
gpodder.net Feed-Service Documentation

Release 0.1

Stefan Kögl

February 02, 2014

1	Contents	3
1.1	REST API	3
1.2	Example Requests	5
1.3	Public instances	6
1.4	Source Code	6

The gpodder.net Feed-Service is a webservice for parsing podcast feeds and returning the simplified contents as [JSON](#).

1.1 REST API

1.1.1 Requests

Parameters to `/parse` (either GET or POST as `application/x-www-form-urlencoded`)

url The URL of the feed that should be parsed (required). This parameter can be repeated multiple times. The values can be URL-encoded.

inline_logo If set to 1, the (unscaled) logos are included in the response as data URIs (default 0).

scale_logo If `inline_logo` is set to 1, scales the included logo down to the given size. The resulting image is fitted into a square with the given side-length. If the given size is greater than the original size, the image won't be scaled at all.

logo_format If `inline_logo` is set to 1, the inlined image is converted to the specified format (either `png` or `jpeg`). If this option is not used, the original format is preserved.

process_text Is used to remove HTML from texts. Can be either `none` (does nothing, default if omitted), `strip_html` (removes HTML and inserts newlines, bullet points, etc) or `markdown` (converts HTML to Markdown).

use_cache Feeds are cached by the service according to the feed's caching headers. If `use_cache` is set to 1 (default) feeds are retrieved from the cache if possible. If set to 0, feeds are always fetched from their URL. Do not use 0 as a default value in your application.

Headers to `/parse`

If-Modified-Since Time when all requested feeds have been accessed the last time. The response will only contain podcasts that have been modified in the meantime.

User-Agent Clients should send a descriptive `User-Agent` string. In case of abuse of the service, misbehaving and/or generic user-agents might be blocked.

Accept Clients should send `Accept: application/json` to indicate that they are prepared to receive JSON data. If you send a different `Accept` header, you will receive a HTML formatted response.

Accept-Encoding Include `gzip` in both headers to ensure `gzip` compression.

1.1.2 Responses

Each response contains a list of feeds, at least one for each `url-Parameter`. HTTP-Redirects are followed automatically (this is reflected in the `urls` field). **RSS-Redirects** are followed by additionally including the new feed in the response.

Each feed contains

title the title of the feed

link the feeds website

description a description of the feed, potentially including HTML characters

subtitle a short subtitle of the feed, potentially including HTML characters

author the feed's author

language the feed's language

urls the redirect-chain of the URL passed in the `url` parameter. This can be used to match the requested URLs to the entries in the response. A permanent redirect is not included here but given in the `new_location` field, as it indicates that the client should update the feed's location.

new_location the referred to location, if the feed uses a permanent HTTP redirect or **RSS-Redirects**. The new location will also be fetched, parsed and included in the response

logo the URL of the feed's logo

logo_data the feed's logo as a **data URI**, if `inline_logo` has been used. To save bandwidth, the logo is not included if it changed since the date sent in `If-Modified-Since`

content_types the content types of the feed, either `audio`, `video` or `image`

hub the endpoint URL of the **hub** through which the feed is published

errors a dictionary of occurred errors, where the key contains an error code and the value a string representation.

warnings a dictionary of warnings. The key contains an warning code and the value a string representation.

http_last_modified the Unix timestamp of the last modification of the feed (according to the HTTP header).

http_etag the HTTP `E-Tag` of the feed

license The URL of the license under which the podcast is published

episodes the list of episodes

Episodes

Each episode contains

guid an unique endentifier for the episode (provided by the feed in the `GUID` property)

title the title of the episode

short_title the non-repetitive part of the episode title. If an episode number is found, it is also removed and provided separately.

number the episode number which is parsed from the title

description the description of the episode, potentially including HTML characters

subtitle a short subtitle of the episode, potentially including HTML characters

link the website link for the episode

released the Unix timestamp of the episode's release

author the episode's author

duration the episode's duration in seconds

language the episode's language

license The URL of the license under which the episode is published

files a list of all files linked by the episode. Each files is represented by an object containing `urls`, `filesize` (in Bytes) and `mimetype`.

Current Error Codes

fetch-feed The feed could not be retrieved. The URL is given in the `urls` list

Current Warning Codes

fetch-logo The feed's logo could not be retrieved. Its URL is given in the `logo` field

hub-subscription An error occured while subscribing to the feed's hub for instant updates.

Headers

Last-Modified The earliest of the `Last-Modified` values of the requested podcast feeds. This value can be used in the `If-Modified-Since` parameter to subsequent requests. This header is not sent for the HTML formatted response.

Content-Type `application/json` if your request contains `Accept: application/json`, otherwise the response will contain the HTML representation with `text/html`.

Content-Encoding `gzip` if the response is compressed. See `Accept-Encoding` for details.

Vary Contains the request headers for which the response can vary. Currently this is `Accept`, `User-Agent`, `Accept-Encoding`.

1.2 Example Requests

Some example requests

- `http://feeds.gpodder.net/parse?url=http://feeds.feedburner.com/linuxoutlaws&inline_logo=1&scale_logo=30`
- `http://feeds.gpodder.net/parse?url=http://youtube.com/rss/user/TEDtalksDirector/videos.rss`
- `http://feeds.gpodder.net/parse?url=http://soundcloud.com/scheibosan`
- `http://feeds.gpodder.net/parse?url=http://onapp1.orf.at/webcam/fm4/fod/soundpark.xspf`
- `http://feeds.gpodder.net/parse?url=http://leo.am/podcasts/floss&url=http://feeds.twit.tv/floss_video_large`
- `http://feeds.gpodder.net/parse?url=http://www.dancarlin.com/cswdc.xml&process_text=strip_html`
- `http://feeds.gpodder.net/parse?url=http://feeds.feedburner.com/linuxoutlaws`

For executing requests from the commandline you can use `curl`.

```
SERVER=http://feeds.gpodder.net/parse
curl --header "Accept: application/json" "$SERVER?url=http://feeds.feedburner.com/linuxoutlaws&inlin
curl --header "Accept: application/json" "$SERVER?url=http://youtube.com/rss/user/TEDtalksDirector/
```

1.3 Public instances

A public instance of the gpodder.net Feed-Service is deployed at feeds.gpodder.net. If you intend to use this instance of the webservice in your application, please contact stefan@skoegl.net beforehand.

1.4 Source Code

The sourcecode is available from <https://github.com/gpodder/mygpo-feedservice> under the terms of the [AGPLv3](#). Please report bugs on the [bugtracker](#) and mention the exact URL of your failed request.