Open Podcast API

A NEW API SPECIFICATION ENABLING END-USERS TO SYNC EPISODE LISTENING STATUSES & MORE

The problem

Existing de-facto standard

- is no longer actively maintained
- has technical issues & challenges, e.g.:
  - episode identification (based on media URL; not unique in RSS wild west)
  - feed duplication (feed URL updates not handled adequately)
- is effectively used as centralised service
  - overload on server = users facing errors in their apps

The solution

A new API that builds upon the existing solution in a more extensible and standards-compliant way

END-USERS

- Synchronise subscriptions, listening progress, favourites, queues, and more
- Connect all compliant apps and online services
- Switch between providers without losing any information

DEVELOPERS

- Clear and comprehensive documentation for features and behaviours
- Reliable specs that are easy to implement
- Feature-complete specs that are decentralisation-ready

The approach

- Building a new API inspired by the gPodder.net API specs
  - gPodder compatibility is not a focus
- Following OpenAPI specification
  - Plans for libraries via CI for quick integration (Java, Typescript, PHP, C++, Dart)
- RSS is single source of truth, wherever possible
  - E.g. don't sync episode titles
  - E.g. do create & sync true GUID
  - Podcasting 2.0 ready

The technical challenges – Episode identification

Issue:

- Hash of combination of `<guid>`, `<link>` and `<enclosure>`?
- Former not always ‘globally unique’ (even within feed), latter not always stable (e.g. corrected audio files)

Solution:

- GUID by ‘first discoverer’ of the episode
- Sync first, post later
- Deduplication responsibility of client

More considerations in meeting notes

The technical challenges – Feature compatibility

Issue:

- Cannot expect all connecting apps/services to support all end-points & calls, because of different feature sets

Solution:

- Core: every user-facing (web) application expected to offer
- Optional: syncing of data if server supports it
- Documented in the API, core mandatory for ‘compliance’
- Mapping of capabilities to be done

The endpoints

✅ Subscriptions (add, get all/single, update, delete, deletion status)

✅ Versioning (get supported versions, get supported capabilities)

🔍 Episodes (add, get all/single, update, …: GUID, status, playback position, total time played, last played, metadata changed, …)

🔮 Settings

🔮 Search

🔮 Discovery

🔮 Ratings/Reviews

The interested projects

AntennaPod

Funkwhale

Kasts

Podfriend

gPodder Sync for Nextcloud

Musicpod

The next steps

- Episodes end-point discussions
- Authentication discussions
- New website (easier navigation of specs & end-points)
- Mapping of features across apps (to be supported by API specs)
- ‘Beta implementation’ in a few apps
- Reference server implementation

The contact details

https://github.com/OpenPodcastAPI

https://OpenPodcastAPI.org

#OpenPodcastAPI:matrix.org

The questions

The contact details

https://github.com/OpenPodcastAPI

https://OpenPodcastAPI.org

#OpenPodcastAPI:matrix.org