fyyd api

Documentations version: 0.12 (2018-05-20)

This is the documentation for the API to fyyd, the podcast search engine, directory and slow cooker.

Take my apologies for this simple markdown, I tried to use something more fancy, but I took an arrow to my knee.

~~Please be aware: This version is most likely to change in the very near future. It's my first kind of that API and I'm still learning how to do things the more or less correct way. All that works and all that is fine, but it may not be optimal. I want it to be at least near to optimal.~~

Version 0.2 is now fixed and nailed to the wall. What does that mean? It means, that requests, parameters and responses won't change for that version and applications and scripts will not face substantial modifications, rendering them inoperabel. What will happen though, is that requests and parameters will be added and additional data inside responses will occur, as long as these changes don't touch functionality.

Bugs or inconsistencies, flaws and so on will stay forever with v0.2, but may be addressed by additional requests or responses. Whatever I learn with this version will find it's way into 0.3 or whatever I'll number the next substantial release.

Endpoint for version 0.2

https://api.fyyd.de/0.2/

always prefix this for every request or else you'll end up in version 0.1 which is mostly different from 0.2.

Table of contents

Preface

Podcasts and their metadata always has been a fascinating subject for me. So much data, so few approaches to collect and bring it to good use.

Now it's 2017 and there are a few (I know of three, maybe four) more or less experimental projects to work with this gigantic loads of data. One of that projects is fyyd, my attempt to create a

- podcast search engine
- podcast directory
- podcast recommendation system

fyyd is a freetime project, that get's some love now and then. The progress is as fast as I can afford some free time beside family, job and whatever one has to do.

As I started fyyd, I knew that an API would be crucial for this project to escape the scope of just a webpage and to find more stunning ways to do whatever one can do with that data. As always: The crowd will find a way to use the data you would never have dreamed of.

Version 0.1 of that API found it's way to a small group of developers, leading into the development of [Skoon](#), an iOS-App to curate podcast's episodes directly from some podcatchers and the integration of fyyd into [podcat](#), a lovely designed podcatcher that pleasantly differs from most machine like user interfaces.

Version 0.1 had its problems and was a rush job. Thanks to Jeanette and Stefan here and now you see the documentation for version 0.2.

Thanks to all who helped, will help and contribute to this small project.

Authorization

Authorization for 3rd party apps and services is provided with an oauth2 interface. At the moment this is a little bit rudimentary, but working.

App registration

First step is to register your app at fyyd.de. At the time of writing this documentation (2017-04-17), you have to be a registered developer at fyyd. This is not as a big deal as it might sound, but I'd like to know, who is using auth with the API and what happens. Just register an account at fyyd.de and give me a short note that you'd like to use the API via mail.

When this is done, head to https://fyyd.de/dev/app/ to create the app's registration at fyyd.

You then create the credentials and maybe an accesstoken (which only works on app context, which is non existent at this time).

Auth workflow

user context

Head your client to **https://fyyd.de/oauth/authorize?client_id=XYZ** where client_id is your client's id (wow... ;)

The user logs in (if neccessary) and authorizes the app.

fyyd now redirects to your registered app's callback-URL, appending the token via #token=ACCESSTOKEN.

In all future requests, add an authorization header:

`Authorization: Bearer ACCESSTOKEN`

app context

The authorization for application only level is working, but there is no use-case for that at the moment.

The most simple way would to create the accesstoken directly at the developers page at fyyd. You may also send a POST request to https://fyyd.de/oauth/token, containing

`grant_type=client_credentials`

inside the body.

Additionally you need to either set

an Authorization Header like

`Authorization: Basic base64encode($client_id:$client_secret)`

(please: base64encode has to be calculated and not included in the headers...)

or added as

`&client_id=$client_id&client_secret=$client_secret`

in the requests body.

Error Handling

In case an error occurs, the API repsonses with an appropriate HTTP status code and, in case it's not a 404, a more or less helpful msg along inside a JSON object.

What could possibly go wrong? Here are some examples:

Examples

Missing authorization

```
1   {
2       "errors": {
3           "code": 0,
4           "message": "not authorized!",
5           "http_code": 401,
6           "http_message": "HTTP 401 (GET \/0.2\/account\/info)"
7       },
8       "meta": {
9           "API_INFO": {
10              "API_VERSION": 0.2
11          }
12      }
13  }
```

Doing smth really forbidden

```json
{
  "errors": {
    "code": 0,
    "message": "this is a personal feed and cannot be deleted",
    "http_code": 403,
    "http_message": "HTTP 403 (POST /0.2/curation/delete)"
  },
  "meta": {
    "API_INFO": {
      "API_VERSION": 0.2
    }
  }
}
```

Missing parameter

```json
{
    "errors": {
        "code": 0,
        "message": "no nick or user id provided",
        "http_code": 400,
        "http_message": "HTTP 400 (GET \/0.2\/user\/curations)"
    },
    "meta": {
        "API_INFO": {
            "API_VERSION": 0.2
        }
    }
}
```

Unknown resource

```
1   {
2       "errors": {
3           "code": 0,
4           "message": "user not found",
5           "http_code": 400,
6           "http_message": "HTTP 400 (GET \/0.2\/user\/curations)"
7       },
8       "meta": {
9           "API_INFO": {
10              "API_VERSION": 0.2
11          }
12      }
13  }
```

Account

Requests for /account always refer to the user identified by the accesstoken. Thus, you of course need to authenticate to use this requests. All other requests are free until now.

[GET /account/info]

Description

Returns general information about the account

Parameters

Authorization

required

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "API_INFO": {
6               "API_VERSION": 0.2
7           }
8       },
9       "data": {
```

```
10          "nick": "your-nick",
11          "id": 42,
12          "fullname": "Jon Doe",
13          "bio": "This is me!",
14          "url": "https:\/\/fyyd.de\/",
15          "layoutImageURL": "https:\/\/img.fyyd.de\/user\/layout\/42.jpg",
16          "thumbImageURL": "https:\/\/img.fyyd.de\/user\/thumbs\/42.png",
17          "microImageURL": "https:\/\/img.fyyd.de\/user\/micro\/42.png"
18      }
19  }
```

[GET /account/curations]

Description

Returns this account's curations. Includes non-public curations as well.

Parameters

Authorization

required

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "API_INFO": {
6               "API_VERSION": 0.2
7           }
8       },
9       "data": [
10          {
11              "title": "Retro Computing",
12              "id": 1099,
13              "description": "Alles \u00fcber Rechner, f\u00fcr die es garantiert
    keine Garantie mehr gibt.",
14              "layoutImageURL":
    "https:\/\/img.fyyd.de\/curation\/layout\/1099.jpg",
15              "thumbImageURL":
    "https:\/\/img.fyyd.de\/curation\/thumbs\/1099.png",
16              "microImageURL":
    "https:\/\/img.fyyd.de\/curation\/micro\/1099.png",
17              "public": 0,
```

```
18            "type": 1,
19            "slug": "retro-computing",
20            "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/retro-computing",
21            "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/retro-computing"
22        },
23        {
24            "title": "Netzpolitik",
25            "id": 623,
26            "description": "Vorsicht, k\u00f6nnte Spuren von Netzpolitik
    enthalten!",
27            "layoutImageURL":
    "https:\/\/img.fyyd.de\/curation\/layout\/623.jpg",
28            "thumbImageURL":
    "https:\/\/img.fyyd.de\/curation\/thumbs\/623.png",
29            "microImageURL": "https:\/\/img.fyyd.de\/curation\/micro\/623.png",
30            "public": 1,
31            "type": 1,
32            "slug": "netzpolitik",
33            "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/netzpolitik",
34            "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/netzpolitik"
35        },
36        {
37            "title": "Podcasting",
38            "id": 586,
39            "description": "Alles \u00fcber das Podcasting",
40            "layoutImageURL":
    "https:\/\/img.fyyd.de\/curation\/layout\/586.jpg",
41            "thumbImageURL":
    "https:\/\/img.fyyd.de\/curation\/thumbs\/586.png",
42            "microImageURL": "https:\/\/img.fyyd.de\/curation\/micro\/586.png",
43            "public": 1,
44            "type": 1,
45            "slug": "podcasting",
46            "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/podcasting",
47            "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/podcasting"
48        }
49
50    ]
51 }
```

[GET /account/collections]

Description

Returns this account's collections.

