streamABC Documentation

Release 1.0.0

streamABC

May 17, 2019

Distributors

1 Logimporter **3**
- 1.1 Installation . . . 3
- 1.2 Tail-Mode . . . 3
- 1.3 Single File Mode . . . 4

2 streamABC API **5**

3 streamABC Skip API **7**
- 3.1 URL parameters . . . 7
- 3.2 Initial start of a skippable stream . . . 7
- 3.3 Skip to next element . . . 8
- 3.4 Tune back to live stream . . . 8

4 streamABC Websockets API **9**
- 4.1 Metadata Events for Users Subscription . . . 9
- 4.2 Metadata Events for Channel Subscription . . . 9
- 4.3 Metadata Events for Station Subscription . . . 10

5 SDK for Playerservices **11**

6 SDK for Radio **13**

7 Country-specific parts for costumers **15**

8 Indices and tables **17**

The streamABC documentation is organized into several sections:

- *Distributors*
- *API Documentation*
- *Player-SDK Documentation*
- *Country-specific parts for costumers*

(more coming soon)

CHAPTER 1

Logimporter

The streamABC Logimporter sends logs from Icecast, Shoutcast and AIS servers to our streamABC infrastructure. This is necessary to generate statistics for listeners, create AGMA log files and other statistical analyzations.

You can use Logimporter to directly stream log-data to streamABC ("tail-mode") and to send a single file at once.

Logimporter can work with log-files from the following streaming servers:

- Icecast
- Shoutcast 1
- Windows Media Server
- AIS

1.1 Installation

The Logimporter comes as a single static binary that you can use without further installation. Just download the file and make it executable.

Get in contact with streamABC support to get your installation files.

Logimporter is available for Linux, Windows, macOS and BSD.

Logimporter is controlled with command lines parameters. To get a list of all parameters execute this:

```
./logimporter --help
```

1.2 Tail-Mode

This is the most common use case. In this mode Logimporter works like Unix *tail* command. Tail mode starts at the end of the file. New lines that are added to a file are parsed and sent to streamABC in real-time. Any lines that still exist in the file are not transmitted. The program stays in foreground until you end it with CTRL-C.

```
./logimporter --dbuser=DBUSER --dbpass=DBPASS --dbname=DBNAME --dbhost=DBHOST --
↪logtype=icecast --tail ./logs/access.log
```

Note: All values for data access are provided by streamABC when you need it.

Some log-types like Shoutcast need additional parameters. For more information see below.

1.3 Single File Mode

In this mode Logimporter send the content of one file. All parameters work as in tail-mode. You just need to omit the *–tail* flag.

```
./logimporter --dbuser=DBUSER --dbpass=DBPASS --dbname=DBNAME --dbhost=DBHOST --
↪logtype=icecast ./logs/access.log
```

Note: All values for data access are provided by streamABC when you need it.

The program show the import progress and automatically ends if the file is finished.

For more information, please open a ticket:

Visit our company website:

CHAPTER 2

streamABC API

The streamABC API enables you to interact with several streamABC services.

CHAPTER 3

streamABC Skip API

The streamABC Skip API controls skips in streams using dedicated URL GET parameters appended to a streamABC stream URL.

Note: The channel behind the stream URL has to be skip enabled in the streamABC console backend to make use of this Skip API.

Example streamURL: http://sabc-test.stream.vip/qc/mp3-256/ (or https://sabc-test.stream.vip/qc/mp3-256/)

3.1 URL parameters

Name	Explanation	Example
sabcsid	Uinique listener ID (UUID)	fb3c7ba352e3
skip	Session-ID for actual listener or live for livestream	e1d4c7a3-cae1-4a74-a489-7b41648bd9e7
skip2	Position-ID of last item	7648

The `sabcsid` paramter is mandantory. It has to be unique and always the same for a specific user.

If you are working on a web player you can use our streamABC Javascript-SDK to generate this listener id and decorate an existing stream URL.

3.2 Initial start of a skippable stream

Every listener session has to be started with an valid listener id parameter `sabcsid` like so: http://sabc-test.stream.vip/qc/mp3-256/?sabcsid=fb3c7ba352e3

This starts a default live stream session but prepares our backend system to upcoming skips of this user.

3.3 Skip to next element

Start a new listener session with the same stream url like before but add a valid `skip` and `skip2` paramter: http://sabc-test.stream.vip/qc/mp3-256/?sabcsid=fb3c7ba352e3&skip=e1d4c7a3-cae1-4a74-a489-7b41648bd9e7&skip2=7648

This starts a a new user stream session starting with the next song after position defined with `skip2`.

You get the required values for `skip` and `skip2` over a Websocket connection to our player services API as described here:

See also:

Websockets Documentation

With a subscription to `/metadata/{listenerid}` you get all metadata events in real-time.

If it's not possible to open a websockets connection in addition to the stream playback you can generate your own session id and send this value as *skip* parameter and leave *skip2* blank. Bear in mind that this comes with some drawbacks: A stream start without a valid session id from our streamABC backend will result in slightly longer wait times until audio begins. In addition you can only skip to the next element in the playlist.

A websockets connection on the other hand gives you some more useful data. You get the users metadata (artist, song) of the element playing right now and the next element as long as cover art and some status information. You also get information about whether the actual element is an advertisement and how long it plays.

3.4 Tune back to live stream

Start a new listener session with the same stream url like before but add *live* as *skip* parameter: http://sabc-test.stream.vip/qc/mp3-256/?sabcsid=fb3c7ba352e3&skip=live

You can also just omit all skip parameters to start a livestream. But if the disconnect time is too short our backend tries to re-connect you listener to his already running skip stream. A *skip=live* forces a live stream start.

CHAPTER 4

streamABC Websockets API

This documentation describes the use of the low-level Websockets API as part of our PlayerServices API.

If possible use a higher-level SDK that abstracts all steps. Currently, a Javascript Version is available here: https://github.com/streamABC/api-player

You find all detailed docs for transferred messages here: https://github.com/streamABC/api-player/blob/master/Docs-Playerservices.md

The streamABC Websockets allows subscribing for meta-data events on streamABC audio streams.

Default URL endpoint for websockets services is: `wss://playerservices.streamabc.net`

The websocket endpoint uses nes-Protocol.

4.1 Metadata Events for Users Subscription

To receive meta-data events for individual users subscribe to this path: `/metadata/{userid}`

Note: In order for this to work you have to decorate the stream-url with a GET parameter `sabcsid={userid}` and start streaming.

After subscribing the server publishes messages containing a JSON object as payload.

4.2 Metadata Events for Channel Subscription

To receive meta-data events for specific channels subscribe to this path: `/metachannel/{channelkey}`

Please use the channel key provided by streamABC.

4.3 Metadata Events for Station Subscription

To receive meta-data events for specific channels subscribe to this path: `/metastation/{stationid}`

Please use the station-id provided by streamABC.

For more information, please open a ticket:

Visit our company website:

CHAPTER 5

SDK for Playerservices

Our streamABC Javascript-SDK for Playerservices enables you to receive meta data to display in your player in real-time.

The full developer documentation as well as examples for Radioplayer and stand-alone players are *available on Github*.

For more information, please open a ticket:

Visit our company website:

CHAPTER 6

SDK for Radio

Our streamABC Javascript-SDK for Radio enables you to integrate our skipping-enabled channels into your player.

The full developer documentation as well as examples for Radioplayer and stand-alone players are *available on Github*.

For more information, please open a ticket:

Visit our company website:

CHAPTER 7

Country-specific parts for costumers

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

For more information, please open a ticket:

Visit our company website: