2.8 Get a favicon

GET /public/rest/v2/bookmark/(int: *id*)/favicon

Synopsis Retrieve the favicon of a bookmark

New in version 1.0.0.

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark/7/favicon HTTP/1.1
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: image/png

... binary data ...
```

2.9 Export all bookmarks

GET /public/rest/v2/bookmark/export

Synopsis Export all bookmarks of the current user in a HTML file.

New in version 0.11.0.

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/bookmark/export HTTP/1.1
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/html

<html>
...
```

CHAPTER 3

Tags

Contents

3.1 Tag model

Tag

A tag is just a string that can be added as a label to a bookmark

3.2 Get list of tags

GET /public/rest/v2/tag

Synopsis Retrieve a list of tags

New in version 0.11.0.

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/tag HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

["politics", "satire", "tech", "music", "art", "blogs", "personal"]
```

3.3 Delete a tag

DELETE /public/rest/v2/tag/(string: *tag*)

Synopsis Delete a tag

New in version 0.11.0.

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
DELETE /index.php/apps/bookmarks/public/rest/v2/tag/politics HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "status": "success" }
```

3.4 Rename a tag

PUT /public/rest/v2/tag/(string: *tag*)

Synopsis Rename a tag

New in version 0.11.0.

Request JSON Object

- **name** (*string*) – The new name for the tag

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
PUT /index.php/apps/bookmarks/public/rest/v2/tag/politics HTTP/1.1
Host: example.com
Accept: application/json

{ "name": "satire" }
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "status": "success" }
```

CHAPTER 4

Folders

Contents

4.1 Folder model

Folder

Param int id The folder's unique id

Param string title The humanly-readable label for the folder

Param int parent_folder The folder's parent folder

Note: The root folder has the magic id `-1`, which is the same for every user.

4.2 Get full hierarchy

GET /public/rest/v2/folder

Synopsis Retrieve the folder hierarchy

New in version 0.15.0.

Query Parameters

- **root** (*int*) – The id of the folder whose contents to retrieve (Default: `-1`, which is the root folder)
- **layers** (*int*) – How many layers of folders to return at max. By default, all layers are returned.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **data** (*array*) – The folder hierarchy

Response JSON Array of Objects

- **id** (*int*) – The id of the folder
- **title** (*string*) – the folder title
- **parent_folder** (*int*) – the folder's parent folder
- **children** (*array*) – The folder's children (folders only)

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success", "data": [
    {"id": 1, "title": "work", "parent_folder": -1},
    {"id": 2, "title": "personal", "parent_folder": -1, "children": [
```

(continues on next page)

(continued from previous page)

```
      {"id": 3, "title": "garden", "parent_folder": 2},
      {"id": 4, "title": "music", "parent_folder": 2}
    ]},
  ]
}
```

4.3 Get single folder

GET /public/rest/v2/folder/(**int:** *id*)

Synopsis Retrieve a single folder

New in version 0.15.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*object*) – The retrieved folder

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/2 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": {
    "id": 2,
    "title": "My Personal Bookmarks",
    "parent_folder": -1
  }
}
```

4.4 Create a folder

POST /public/rest/v2/folder

Synopsis Create a new folder

New in version 0.15.0.

Request JSON Object

- **title** (*string*) – The title of the new folder
- **parent_folder** (*int*) – The id of the parent folder for the new folder

Response JSON Object

- **status** (*string*) – `success` or `error`

- **item** (*object*) – The new folder

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/folder HTTP/1.1
Host: example.com
Accept: application/json

{"title": "sports", "parent_folder": "-1"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": {
    "id": 5,
    "title": "sports",
    "parent_folder": "-1"
  }
}
```

4.5 Edit a folder

PUT /public/rest/v2/folder/(**int:** *id*)

Synopsis Edit an existing folder

New in version 0.15.0.

Request JSON Object

- **title** (*string*) – The title of the new folder
- **parent_folder** (*int*) – The id of the parent folder of the folder

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*object*) – The new folder

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/folder/5 HTTP/1.1
Host: example.com
Accept: application/json

{"title": "optional physical activity"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
```

(continues on next page)

(continued from previous page)

```
  "item": {
    "id": 5,
    "title": "optional physical activity",
    "parent_folder": -1
  }
}
```

4.6 Hash a folder

GET /public/rest/v2/folder/(**int:** *id*)**/hash**

Synopsis Compute the hash of a folder

New in version 1.0.0.

Parameters

- **fields** (*array*) – All bookmarks fields that should be hashed (default: `title`, `url`)

Response JSON Object

- **status** (*string*) – `success` or `error`
- **data** (*string*) – The SHA256 hash in hexadecimal notation

This endpoint is useful for synchronizing data between the server and a client. By comparing the hash of the data on your client with the hash from the server you can figure out which parts of the tree have changed.

The algorithm works as follows:

- Hash endpoint: return `hashFolder(id, fields)`
- `hashFolder(id, fields)`
  - set `childrenHashes` to empty array
  - for all children of the folder
    - if it's a folder
      - add to `childrenHashes`: `hashFolder(folderId, fields)`
    - if it's a bookmark
      - add to `childrenHashes`: `hashBookmark(bookmarkId, fields)`
  - set `object` to an empty dictionary
  - set `object[title]` to the title of the folder, if this is not the root folder
  - set `object[children]` to the value of `childrenHashes`
  - set `json` to `to_json(object)`
  - Return `sha256(json)`
- `hashBookmark(id, fields)`
  - set `object` to an empty dictionary/hashmap
  - for all entries in `fields`
    - set `object[field]` to the value of the associated field of the bookmark

  - Return `sha256(to_json(object))`

- `to_json`: A JSON stringification algorithm that adds no unnecessary white-space and doesn't use JSON's backslash escaping unless necessary (character set is UTF-8)
- `sha256`: The SHA-256 hashing algorithm

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/hash HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "status": "success", "data": "6543a23c78aefd0274f3ac98de98723" }
```

4.7 Delete a folder

DELETE /public/rest/v2/folder/(**int:** *id*)

Synopsis Delete a folder

New in version 0.15.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*object*) – The new folder

Example:

```
DELETE /index.php/apps/bookmarks/public/rest/v2/folder/5 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

4.8 Add bookmark to folder

POST /public/rest/v2/folder/(**int:** *folder_id*)**/bookmarks/**
int: *bookmark_id*

Synopsis Add a bookmark to a folder

New in version 0.15.0.

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/folder/5/bookmarks/418 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

4.9 Remove bookmark from folder

DELETE /public/rest/v2/folder/(**int:** *folder_id*)**/bookmarks/**
int: *bookmark_id*

Synopsis Remove a bookmark from a folder

New in version 0.15.0.

Response JSON Object

- **status** (*string*) – `success` or `error`

If this is the only folder this bookmark resides in, the bookmark will be deleted entirely.

Example:

```
DELETE /index.php/apps/bookmarks/public/rest/v2/folder/5/bookmarks/418 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

4.10 Get folder's content order

GET /public/rest/v2/folder/(**int:** *folder_id*)**/childorder**

Synopsis Retrieve the order of contents of a folder

New in version 0.15.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **data** (*array*) – An ordered list of child items

Response JSON Array of Objects

- **type** (*string*) – Either `folder` or `bookmark`
- **id** (*string*) – The id of the bookmark or folder

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/childorder HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "data": [
    {"type": "folder", "id": 17},
    {"type": "bookmark", "id": 204},
    {"type": "bookmark", "id": 192},
    {"type": "bookmark", "id": 210}
  ]
}
```

4.11 Set folder's content order

PATCH /public/rest/v2/folder/(**int:** *folder_id*)**/childorder**

Synopsis Set the order of contents of a folder

New in version 0.15.0.

Request JSON Object

- **data** (*array*) – An ordered list of child items

Request JSON Array of Objects

- **type** (*string*) – Either `folder` or `bookmark`
- **id** (*string*) – The id of the bookmark or folder

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
PATCH /index.php/apps/bookmarks/public/rest/v2/folder/5/childorder HTTP/1.1
Host: example.com
Accept: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "status": "success",
  "data": [
    {"type": "folder", "id": 17},
    {"type": "bookmark", "id": 204},
    {"type": "bookmark", "id": 192},
    {"type": "bookmark", "id": 210}
  ]
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

4.12 Get folder's contents

GET /public/rest/v2/folder/(**int:** *folder_id*)**/children**

Synopsis Retrieve all of a folder's contents (with varying depth)

New in version 3.0.0.

Query Parameters

- **layers** (*int*) – How many layers of descendants to return at max. By default only immediate children are returned.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **data** (*array*) – An ordered list of child items

Response JSON Array of Objects

- **type** (*string*) – Either `folder` or `bookmark`
- **id** (*string*) – The id of the bookmark or folder

If the type of the item is `folder`

Response JSON Array of Objects

- **title** (*string*) – The title of the folder
- **userId** (*string*) – The owner of the folder
- **children** (*array*) – The children of the folder. This is only set, when the number of layers to return includes this folder.

If the type of the item is `bookmark`

Response JSON Array of Objects

- **url** (*string*) – The URL of the bookmark

- **title** (*string*) – The title of the bookmark
- **description** (*string*) – Description of the bookmark

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/children HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "data": [
    {"type": "folder", "id": "17", "title": "foo", "userId": "admin"},
    {"type": "bookmark", "id": "204", "title": "Nextcloud", "url": "https://
↪nextcloud.com/"},
    {"type": "bookmark", "id": "204", "title": "Google", "url": "https://google.
↪com/"},
  ]
}
```