Parameters

Authorization

required

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "API_INFO": {
6               "API_VERSION": 0.2
7           }
8       },
9       "data": [
10          {
11              "title": "Netzpolitik",
12              "id": 1,
13              "description": "(Netz)politische Podcasts",
14              "layoutImageURL":
    "https:\/\/img.fyyd.de\/collection\/layout\/1.jpg",
15              "thumbImageURL":
    "https:\/\/img.fyyd.de\/collection\/thumbs\/1.png",
16              "microImageURL": "https:\/\/img.fyyd.de\/collection\/micro\/1.png",
17              "slug": "netzpolitik",
18              "url": "https:\/\/fyyd.de\/user\/eazy\/collection\/netzpolitik"
19          },
20          {
21              "title": "Gamechanger",
22              "id": 8,
23              "description": "",
24              "layoutImageURL":
    "https:\/\/img.fyyd.de\/collection\/layout\/8.jpg",
25              "thumbImageURL":
    "https:\/\/img.fyyd.de\/collection\/thumbs\/8.png",
26              "microImageURL": "https:\/\/img.fyyd.de\/collection\/micro\/8.png",
27              "slug": "gamechanger",
28              "url": "https:\/\/fyyd.de\/user\/eazy\/collection\/gamechanger"
29          },
30          {
31              "title": "Lieblingspodcasts",
32              "id": 11,
33              "description": "Was ich so mit auf die Insel nehmen w\u00fcrde..",
34              "layoutImageURL":
    "https:\/\/img.fyyd.de\/collection\/layout\/11.jpg",
35              "thumbImageURL":
    "https:\/\/img.fyyd.de\/collection\/thumbs\/11.png",
```

```
36          "microImageURL":
    "https:\/\/img.fyyd.de\/collection\/micro\/11.png",
37          "slug": "lieblingspodcasts",
38          "url":
    "https:\/\/fyyd.de\/user\/eazy\/collection\/lieblingspodcasts"
39       }
40     ]
41 }
```

User

Gets public available information about a registered account at fyyd

[GET /user]

Description

Retrieves the accounts profile information

Parameters

- **user_id (required, int)** the users account id
- **or nick (optional, string)** identifies the user by nick

Response

```
1  {
2    "status": 1,
3    "msg": "ok",
4    "meta": {
5        "API_INFO": {
6            "API_VERSION": 0.2
7        }
8    },
9    "data": {
10       "nick": "eazy",
11       "id": 42,
12       "fullname": "John Doe",
13       "bio": "This is me!",
14       "url": "https:\/\/fyyd.de\/",
15       "layoutImageURL": "https:\/\/img.fyyd.de\/user\/layout\/42.jpg",
16       "thumbImageURL": "https:\/\/img.fyyd.de\/user\/thumbs\/42.png",
17       "microImageURL": "https:\/\/img.fyyd.de\/user\/micro\/42.png"
18   }
```

```
19    }
```

[GET /user/curations[/episodes]]

Description

Retrieves the curations maintained by the given user. You may append /episodes to also get the content of this curations. If you're authenticated and you request your own curations, non-public curations will be included also.

Parameters

- **user_id (required, int)** the users account id
- **or nick (optional, string)** identifies the user by nick

Response

```
 1   {
 2     "status": 1,
 3     "msg": "ok",
 4     "meta": {
 5         "API_INFO": {
 6             "API_VERSION": 0.2
 7         }
 8     },
 9     "data": [
10         {
11             "title": "Netzpolitik",
12             "id": 623,
13             "description": "Vorsicht, k\u00f6nnte Spuren von Netzpolitik
     enthalten!",
14             "layoutImageURL": "https:\/\/img.fyyd.de\/curation\/layout\/623.jpg",
15             "thumbImageURL": "https:\/\/img.fyyd.de\/curation\/thumbs\/623.png",
16             "microImageURL": "https:\/\/img.fyyd.de\/curation\/micro\/623.png",
17             "public": 1,
18             "type": 1,
19             "slug": "netzpolitik",
20             "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/netzpolitik",
21             "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/netzpolitik"
22         },
23         {
24             "title": "Podcasting",
25             "id": 586,
26             "description": "Alles \u00fcber das Podcasting",
27             "layoutImageURL": "https:\/\/img.fyyd.de\/curation\/layout\/586.jpg",
28             "thumbImageURL": "https:\/\/img.fyyd.de\/curation\/thumbs\/586.png",
29             "microImageURL": "https:\/\/img.fyyd.de\/curation\/micro\/586.png",
30             "public": 1,
31             "type": 1,
```

```
32          "slug": "podcasting",
33          "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/podcasting",
34          "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/podcasting"
35        }
36    ]
37  }
```

[GET /user/collections[/podcasts]]

Description

Retrieves the collections maintained by the given user. You may append /podcasts to also get the content of this collections.

Parameters

- **user_id (required, int)** the users account id
- **or nick (optional, string)** identifies the user by nick

Response

```
1   {
2     "status": 1,
3     "msg": "ok",
4     "meta": {
5         "API_INFO": {
6             "API_VERSION": 0.2
7         }
8     },
9     "data": [
10        {
11            "title": "Netzpolitik",
12            "id": 1,
13            "description": "(Netz)politische Podcasts",
14            "layoutImageURL": "https:\/\/img.fyyd.de\/collection\/layout\/1.jpg",
15            "thumbImageURL": "https:\/\/img.fyyd.de\/collection\/thumbs\/1.png",
16            "microImageURL": "https:\/\/img.fyyd.de\/collection\/micro\/1.png",
17            "slug": "netzpolitik",
18            "url": "https:\/\/fyyd.de\/user\/eazy\/collection\/netzpolitik"
19        },
20        {
21            "title": "Gamechanger",
22            "id": 8,
23            "description": "",
24            "layoutImageURL": "https:\/\/img.fyyd.de\/collection\/layout\/8.jpg",
25            "thumbImageURL": "https:\/\/img.fyyd.de\/collection\/thumbs\/8.png",
26            "microImageURL": "https:\/\/img.fyyd.de\/collection\/micro\/8.png",
27            "slug": "gamechanger",
28            "url": "https:\/\/fyyd.de\/user\/eazy\/collection\/gamechanger"
```

```
29          }
30      ]
31  }
```

Action

This is the actions API to fyyd. It's a more or less formfree key-value-store.

Requests for /action always refer to the user identified by the accesstoken and the used app.

[POST /action]

Description

Saves an action for the authorized user.

The action always referes to an object inside fyyd's database, which may be:

- an episode
- a podcast
- a curation
- a collection
- a user

The action itself is represented by a string you may choose freely. Some of these actions will be used in future features of fyyd. This is not implemented at the time this documentation is written.

Authorization

required

Parameters

- **object_id (required, int)** the id of the object
- **object_type (required, string)** the type of the object. episode, podcast, curation, collection, user
- **action (required, string)** a free to choose action
- **metadata (optional, string)** further data you'd like to associate to this action

Response

Returns 204 - No Content

[POST /action/delete]

Description

Deletes an action for the authorized user.

Authorization

required

Parameters

- **action_id (required, int)** the id of the action

Response

Returns 204 - No Content

[GET /action]

Description

Gets all actions for the authorized user, triggered by the authorized app and filtered by the given criteria.

Authorization

required

Parameters

- **object_id (optional, int)** the id of the object
- **object_type (optional, string)** the type of the object. episode, podcast, curation, collection, user
- **action (optional, string)** a free to choose action
- **metadata (optional, string)** further data you'd like to associate to this action
- **date_start (optional, date string)** optional start date for this query
- **date_end (optional, date string)** optional end date for this query

Choose one or more of these parameters. Providing none of them returns all actions issued by the logged in user with the authorized app.

Response

```
1  {
2    "status": 1,
3    "msg": "ok",
4    "meta": {
5      "API_INFO": {
6        "API_VERSION": 0.2
7      }
8    },
```

```
 9      "data": [
10          {
11              "id": 1,
12              "object_id": 85,
13              "object_type": "podcast",
14              "action": "test",
15              "metadata": "More data",
16              "date": "2017-04-17 20:04:22",
17              "user_id": 23,
18              "app_id": 42
19          },
20          {
21              "id": 2,
22              "object_id": 85,
23              "object_type": "podcast",
24              "action": "test",
25              "metadata": "More data, really!",
26              "date": "2017-04-17 20:09:01",
27              "user_id": 23,
28              "app_id": 42
29          }
30      ]
31  }
```

Podcast

Retrieves information about a podcast

[GET /podcast/[/episodes]]

Description

Gets information about the podcast with id {id}. You may retrieve the episodes by appending /episodes.

Additionally you can address the resultset with {page} and {count}.

Parameters

- **podcast_id (required, int)** the podcast's id
- *or* **podcast_slug (alternative, string)** the podcast's slug
- **page (optional, int)** the page you want to address, default: 0
- **count (optional,int)** the page's size, default: 50

Response

```json
{
  "status": 1,
  "msg": "ok",
  "meta": {
    "paging": {
            "count": 1,
            "page": 0,
            "first_page": 0,
            "last_page": 22,
            "next_page": 1,
            "prev_page": null
        },
        "API_INFO": {
            "API_VERSION": 0.2
        }
    },
    "data": {
        "title": "Schall und Rauch",
        "id": 64,
        "xmlURL": "http:\/\/schallrauch.hoersuppe.de\/sur\/m4a",
        "htmlURL": "http:\/\/schallrauch.hoersuppe.de\/sur\/show",
        "imgURL":
"http:\/\/schallrauch.hoersuppe.de\/feedimages\/itunes1400.png",
        "status": 399,
        "slug": "schall-und-rauch",
        "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/64.jpg",
        "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/64.png",
        "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/64.png",
        "language": "de",
        "lastpoll": "2017-04-17 14:42:41",
        "generator": "firtz podcast publisher v2.0",
        "categories": [
              52
          ],
        "lastpub": "2016-01-08 21:15:22",
        "rank": 22,
        "url_fyyd": "https:\/\/fyyd.de\/podcast\/64",
        "description": "Alexander und Christian, zwei mittelreife Fr\u00fcchte
der 70er, sozialisiert in den 80ern ver\u00f6ffentlichen ihre monatlichen
Schw\u00e4tzchen auf der Parkbank.",
        "subtitle": "Zwei Opas erz\u00e4hlen vom Krieg",
        "episodes": [
            {
                "title": "Stasi, Sherlock und die ARD",
                "id": 1742,
                "guid": "sur-sur022",
                "url": "http:\/\/schallrauch.hoersuppe.de\/sur\/show\/sur022",
                "enclosure":
"http:\/\/cdn.podseed.org\/schallundrauch\/sur022.m4a",
```

```
46              "podcast_id": 64,
47              "imgURL":
    "http:\/\/schallrauch.hoersuppe.de\/feedimages\/itunes1400.png",
48              "pubdate": "2016-01-08 21:15:22",
49              "duration": 5579,
50              "url_fyyd": "https:\/\/fyyd.de\/episode\/1742",
51              "description": "<p>Alexander erz\u00e4hlt von seinen
    Eindr\u00fccken beim Besuch der BStU, war dar\u00fcber hinaus in London und
    macht mir die Nase lang, weil er dort die aktuelle Folge Sherlock schon
    geschaut hat. Im Hotel. So ein Hund!<\/p>\n\n<p>Daf\u00fcr aber hat Deutschland
    durchaus etwas zu bieten im Bereich der Serien, man sollte nur nicht unbedingt
    beim Tatort suchen. Alexander rezitiert Schweiger, was dann schnell das Ende
    der Folge einl\u00e4utet&hellip;<\/p>\n"
52          }
53      ]
54  }
55 }
```

[GET /podcast/season]

Description

Gets episodes from a season of a podcast with id {podcast_id}.

Additionally you can address the resultset with {page} and {count}.

Parameters

- **podcast_id (required, int)** the podcast's id
- *or* **podcast_slug (alternative, string)** the podcast's slug
- **season_number (required, int)** the seasons number
- **episode_number (optional, int)** an episodes number to retrieve
- **page (optional, int)** the page you want to address, default: 0
- **count (optional,int)** the page's size, default: 50

Response

```
1  {
2      "status": 1,
3      "msg": "ok",
4      "meta": {
5          "paging": {
6              "count": 50,
7              "page": 0,
8              "first_page": 0,
9              "last_page": 0,
10             "next_page": null,
```

```json
11              "prev_page": null
12          },
13          "API_INFO": {
14              "API_VERSION": 0.2
15          }
16      },
17      "data": {
18          "title": "Raumzeit",
19          "id": 703,
20          "xmlURL": "https:\/\/feeds.metaebene.me\/raumzeit\/m4a",
21          "htmlURL": "http:\/\/raumzeit-podcast.de",
22          "imgURL": "https:\/\/meta.metaebene.me\/media\/raumzeit\/raumzeit-icon-
    1400x1400.jpg",
23          "status": 200,
24          "slug": "raumzeit",
25          "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/703.jpg",
26          "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/703.png",
27          "smallImageURL": "https:\/\/img.fyyd.de\/pd\/small\/703.jpg",
28          "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/703.png",
29          "language": "de",
30          "lastpoll": "2018-01-21T11:04:01+01:00",
31          "generator": "Podlove Podcast Publisher v2.7.0.build467",
32          "categories": [
33              48,
34              62
35          ],
36          "lastpub": "2018-01-20T11:03:23+01:00",
37          "rank": 46,
38          "url_fyyd": "https:\/\/fyyd.de\/podcast\/raumzeit\/0",
39          "description": "Raumzeit ist eine Serie von Gespr\u00e4chen mit
    Wissenschaftlern, Ingenieuren, Astronauten und Projektleitern \u00fcber
    Raumfahrt. Jede Episode r\u00fcckt einen Themenbereich in den Fokus und
    diskutiert ausf\u00fchrlich alle Aspekte und Details.  ",
40          "subtitle": "Raumfahrt und kosmische Angelegenheiten",
41          "episode_count": 68,
42          "episodes": [
43              {
44                  "title": "Philae",
45                  "id": 24580,
46                  "guid": "podlove-2015-04-28t23:48:53+00:00-4a76622f3e60548",
47                  "url": "http:\/\/raumzeit-podcast.de\/2015\/06\/16\/rz058-
    philae\/",
48                  "enclosure":
    "https:\/\/tracking.feedpress.it\/link\/13437\/1885139\/rz058-philae.m4a",
49                  "podcast_id": 703,
50                  "imgURL": "https:\/\/meta.metaebene.me\/media\/raumzeit\/rz058-
    philae.jpg",
51                  "pubdate": "2015-06-16T19:51:06+02:00",
52                  "duration": 5587,
```

```
53              "status": 200,
54              "num_season": 2,
55              "num_episode": 58,
56              "url_fyyd": "https:\/\/fyyd.de\/episode\/24580",
57              "description": "...",
58              "chapters": [
59                  {
60                      "start": "00:00:00.000",
61                      "start_ms": 0,
62                      "title": "Intro"
63                  }
64
65              ],
66              "content_type": "audio\/mp4"
67          }
68      ]
69    }
70  }
```

[POST /podcast/action]

Description

Performes an action to a podcast owned by you (see claim-button on every podcast page).

At the moment, 'check' is the only action to perform. This triggers an update within the next poll cycle.

Authorization

required

Parameters

- **podcast_id (required, int)** the podcast's id
- *or* **podcast_slug (alternative, string)** the podcast's slug
- **action (required, string)** the action to perform:
  - **check**

Response

returns a 204: No Content

[GET /podcasts]

Description

Gets information about all podcasts within fyyd.

Additionally you can address the resultset with {page} and {count}.

Please note that this request is the plural form of podcast, podcasts ;-)

Parameters

- **page (optional, int)** the page you want to address, default: 0
- **count (optional,int)** the page's size, default: 50

Response

The response includes data about the pagination.

```
 1  {
 2    "status": 1,
 3    "msg": "ok",
 4    "meta": {
 5        "paging": {
 6            "count": 2,
 7            "page": 0,
 8            "first_page": 0,
 9            "last_page": 7094,
10            "next_page": 1,
11            "prev_page": null
12        },
13        "API_INFO": {
14            "API_VERSION": 0.2
15        }
16    },
17    "data": [
18        {
19            "title": "Freakonomics Radio",
20            "id": 1,
21            "xmlURL": "http:\/\/feeds.feedburner.com\/freakonomicsradio",
22            "htmlURL": "http:\/\/www.wnyc.org\/articles\/freakonomics-podcast",
23            "imgURL":
    "https:\/\/media2.wnyc.org\/i\/1400\/1400\/l\/80\/1\/wn16_wnycstudios_freakonom
    ics-rev3.png",
24            "status": 390,
25            "slug": "freakonomics-radio",
26            "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/1.jpg",
27            "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/1.png",
28            "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/1.png",
29            "language": "en",
30            "lastpoll": "2017-04-16 00:34:49",
31            "generator": null,
32            "categories": [
33                52
34            ],
35            "lastpub": "2017-03-16 04:00:00",
```

```
36            "rank": 1,
37            "url_fyyd": "https:\/\/fyyd.de\/podcast\/1",
38            "description": "Have fun discovering the hidden side of everything
      with host Stephen J. Dubner, co-author of the best-selling \"Freakonomics\u201d
      books. Each week, hear surprising conversations that explore the riddles of
      everyday life and the weird wrinkles of human nature\u2014from cheating and
      crime to parenting and sports. Dubner talks with Nobel laureates and
      provocateurs, social scientists and entrepreneurs \u2014 and his
      \u201cFreakonomics\u201d co-author Steve Levitt. After just a few episodes,
      this podcast will have you too thinking like a Freak. Produced by WNYC Studios,
      home of other great podcasts such as \u201cRadiolab,\" \"Death, Sex & Money,\"
      and \"On the Media.\" ",
39            "subtitle": "Have fun discovering the hidden side of everything with
      host Stephen J. Dubner, co-author of the best-selling \"Freakonomics\u201d
      books. Each week, hear surprising conversations that explore the riddles of
      everyday life and the weird wrinkles of human nature\u2014f"
40        },
41        {
42            "title": "Skeptoid",
43            "id": 3,
44            "xmlURL": "http:\/\/skeptoid.com\/podcast.xml",
45            "htmlURL": "https:\/\/skeptoid.com",
46            "imgURL": "http:\/\/skeptoid.com\/images\/1500b.jpg",
47            "status": 304,
48            "slug": "skeptoid",
49            "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/3.jpg",
50            "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/3.png",
51            "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/3.png",
52            "language": "en",
53            "lastpoll": "2017-04-17 02:39:44",
54            "generator": null,
55            "categories": [
56                    52
57             ],
58            "lastpub": "2017-04-11 02:00:00",
59            "rank": 1,
60            "url_fyyd": "https:\/\/fyyd.de\/podcast\/3",
61            "description": "Since 2006, the weekly Skeptoid podcast has been
      taking on all the most popular myths and revealing the true science, true
      history, and true lessons we can learn from each. Free subscribers get the most
      recent 50 episodes, premium subscribers (skeptoid.com) can access the full
      archive, all ad-free.",
62            "subtitle": "A weekly look at the true facts behind urban legends and
      stories you've probably heard."
63        }
64    ]
65 }
```

Additionally links inside the header are provided:

```
Link: <https://api.fyyd.de/0.2/podcasts/1/2>; rel="next",
<https://api.fyyd.de/0.2/podcasts/0/2>; rel="first",
<https://api.fyyd.de/0.2/podcasts/7094/2>; rel="last"
```

[GET /categories]

Description

Gets the complete categories tree

Parameters

Response

```
 1  {
 2      "status": 1,
 3      "msg": "ok",
 4      "meta": {
 5          "API_INFO": {
 6              "API_VERSION": 0.2
 7          }
 8      },
 9      "data": [
10          {
11              "id": 1,
12              "slug": "arts",
13              "name": "Arts",
14              "name_de": "Kunst",
15              "subcategories": [
16                  {
17                      "id": 2,
18                      "slug": "design",
19                      "name": "Design",
20                      "name_de": "Design"
21                  },
22                  {
23                      "id": 3,
24                      "slug": "fashion-beauty",
25                      "name": "Fashion & Beauty",
26                      "name_de": "Mode & Sch\u00f6nheit"
27                  },
28                  {
29                      "id": 4,
30                      "slug": "food",
31                      "name": "Food",
32                      "name_de": "Essen"
```

```json
                },
                {
                    "id": 5,
                    "slug": "literature",
                    "name": "Literature",
                    "name_de": "Literatur"
                },
                {
                    "id": 6,
                    "slug": "performing-arts",
                    "name": "Performing Arts",
                    "name_de": "Darstellende K\u00fcnste"
                },
                {
                    "id": 7,
                    "slug": "visual-arts",
                    "name": "Visual Arts",
                    "name_de": "Bildende K\u00fcnste"
                }
            ]
        },
        {
            "id": 8,
            "slug": "business",
            "name": "Business",
            "name_de": "Wirtschaft",
            "subcategories": [
                {
                    "id": 9,
                    "slug": "business-news",
                    "name": "Business News",
                    "name_de": "Wirtschaftsnachrichten"
                },
                {
                    "id": 10,
                    "slug": "careers",
                    "name": "Careers",
                    "name_de": "Karriere"
                },
                {
                    "id": 11,
                    "slug": "investing",
                    "name": "Investing",
                    "name_de": "Geldanlage"
                },
                {
                    "id": 12,
                    "slug": "management-marketing",
                    "name": "Management & Marketing",
```

```
82                    "name_de": "Management & Markteting"
83                },
84                {
85                    "id": 13,
86                    "slug": "shopping",
87                    "name": "Shopping",
88                    "name_de": "Shopping"
89                }
90            ]
91        },
92        {
93            "id": 14,
94            "slug": "comedy",
95            "name": "Comedy",
96            "name_de": "Kom\u00f6dien",
97            "subcategories": []
98        }
99        ]
100     ]
101 }
```

[GET /category]

Description

Retrieves the podcasts inside the specified category. The categories system referres to Apple's iTunes Categories. Categories not in Apple's specification are ignored on import.

Parameters

- **category_id (required, int)** the category's id (see /categories)
- **page (optional, int)** the page you want to address, default: 0
- **count (optional,int)** the page's size, default: 50

Response

```
1  {
2    "status": 1,
3    "msg": "ok",
4    "meta": {
5        "paging": {
6            "count": 2,
7            "page": 0,
8            "first_page": 0,
9            "last_page": 117,
10           "next_page": 1,
11           "prev_page": null
```

```
12          },
13        "API_INFO": {
14            "API_VERSION": 0.2
15        }
16    },
17    "data": {
18        "category": {
19            "id": 61,
20            "slug": "professional",
21            "title": "Professional",
22            "url": "https:\/\/fyyd.de\/discover\/cat\/professional\/0"
23        },
24        "parent": {
25            "id": 57,
26            "slug": "sports-recreation",
27            "title": "Sports & Recreation",
28            "url": "https:\/\/fyyd.de\/discover\/cat\/sports-recreation\/0"
29        },
30        "podcasts": [
31            {
32                "title": "vollraute - der gladbach podcast mit der raute im
herzen -",
33                "id": 170,
34                "xmlURL": "http:\/\/feedpress.me\/vollrautepodcast",
35                "htmlURL": "http:\/\/vollraute.de",
36                "imgURL": "http:\/\/vollraute.de\/wp-
content\/cache\/podlove\/5a\/cc055669c606ec882f928be09be530\/vollraute-der-
gladbach-podcast-mit-der-raute-im-herzen_original.png",
37                "status": 390,
38                "slug": "vollraute-der-gladbach-podcast-mit-der-raute-im-herzen",
39                "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/170.jpg",
40                "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/170.png",
41                "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/170.png",
42                "language": "de",
43                "lastpoll": "2017-04-16 00:36:11",
44                "generator": "Podlove Podcast Publisher v2.4.1",
45                "categories": [
46                    52
47                ],
48                "lastpub": "2017-04-11 21:14:53",
49                "rank": 1,
50                "url_fyyd": "https:\/\/fyyd.de\/podcast\/170",
51                "description": "Ein Podcast \u00fcber die Borussia aus
M\u00f6nchengladbach.\nNie neutral, sondern immer durch die Vereinsbrille.",
52                "subtitle": "Ein Podcast \u00fcber die Borussia aus
M\u00f6nchengladbach"
53            },
54            {
55                "title": "Collinas Erben",
```

```
56              "id": 296,
57              "xmlURL": "http:\/\/fokus-fussball.de\/feed\/mp3\/",
58              "htmlURL": "http:\/\/fokus-fussball.de",
59              "imgURL": "http:\/\/fokus-fussball.de\/wp-
      content\/cache\/podlove\/5f\/716d5e4a7909b9b87b75a456524d89\/collinas-
      erben_original.jpg",
60              "status": 304,
61              "slug": "collinas-erben",
62              "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/296.jpg",
63              "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/296.png",
64              "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/296.png",
65              "language": "de",
66              "lastpoll": "2017-04-17 16:12:25",
67              "generator": "Podlove Podcast Publisher v2.3.8",
68              "categories": [
69                  52
70              ],
71              "lastpub": "2016-11-15 08:53:35",
72              "rank": 10,
73              "url_fyyd": "https:\/\/fyyd.de\/podcast\/296",
74              "description": "Fu\u00dfball ist ein einfaches Spiel \u2013
      deshalb ist es so erfolgreich: Der Ball ist rund. Das Spiel dauert um die 90
      Minuten.\n\nProblematisch wird es nur, wenn Sonderf\u00e4lle auftreten. Wenn
      ein Spieler pl\u00f6tzlich mit nacktem Oberk\u00f6rper auf den Zaun springt,
      wenn ein Engl\u00e4nder doch mal mit einem Wembley-Tor trifft oder wenn auch
      nach der zigsten Wiederholung nicht klar ist, ob es ein Foul in der 88. Minute
      war und ob es im oder vor dem Strafraum passierte.\n\nDer Schiedsrichter-
      Podcast Collinas Erben m\u00f6chte f\u00fcr mehr Fachwissen auf den
      Trib\u00fcnen und an den Empfangsger\u00e4ten sorgen. Alex Feuerherdt und Klaas
      Reese sprechen \u00fcber das Schiedsrichterwesen im Allgemeinen und die
      Spielregeln im Speziellen.",
75              "subtitle": "Der Schiedsrichter-Podcast"
76          }
77      ]
78    }
79 }
```

[GET /podcast/recommend]

Description

Gets information about what fyyd thinks, might be a good idea to listen to also. At the moment, this is based on collections information.

Parameters

- **podcast_id (required, int)** the Podcast's id

- *or* **podcast_slug (alternative, string)** the podcast's slug
- **count (optional,int)** the number of recommended podcasts, default: 10

Response

```
1    {
2        "status": 1,
3        "msg": "ok",
4        "meta": {
5            "API_INFO": {
6                "API_VERSION": 0.2
7            }
8        },
9        "data": [
10           {
11               "title": "Explikator",
12               "id": 158,
13               "xmlURL": "http:\/\/explikator.de\/feed\/podcast\/",
14               "htmlURL": "http:\/\/explikator.de\/podcast",
15               "imgURL": "http:\/\/explikator.de\/wp-
     content\/cache\/podlove\/ed\/92555a31e2f10f4922dd5b5b8b2b77\/explikator_origina
     l.png",
16               "status": 304,
17               "slug": "explikator",
18               "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/158.jpg",
19               "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/158.png",
20               "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/158.png",
21               "language": "de",
22               "lastpoll": "2017-04-23 10:13:56",
23               "generator": "Podlove Podcast Publisher v2.3.18",
24               "user_id": null,
25               "categories": [
26                   52
27               ],
28               "lastpub": "2017-04-20 00:01:21",
29               "rank": 29,
30               "url_fyyd": "https:\/\/fyyd.de\/podcast\/158",
31               "description": "Jeden Werktag 10 Minuten Podcast und ein neuer
     Musiktitel. Themen sind Wissenschaft, Bewusstsein, Geschichte, Film & TV und
     Entertainment. Unbekanntes & Unbeachtetes, Hintergrund & Oberfl\u00e4che. Mal
     ernster, mal satirischer aber immer Morgen-Kaffee-tauglich!",
32               "subtitle": "Morgenradio 2.0: Jeden Werktag 10 Minuten Unbeachtetes
     & Unbekanntes aus Gesellschaft und Kultur."
33           }
34       ]
35   }
```

[GET /podcast/latest]

Description

Returns information about the last added podcasts. Provide a count and/or an ID to be the starting point.

Parameters

- **since_id (optional, int)** the last podcast's id you know of. Get the podcasts added after that ID. Maximum: 1000 episodes.
- **count (optional, int, default: 20)** get the latest count nums of podcasts.

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "API_INFO": {
6               "API_VERSION": 0.2
7           }
8       },
9       "data": [
10          {
11              "title": "Polytox Podcast (Polytox-Podcast)",
12              "id": 49782,
13              "xmlURL": "http:\/\/polytox.org\/feed\/mp3\/",
14              "htmlURL": "http:\/\/polytox.org\/polytox-podcast",
15              "imgURL": "http:\/\/polytox.org\/wp-
    content\/cache\/podlove\/a3\/c6f4f8b44edc93e3b7b3ef2205f69c\/polytox-
    podcast_original.jpg",
16              "status": 200,
17              "slug": "polytox-podcast-polytox-podcast",
18              "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/49782.jpg",
19              "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/49782.png",
20              "smallImageURL": "https:\/\/img.fyyd.de\/pd\/small\/49782.jpg",
21              "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/49782.png",
22              "language": "de",
23              "lastpoll": "2017-11-06T20:10:05+01:00",
24              "generator": "Podlove Podcast Publisher v2.6.2",
25              "categories": [],
26              "lastpub": "2017-11-05T17:04:18+01:00",
27              "rank": null,
28              "url_fyyd": "https:\/\/fyyd.de\/podcast\/polytox-podcast-polytox-
    podcast\/0",
29              "description": "Der Podcast des Polytox Zines mit Falk Fatal. Jede
    Folge gibt es Punk, Pop & Papperlapapp rund um die Subkultur.",
30              "subtitle": "Der Subkultur-Podcast",
```

```
31              "episode_count": 15
32          }
33      ]
34 }
```

[GET /podcast/collections]

Description

Returns information about the collections this podcast is collected in.

Parameters

- **podcast_id (required, int)** the Podcast's id
- *or* **podcast_slug (alternative, string)** the podcast's slug
- **count (optional,int)** the number of recommended podcasts, default: 10
- **page (optional,int)** the page of the result, default: 0

Response

```
1
2      "status": 1,
3      "msg": "ok",
4      "meta": {
5          "paging": {
6              "count": 3,
7              "page": 0,
8              "first_page": 0,
9              "last_page": 19,
10             "next_page": 1,
11             "prev_page": null
12         },
13         "API_INFO": {
14             "API_VERSION": 0.2
15         }
16     },
17     "data": [
18         {
19             "title": "H\u00f6rt das!",
20             "id": 48,
21             "description": "",
22             "layoutImageURL": "https:\/\/fyyd.de\/images\/collection300.png?
    et=",
23             "smallImageURL": "https:\/\/fyyd.de\/images\/collection300.png?
    et=",
24             "thumbImageURL": "https:\/\/fyyd.de\/images\/collection80.png?et=",
25             "microImageURL": "https:\/\/fyyd.de\/images\/collection20.png?et=",
26             "slug": "88610109b7c51842990bcfdbc1938696",
27 }           "user_id": 1111,
```

```
28              "url":
     "https:\/\/fyyd.de\/user\/Lesefreude\/collection\/88610109b7c51842990bcfdbc1938
     696"
29          },
30          {
31              "title": "Bildung\/Schule\/Sozio",
32              "id": 64,
33              "description": "",
34              "layoutImageURL": "https:\/\/fyyd.de\/images\/collection300.png?
     et=",
35              "smallImageURL": "https:\/\/fyyd.de\/images\/collection300.png?
     et=",
36              "thumbImageURL": "https:\/\/fyyd.de\/images\/collection80.png?et=",
37              "microImageURL": "https:\/\/fyyd.de\/images\/collection20.png?et=",
38              "slug": "bildung-schule-sozio",
39              "user_id": 1135,
40              "url": "https:\/\/fyyd.de\/user\/Anke\/collection\/bildung-schule-
     sozio"
41          }
42      ]
43  }
```

Episode

Request to retrieve an episode's information

[GET /episode]

Description

Returns information about a single episode

Parameters

- **episode_id (required, int)** the episode's id

Response

```
1  {
2    "status": 1,
3    "msg": "ok",
4    "meta": {
5        "API_INFO": {
```

```
 6              "API_VERSION": 0.2
 7          }
 8      },
 9      "data": {
10          "title": "Space Hulk (Stay Forever, Folge 33)",
11          "id": 42,
12          "guid": "796a53669cb29b0504fdd4f0a3f9a41d",
13          "url":
    "http:\/\/kaliban.podspot.de\/files\/Stay_Forever_Ep33_Space_Hulk.mp3",
14          "enclosure":
    "http:\/\/kaliban.podspot.de\/files\/Stay_Forever_Ep33_Space_Hulk.mp3",
15          "podcast_id": 5,
16          "imgURL": "",
17          "pubdate": "2014-03-06 18:10:30",
18          "duration": null,
19          "url_fyyd": "https:\/\/fyyd.de\/episode\/42",
20          "description": "Christian hasst Space Hulk, Gunnar liebt es. Fight!"
21      }
22 }
```

[GET /episode/latest]

Description

Returns information about the last added episodes. Provide a count and/or an ID to be the starting point.

Parameters

- **since_id (optional, int)** the last episode's id you know of. Get the episodes added after that ID. Maximum: 1000 episodes.
- **count (optional, int, default: 20)** get the latest count nums of episodes.

Response

```
 1 {
 2     "status": 1,
 3     "msg": "ok",
 4     "meta": {
 5         "API_INFO": {
 6             "API_VERSION": 0.2
 7         }
 8     },
 9     "data": [
10         {
11             "title": "CR123 Biometrische Vollerfassung",
12             "id": 3334,
13             "guid": "http:\/\/chaosradio.ccc.de\/cr123.html",
14             "url": "http:\/\/chaosradio.ccc.de\/cr123.html",
```

```
15            "enclosure":
     "http:\/\/chaosradio.ccc.de\/archive\/chaosradio_123.mp3",
16            "podcast_id": 119,
17            "imgURL": "https:\/\/img.fyyd.de\/pd\/layout\/119.jpg",
18            "pubdate": "2007-05-06T03:20:00+02:00",
19            "duration": 10777,
20            "url_fyyd": "https:\/\/fyyd.de\/episode\/3334",
21            "description": "Fotofahndung, zentrale Fingerabdruckdatei und die
     Personenkennziffer"
22          },
23          {
24            "title": "CR122 Der Bundestrojaner",
25            "id": 3335,
26            "guid": "http:\/\/chaosradio.ccc.de\/cr122.html",
27            "url": "http:\/\/chaosradio.ccc.de\/cr122.html",
28            "enclosure":
     "http:\/\/chaosradio.ccc.de\/archive\/chaosradio_122.mp3",
29            "podcast_id": 119,
30            "imgURL": "https:\/\/img.fyyd.de\/pd\/layout\/119.jpg",
31            "pubdate": "2007-03-29T13:00:00+02:00",
32            "duration": 10603,
33            "url_fyyd": "https:\/\/fyyd.de\/episode\/3335",
34            "description": "Nie wurden Ihre Grundrechte so verletzt..."
35          }
36      ]
37  }
```

[GET /episode/curations]

Description

Returns information about the curations this episode is in.

Parameters

- **episode_id (required, int)** the episode's id
- **count (optional,int)** the number of recommended podcasts, default: 10
- **page (optional,int)** the page of the result, default: 0

Response

```
1  {
2      "status": 1,
3      "msg": "ok",
4      "meta": {
5          "paging": {
6              "count": 2,
```

```
 7              "page": 0,
 8              "first_page": 0,
 9              "last_page": 1,
10              "next_page": 1,
11              "prev_page": null
12          },
13          "API_INFO": {
14              "API_VERSION": 0.2
15          }
16      },
17      "data": [
18          {
19              "title": "Museumsfeed",
20              "id": 593,
21              "description": "Eine Zusammenstellung von Podcast-Episoden mit
    Museums- & Ausstellungsfokus: Besuche, Menschen, Projekte...",
22              "layoutImageURL":
    "https:\/\/img.fyyd.de\/curation\/layout\/593.jpg?et=",
23              "thumbImageURL": "https:\/\/img.fyyd.de\/curation\/thumbs\/593.png?
    et=",
24              "smallImageURL": "https:\/\/img.fyyd.de\/curation\/small\/593.jpg?
    et=",
25              "microImageURL": "https:\/\/img.fyyd.de\/curation\/micro\/593.png?
    et=",
26              "public": 1,
27              "slug": "museumsfeed",
28              "user_id": 1097,
29              "type": 1,
30              "categories": [
31                  52,
32                  1,
33                  15
34              ],
35              "url": "https:\/\/fyyd.de\/user\/tinowa\/curation\/museumsfeed",
36              "xmlURL": "https:\/\/feeds.fyyd.de\/tinowa\/museumsfeed"
37          },
38          {
39              "title": "Netzpolitik",
40              "id": 623,
41              "description": "Vorsicht, k\u00f6nnte Spuren von Netzpolitik
    enthalten!",
42              "layoutImageURL":
    "https:\/\/img.fyyd.de\/curation\/layout\/623.jpg?et=",
43              "thumbImageURL": "https:\/\/img.fyyd.de\/curation\/thumbs\/623.png?
    et=",
44              "smallImageURL": "https:\/\/img.fyyd.de\/curation\/small\/623.jpg?
    et=",
45              "microImageURL": "https:\/\/img.fyyd.de\/curation\/micro\/623.png?
    et=",
```

```
46          "public": 1,
47          "slug": "netzpolitik",
48          "user_id": 1000,
49          "type": 1,
50          "categories": [
51              52,
52              39,
53              27
54          ],
55          "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/netzpolitik",
56          "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/netzpolitik"
57      }
58    ]
59 }
```

Curations

All about Curations inside fyyd. Retrieve and set.

[GET /curation[/episodes]]

Description

Returns information about a single curation, given the curation's id. May append episodes, not paged at the moment. If you're authenticated and you request your own curation, non-public curations will be accessable too.

Authorization

required for getting your own curation, otherwise you only get public curations.

Parameters

- **curation_id (required, int)** the curation's id

Response

```
1 {
2   "status": 1,
3   "msg": "ok",
4   "meta": {
5       "API_INFO": {
6           "API_VERSION": 0.2
7       }
8   },
9   "data": {
```

```
10        "title": "Testfeed",
11        "id": 601,
12        "description": "nur ein Testfeed",
13        "layoutImageURL": "https:\/\/fyyd.de\/images\/curation300.png",
14        "thumbImageURL": "https:\/\/fyyd.de\/images\/curation80.png",
15        "microImageURL": "https:\/\/fyyd.de\/images\/curation20.png",
16        "public": 1,
17        "type": 1,
18        "slug": "testfeed",
19        "user_id": 1000,
20        "url": "https:\/\/fyyd.de\/user\/eazy\/curation\/testfeed",
21        "xmlURL": "https:\/\/feeds.fyyd.de\/eazy\/testfeed",
22        "episodes": [
23            {
24                "title": "A!195 \u2013 Raussismus",
25                "id": 1662345,
26                "guid": "podlove-2017-04-14t12:27:42+00:00-56a19c3a4209a8d",
27                "url": "https:\/\/aufwachen-podcast.de\/2017\/04\/14\/a195-
raussismus\/",
28                "enclosure": "https:\/\/aufwachen-
podcast.de\/podlove\/file\/3244\/s\/feed\/c\/itunes\/195raussismus.m4a",
29                "podcast_id": 95,
30                "imgURL": "",
31                "pubdate": "2017-04-14 15:05:11",
32                "duration": 16018,
33                "favedDate": "2017-04-17 16:46:39",
34                "url_fyyd": "https:\/\/fyyd.de\/episode\/1662345",
35                "description": "Freitag, 14. April 2017, 15:05 Uhr<p>Frankreich
steht seit Monaten Kopf."
36            },
37            {
38                "title": "LdN051 BVB-Bus, Steuern, Syrien, US-Au\u00dfen-
\u201cPolitik\u201d, Altmaier",
39                "id": 1661432,
40                "guid": "podlove-2017-04-14t07:57:05+00:00-5ee52bc0e03190f",
41                "url":
"https:\/\/www.kuechenstud.io\/lagedernation\/2017\/04\/14\/ldn051-bvb-bus-
steuern-syrien-us-aussen-politik-altmaier\/",
42                "enclosure":
"https:\/\/www.kuechenstud.io\/lagedernation\/podlove\/file\/1795\/s\/feed\/c\/
mp3\/LdN051.mp3",
43                "podcast_id": 44980,
44                "imgURL": "",
45                "pubdate": "2017-04-14 10:07:59",
46                "duration": 3945,
47                "favedDate": "2017-04-17 16:45:58",
48                "url_fyyd": "https:\/\/fyyd.de\/episode\/1661432",
49                "description": "<p>Liebe Lage-Gemeinde!<\/p>\n<p>P\u00fcnktlich
zu Ostertagen die neue Lage."
```

```
50            }
51        ]
52    }
53 }
```

[POST /curation]

Description

Creates or modifies a curation.

Authorization

required

Parameters

- **curation_id (opt, int)** when set, an existing curation is modified
- **title (req when created, string)** the title to set for this curation
- **description (opt, string)** the description for this curation
- **slug (opt, string)** the slug to set for this curation. If not set while creation, the title will be slugified. *When set, but empty while modifying a curation, also a slugified title will overwrite the existing slug*.
- **public (opt, int, def: 0)** defines, if this curation will be set to public and thus accessable for all.
- **image (opt, binary)** upload an optional image for the curation. Use multipart/form-data ([RFC 2388](#)) and provide a content-type.
- **categories (opt, array)** A json encoded array of category IDs. These are the same you get by requesting /categories

Response

Creating and mofiying a curation returns the complete object of the curation as in GET /curation.

[POST /curation/delete]

Description

Deletes a curation. Please note: there's no coming back, no questions, no backup. It's deleted!

Please note also: You cannot delete your very own personal curation, it's tied to your account.

Authorization

required

Parameters

- **curation_id (req, int)** the id of the curation you want to delete

Response

returns 204: No content

[POST /curate]

Description

Adds OR removes the episode identified by episode_id into the curation identified by curation_id. The curation must be owned by the user identified by the accesstoken. The state of the episode inside this curation toggles with each call, the state is returned inside response's data section.

If you want to enforce a state without knowing what state the episode had before, use **force_state**.

Authorization

required

Parameters

- **curation_id (required, int)** the curation's id
- **episode_id (required, int)** the episode's id
- **why (optional, string)** a text to indicate, why you curate this episode
- **force_state (optional, bool)** enforce adding or removing the episode from the curation

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "API_INFO": {
6               "API_VERSION": 0.2
7           }
8       },
9       "data": {
10          "state": 1,
11          "text": "added AdH039: Singende Nonne to TestKuration"
12      }
13  }
```

[GET /curate]

Description

Retuns the state of the curation of an episode in a curation.

Authorization

required

Parameters

- **curation_id (required, int)** the curation's id
- **episode_id (required, int)** the episode's id

Response