4.13 Get folder's contents

GET /public/rest/v2/folder/(int: *folder_id*)/count

Synopsis Retrieve the number of bookmarks contained in this folder and all descendants

New in version 3.4.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*int*) – The number of descendant bookmarks

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/count HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": 512
}
```

4.14 Get public token

GET /public/rest/v2/folder/(int: *folder_id*)/publictoken

Synopsis Retrieve the public token of a folder that has been shared via a public link

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*string*) – The public token

To use the token either make API requests with it (see *User-based authentication*). Or point your browser to `https://yournextcloud.com/index.php/apps/bookmarks/public/{token}`

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/publictoken HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": "dk3J8Qm"
}
```

4.15 Get public token

POST /public/rest/v2/folder/(int: *folder_id*)/publictoken

Synopsis Create a public link for a folder

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*string*) – The token that can be used to access the folder publicly.

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/folder/5/publictoken HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```
  "status": "success",
  "item": "dk3J8Qm"
}
```

4.16 Delete public token

DELETE /public/rest/v2/folder/(**int:** *folder_id*)**/publictoken**

Synopsis Remove the public link for a folder

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/folder/5/publictoken HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
}
```

CHAPTER 5

Shares

Contents

5.1 Share model

`Share`

Param int id The share's unique id

Param int folderId The id of the folder that was shared

Param string participant The id of the participant that the folder was shared to

Param int type The participant type. Currently either `0` for users, or `1` for groups.

Param boolean canWrite Whether the participant has write access.

Param boolean canShare Whether the participant is allowed to reshare the folder or subfolders to other users, including the creation of public links.

5.2 Create a share

POST /public/rest/v2/folder/(**int:** *folder_id*)**/shares**

Synopsis Create a share for a folder

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*share*) – The new share

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/folder/5/shares HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
  "item": {
      "id": 5,
      "folderId": 201,
      "participant": "friends",
      "type": 1,
      "canWrite": false,
      "canShare": false
  }
}
```

5.3 Get folder's shares

GET /public/rest/v2/folder/(**int:** *folder_id*)**/shares**

Synopsis Retrieves all shares of a folder

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **data** (*array*) – The shares of this folder

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/shares HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
  "data": [
      {
          "id": 5,
          "folderId": 201,
          "participant": "friends",
          "type": 1,
          "canWrite": false,
          "canShare": false
      }
  ]
}
```

5.4 Get public token

GET /public/rest/v2/folder/(**int:** *folder_id*)**/publictoken**

Synopsis Retrieve the public token of a folder that has been shared via a public link

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*share*) – The public token

To use the token either make API requests with it (see *User-based authentication*). Or point your browser to `https://yournextcloud.com/index.php/apps/bookmarks/public/{token}`

Example:

```
GET /index.php/apps/bookmarks/public/rest/v2/folder/5/publictoken HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
  "item": "dk3J8Qm"
}
```

5.5 Get share

POST /public/rest/v2/share/(**int:** *share_id*)

Synopsis Get a share by id

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*share*) – The requested share

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/share/17 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
  "data": [
      {
          "id": 17,
          "folderId": 201,
          "participant": "friends",
          "type": 1,
          "canWrite": false,
          "canShare": false
      }
  ]
}
```

5.6 Edit share

PUT /public/rest/v2/share/(**int:** *share_id*)

Synopsis Get a share by id

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`
- **item** (*share*) – The requested share

Example:

```
PUT /index.php/apps/bookmarks/public/rest/v2/share/17 HTTP/1.1
Host: example.com
Accept: application/json

{
  "canWrite": true,
  "canShare": false
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
  "data": [
      {
          "id": 17,
          "folderId": 201,
          "participant": "friends",
          "type": 1,
          "canWrite": true,
          "canShare": false
      }
  ]
}
```

5.7 Delete share

DELETE /public/rest/v2/share/(**int:** *share_id*)

Synopsis Delete a share

New in version 3.0.0.

Response JSON Object

- **status** (*string*) – `success` or `error`

Example:

```
POST /index.php/apps/bookmarks/public/rest/v2/share/17 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success",
}
```

CHAPTER 6

Changes

6.1 Breaking changes from v2.x to v3.x

With the upgrade from v2.x of the bookmarks app to v3.x several breaking changes in the API were introduced which were not possible to avoid.

`Bookmark`

> **Param int id** All bookmark IDs are now integer values instead of strings

`Folder`

> **Param int id** All folder IDs are now integer values instead of strings

`GET /public/rest/v2/bookmark`
: This endpoint no longer accepts the `item[tags]` query parameter. Use the normal `tags` param

`POST /public/rest/v2/bookmark`
: This endpoint no longer accepts the `item[tags]` query parameter. Use the normal `tags` param

`PUT /public/rest/v2/bookmark`
: This endpoint no longer accepts the `item[tags]` query parameter. Use the normal `tags` param

HTTP Routing Table

/public