```
1  {
2      "status": 1,
3      "msg": "ok",
4      "meta": {
5          "API_INFO": {
6              "API_VERSION": 0.2
7          }
8      },
9      "data": {
10         "state": 1
11     }
12 }
```

[GET /category/curation]

Description

Retrieves the curations inside the specified category. The categories system referres to Apple's iTunes Categories.

Please note: Curations' categories calculate automatically by the podcasts' categories of the curated episodes. In future, those categories also could get set manually by the user if desired. At the moment the categories are calculated and thus could change with every curated episode.

Additionally you can only find public curations with that function.

Parameters

- **category_id (required, int)** the category's id (see /categories)
- **page (optional, int)** the page you want to address, default: 0
- **count (optional,int)** the page's size, default: 50

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "paging": {
6               "count": 50,
7               "page": 0,
8               "first_page": 0,
9               "last_page": 0,
10              "next_page": null,
11              "prev_page": null
12          },
13          "API_INFO": {
14              "API_VERSION": 0.2
15          }
16      },
17      "data": {
18          "category": {
19              "id": 5,
20              "slug": "literature",
21              "title": "Literatur",
22              "url": "https:\/\/fyyd.de\/discover\/cat\/literature\/0"
23          },
24          "parent": {
25              "id": 1,
26              "slug": "arts",
27              "title": "Kunst",
28              "url": "https:\/\/fyyd.de\/discover\/cat\/arts\/0"
29          },
30          "curations": [
31              {
32                  "title": "Krieg der Poesie",
33                  "id": 569,
34                  "description": "Alle Episoden aus dem Krieg der Poesie der
    Geschichtenkapsel",
35                  "layoutImageURL":
    "https:\/\/img.fyyd.de\/curation\/layout\/569.jpg?et=",
36                  "thumbImageURL":
    "https:\/\/img.fyyd.de\/curation\/thumbs\/569.png?et=",
37                  "smallImageURL":
    "https:\/\/img.fyyd.de\/curation\/small\/569.jpg?et=",
38                  "microImageURL":
    "https:\/\/img.fyyd.de\/curation\/micro\/569.png?et=",
39                  "public": 1,
40                  "slug": "kriegderpoesie",
41                  "user_id": 1016,
42                  "type": 1,
43                  "categories": [
44                      1,
```

```
45                        5,
46                        52
47                    ],
48                    "url":
     "https:\/\/fyyd.de\/user\/Herrvonspeck\/curation\/kriegderpoesie",
49                    "xmlURL":
     "https:\/\/feeds.fyyd.de\/Herrvonspeck\/kriegderpoesie"
50            },
51            {
52                    "title": "Jahresendfestkalender 2016",
53                    "id": 1008,
54                    "description": "Alle Geschichten, Gedichte, H\u00f6rspiele und
     Wortklaubereien, die im Rahmen des Jahresendfestkalenders 2016 erschienen.",
55                    "layoutImageURL":
     "https:\/\/img.fyyd.de\/curation\/layout\/1008.jpg?et=",
56                    "thumbImageURL":
     "https:\/\/img.fyyd.de\/curation\/thumbs\/1008.png?et=",
57                    "smallImageURL":
     "https:\/\/img.fyyd.de\/curation\/small\/1008.jpg?et=",
58                    "microImageURL":
     "https:\/\/img.fyyd.de\/curation\/micro\/1008.png?et=",
59                    "public": 1,
60                    "slug": "jefk2016",
61                    "user_id": 1016,
62                    "type": 1,
63                    "categories": [
64                        1,
65                        5,
66                        52
67                    ],
68                    "url":
     "https:\/\/fyyd.de\/user\/Herrvonspeck\/curation\/jefk2016",
69                    "xmlURL": "https:\/\/feeds.fyyd.de\/Herrvonspeck\/jefk2016"
70            },
71            {
72                    "title": "alboh - h\u00f6rspiele",
73                    "id": 1072,
74                    "description": "",
75                    "layoutImageURL": "https:\/\/fyyd.de\/images\/curation300.png?
     et=",
76                    "thumbImageURL": "https:\/\/fyyd.de\/images\/curation80.png?
     et=",
77                    "smallImageURL": "https:\/\/fyyd.de\/images\/curation300.png?
     et=",
78                    "microImageURL": "https:\/\/fyyd.de\/images\/curation20.png?
     et=",
79                    "public": 1,
80                    "slug": "073e20f0dd79291717527c2405bad9ba",
81                    "user_id": 1409,
```

```
82                "type": 1,
83                "categories": [
84                    1,
85                    5,
86                    39
87                ],
88                "url":
     "https:\/\/fyyd.de\/user\/alboh\/curation\/073e20f0dd79291717527c2405bad9ba",
89                "xmlURL":
     "https:\/\/feeds.fyyd.de\/alboh\/073e20f0dd79291717527c2405bad9ba"
90            }
91        ]
92    }
93 }
```

Collections

This part is all about collections at fyyd. The content part of collections are podcasts.

[GET /collection[/podcasts]]

Description

Returns information about a single collection, given the collection's id. May append podcasts, not paged at the moment.

Parameters

- **collection_id (required, int)** the collection's id

Response

```
1  {
2    "status": 1,
3    "msg": "ok",
4    "meta": {
5        "API_INFO": {
6            "API_VERSION": 0.2
7        }
8    },
9    "data": {
10       "title": "Netzpolitik",
```

```json
      "id": 1,
      "description": "(Netz)politische Podcasts",
      "layoutImageURL": "https:\/\/img.fyyd.de\/collection\/layout\/1.jpg",
      "thumbImageURL": "https:\/\/img.fyyd.de\/collection\/thumbs\/1.png",
      "microImageURL": "https:\/\/img.fyyd.de\/collection\/micro\/1.png",
      "slug": "netzpolitik",
      "user_id": 1000,
      "url": "https:\/\/fyyd.de\/user\/eazy\/collection\/netzpolitik",
      "podcasts": [
          {
              "title": "#heiseshow (Audio)",
              "id": 5037,
              "xmlURL": "https:\/\/www.heise.de\/heiseshow.rss",
              "htmlURL": "http:\/\/www.heise.de\/heiseshow.rss",
              "imgURL":
"https:\/\/www.heise.de\/imgs\/02\/5\/1\/1\/7\/THUMB_heiseshow_quadrat-
b3e99ad376d2732e.jpg",
              "status": 304,
              "slug": "heiseshow-audio",
              "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/5037.jpg",
              "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/5037.png",
              "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/5037.png",
              "language": "de",
              "lastpoll": "2017-04-17 05:20:40",
              "generator": null,
              "categories": [
                  52
              ],
              "lastpub": "2017-04-13 02:00:00",
              "rank": 1,
              "url_fyyd": "https:\/\/fyyd.de\/podcast\/5037",
              "description": "Immer donnerstags um 16 Uhr diskutieren wir in
der #heiseshow mit G\u00e4sten \u00fcber aktuelle Ereignisse aus der Hightech-
Welt und der Netzpolitik",
              "subtitle": "Jede Woche Donnerstag live (Audio)"
          },
          {
              "title": "Breitband - Medien und digitale Kultur (ganze Sendung)
- Deutschlandradio Kultur",
              "id": 503,
              "xmlURL": "http:\/\/www.deutschlandradiokultur.de\/podcast-
breitband-medien-und-digitale-kultur-ganze-sendung.1552.de.podcast.xml",
              "htmlURL": "http:\/\/www.deutschlandradio.de\/weiterleitung-
breitband-de.233.de.html",
              "imgURL":
"http:\/\/www.deutschlandradiokultur.de\/media\/files\/2\/258cfe6db750912b0bb36
410d2fdf775v2.jpg",
              "status": 390,
```

```
50              "slug": "breitband-medien-und-digitale-kultur-ganze-sendung-
       deutschlandradio-kultur",
51              "layoutImageURL": "https:\/\/img.fyyd.de\/pd\/layout\/503.jpg",
52              "thumbImageURL": "https:\/\/img.fyyd.de\/pd\/thumbs\/503.png",
53              "microImageURL": "https:\/\/img.fyyd.de\/pd\/micro\/503.png",
54              "language": "de",
55              "lastpoll": "2017-04-17 17:12:06",
56              "generator": null,
57              "categories": [
58                  52
59              ],
60              "lastpub": "2017-04-15 13:05:01",
61              "rank": 14,
62              "url_fyyd": "https:\/\/fyyd.de\/podcast\/503",
63              "description": "Feed provided by Deutschlandradio. Click to
       visit.",
64              "subtitle": "Die Beitr\u00e4ge zur Sendung"
65          }
66      ]
67  }
68 }
```

[POST /collection]

Description

Creates or modifies a collection.

Authorization

required

Parameters

- **collection_id (opt, int)** when set, an existing collection is modified
- **title (req when created, string)** the title to set for this collection
- **description (opt, string)** the description for this collection
- **slug (opt, string)** the slug to set for this collection. If not set while creation, the title will be slugified. *When set, but empty while modifying a collection, also a slugified title will overwrite the existing slug.*
- **public (opt, int, def: 0)** defines, if this collection will be set to public and thus accessable for all.
- **image (opt, binary)** upload an optional image for the collection. Use multipart/form-data ([RFC 2388](#)) and provide a content-type.

Response

Creating and mofiying a collection returns the complete object of the collection as in GET /collection.

[POST /collection/delete]

Description

Deletes a collection. Please note: there's no coming back, no questions, no backup. It's deleted!

Authorization

required

Parameters

- **collection_id (req, int)** the ID of the collection you want to delete

Response

returns 204: No content

[POST /collect]

Description

Adds OR removes the podcast identified by podcast_id into or from the collection identified by collection_id. The collection must be owned by the user identified by the accesstoken. The status of the podcast inside this collection toggles.

If you want to enforce a state without knowing what state the podcast had before, use **force_state**.

Authorization

required

Parameters

- **collection_id (required, int)** the collection's id
- **podcast_id (required, int)** the podcasts's id
- **force_state (optional, bool)** enforce adding or removing the podcast from the collection.

Response

```
1  {
2      "status": 1,
3      "msg": "ok",
4      "meta": {
5          "API_INFO": {
6              "API_VERSION": 0.2
7          }
8      },
9      "data": {
10         "state": 1,
11         "text": "added Auf dem Holzweg to Gamechanger"
12     }
13 }
```

[GET /collect]

Description

Retuns the state of the collection of a podcast in a collection.

Authorization

required

Parameters

- **collection_id (required, int)** the collection's id
- **podcast_id (required, int)** the podcast's id

Response

```
1  {
2      "status": 1,
3      "msg": "ok",
4      "meta": {
5          "API_INFO": {
6              "API_VERSION": 0.2
7          }
8      },
9      "data": {
10         "state": 1
11     }
12 }
```

Search

This request is the only one in this section for now. The search API will grow in the near future, drafts are already written on how to manage this important part of fyyd's API.

[GET /search/episode]

Description

This request tries to find an episode inside fyyd's database, matching any or some of a set of given criteria.

This reflects Skoon's need to find an episode's id to add to the user's curation.

Parameters

- **title (optional, string)** the episode's title. Search might use parts of the string to find the episode.
- **guid (optional, string)** the episode's GUID as stated inside the podcasts feed.
- **podcast_id (optional, int)** the podcast's id in fyyd's database.
- **podcast_title (optional, string)** the podcast's title. Search might use parts of the string to find the podcast.
- **pubdate (optional, string)** the pubDate as stated inside the podcasts feed.
- **duration (optional, int)** the duration of the episode in seconds.
- **url (optional, string)** the episode's url as stated inside the podcast's feed.
- **term (optional,string)** a search term to find inside the episodes.

Please note: title, guid, pubdate, duration, url and term add episodes together. Think of a logical OR.

In contrast to that, podcast_id and podcast_title restrict all episodes to podcasts matching to one of podcast_id or podcast_title.

- **count (optional, int, default: 10)**

Response

```
1   {
2     "status": 1,
3     "msg": "ok",
4     "meta": {
5        "API_INFO": {
6           "API_VERSION": 0.2
7        }
8     },
9     "data": [
10       {
11          "title": "FS195 Dein Butler ist ein Zombie",
```

```
12          "id": 1597636,
13          "guid": "podlove-2017-03-22t23:23:57+00:00-f1bfca935c876bc",
14          "url": "http:\/\/freakshow.fm\/fs195-dein-butler-ist-ein-zombie",
15          "enclosure":
   "https:\/\/tracking.feedpress.it\/link\/13453\/5554460\/fs195-dein-butler-ist-
   ein-zombie.m4a",
16          "podcast_id": 85,
17          "imgURL": "",
18          "pubdate": "2017-03-23 02:16:34",
19          "duration": 14632,
20          "url_fyyd": "https:\/\/fyyd.de\/episode\/1597636",
21          "description": "\n<p>Open Source Development \u2014 Ultraschall
   \u2014 Podlove Publisher \u2014 Podcast Empanzipation \u2014 Thunderbolt..."
22        }
23    ]
24  }
```

[GET /search/podcast]

Description

This request tries to find a podcast inside fyyd's database, matching any or some of a set of given criteria.

Parameters

- **title (optional, string)** the podcasts's title. Search might use parts of the string to find the podcast.
- **url (optional, string)** the podcast's url as stated inside the podcast's feed.
- **term (optional,string)** a search term to find inside the podcast.

Please note: title, url and term add episodes together. Think of a logical OR.

- **count (optional, int, default: 10)**

Response

[   ]

[GET /search/curation]

Description

This request tries to find a curation inside fyyd's database, matching any or some of a set of given criteria.

Parameters

- **category_id (optional, int)** the id of a podcast category that this curation belongs to

- **term (optional,string)** a search term to find inside the curation.

- **count (optional, int, default: 10)**

Response

```

```

[GET /search/user]

Description

This request tries to find users based on nick and/or fullname.

Parameters

- **nick (optional, string)** a nick or part of it
- **fullname (optional,string)** a full name or part of it

- **count (optional, int, default: 10)**

Response

```

```

Feature

This will be an additional, a higher level interface to discover podcasts. You could take all the data the podcasts API deliveres, ranks including, to calculate your own hot and wanted stuff. But why not take my data? Here we are with the first request: Hot podcasts.

[GET /feature/podcast/hot]

Description

This request delivers the hottest podcasts in fyyd's directory. This are NOT charts. This one is about the most active (listener and producer) podcasts of the last eight days. Hotness values are created once a day for all languages with 50 or more podcasts listed.

Parameters

- **count (optional, int, default: 10)**
- **language (optional, var, default: none)**

Response

```
1   {
```

```json
    "status": 1,
    "msg": "ok",
    "meta": {
        "API_INFO": {
            "API_VERSION": 0.2
        },
        "SERVER": "195.201.115.6",
        "duration": 1
    },
    "data": [
        {
            "title": "\u00d61 Betrifft Geschichte",
            "id": 41507,
            "xmlURL":
"http:\/\/static.orf.at\/podcast\/oe1\/oe1_geschichte.xml",
            "htmlURL": "http:\/\/oe1.orf.at\/podcast\/",
            "imgURL":
"http:\/\/files.orf.at\/podcast\/oe1\/img\/oe1_geschichte.png",
            "status": 200,
            "slug": "oe1-betrifft-geschichte",
            "language": "de",
            "lastpoll": "2018-05-20T18:37:18+02:00",
            "generator": "ORF.at Publisher for Podcast Services v1.0",
            "categories": [
                48,
                51
            ],
            "lastpub": "2018-05-18T00:00:00+02:00",
            "rank": 138,
            "url_fyyd": "https:\/\/fyyd.de\/podcast\/oe1-betrifft-
geschichte\/0",
            "description": "Ob zu Rittern oder Terroristen - Expertinnen und
Experten werden so geschickt gefragt, dass sich gut und gerne f\u00fcnf Minuten
zuh\u00f6ren l\u00e4sst, wie sich eine Antwort entfaltet. Von Montag bis
Freitag.",
            "subtitle": "Fachleute erz\u00e4hlen",
            "episode_count": 508
        },
        {
            "title": "Geschichtenkapsel",
            "id": 5095,
            "xmlURL": "http:\/\/geschichtenkapsel.de\/feed\/mp3\/",
            "htmlURL": "http:\/\/geschichtenkapsel.de",
            "imgURL": "http:\/\/geschichtenkapsel.de\/wordpress\/wp-
content\/uploads\/2018\/04\/KapsleLogo_Geschichtenkapsel-1024x1024.png",
            "status": 200,
            "slug": "geschichtenkapsel",
            "language": "de",
            "lastpoll": "2018-05-20T18:38:08+02:00",
```

```
44              "generator": "Podlove Podcast Publisher v2.7.6",
45              "categories": [
46                  1,
47                  5,
48                  52,
49                  21,
50                  24
51              ],
52              "lastpub": "2018-05-20T13:37:39+02:00",
53              "rank": 94,
54              "url_fyyd": "https:\/\/fyyd.de\/podcast\/geschichtenkapsel\/0",
55              "description": "L\u00e4sst du dir gerne Erz\u00e4hlungen,
    Kurzgeschichten, M\u00e4rchen oder Fabeln ins Ohr s\u00e4useln? Dann bist du
    hier richtig! In diesem Podcast gibt es selbstgeschriebene Geschichten, die du
    sonst nirgendwo findest! \u00d6ffne die Geschichtenkapsel und lasse dir
    fabelhafte Fantastereien, grausigen Grusel und wilde Worte um die Ohren
    fliegen!",
56              "subtitle": "Lass dir was erz\u00e4hlen",
57              "episode_count": 156
58          }
59      ]
60  }
```

[GET /feature/podcast/hot/languages]

Description

As described above, only for all languages with 50 or more podcasts listed, hotness values are calculated. To get these languages, take this request.

Parameters

Response

```
1   {
2       "status": 1,
3       "msg": "ok",
4       "meta": {
5           "API_INFO": {
6               "API_VERSION": 0.2
7           },
8           "SERVER": "195.201.115.6",
9           "duration": 0
10      },
11      "data": [
```

```
12          "en",
13          "fr",
14          "ru",
15          "it",
16          "de",
17          "es",
18          "nl"
19       ]
20   }
```

3rd party interfaces

As this API grows over time, others have made great efforts to create implementations for several languages and systems.

FyydKit

FyydKit is an implementation for Apple's swift. Stefan Trauth created it while developing Skoon, an iOS app for browsing and administrating fyyd's curations. FyydKit is up to date with the API's v0.2.

fyydAPI

fyydAPI is another switft implemenation but for v0.1. Jeanette Müller created it for her great podcatcher Podcat. You can manage and browse your curations and collections inside podcat.

fyyd_ex

fyyd_ex is an implementation for Elixir. It's WIP at the moment, but looks great to me. Well... I don't have the slightes idea of Elixir, but it really looks great, what optikfluffel (remember superfav.de?) did there :)