

Your Dashboard →

The Transistor API is built around the [JSON:API](#) specification. Our API allows you to work with podcasts, episodes, private podcast subscribers, and analytics. Endpoints accept JSON or form-encoded request bodies, return JSON-encoded responses, and use standard HTTP response codes.

Authentication

The Transistor API uses API keys to authenticate all requests. Your API keys can be viewed and reset in the [Account Area](#) of your Transistor Dashboard. Each API request must include an HTTP header `x-api-key` with its value being your API key.

Authenticated requests grant you access to any podcast or episode you would have access to through the Dashboard, along with the same level of access depending on if you're an owner, an admin, or a regular team member of a podcast.

Rate Limits

API requests are rate-limited to 10 requests per 10 seconds. If this limit is exceeded, you will be receive a `429` HTTP error code, and access will be blocked for 10 seconds. After these 10 seconds, you are free to use the API again.

If you are continually hitting the rate-limit, please review your use of the API and always cache responses when possible. The Transistor API is not meant to be the main data source for website content. If this is what you're looking for, you may want to simply parse the XML from your RSS Feed.

Endpoints

Because our API conforms to the [JSON:API](#) spec, each endpoint also accepts extra parameters that can be helpful in certain scenarios: [sparse fieldsets](#) and [including related resources](#). We've included examples of these in some of our example requests and responses.

Sparse fieldsets

Sometimes you may wish to only have a handful of a resource's fields be returned from an API request, instead of the entire resource. Let's say you want to retrieve an Episode but only return the title and `media_url`. In your request you can specify `fields[episode][]=title&fields[episode][]=media_url`.

Including related resources

You can also return a resource's related resources in one single API request instead of multiple requests. For example, you may want to retrieve an Episode, but also include its parent Show resource. In your request you can specify `include[]=show`. You can also combine this with sparse fieldsets to request an Episode's related Show, but only return the Show's title and RSS Feed URL, for example. `include[]=show&fields[show][]=title&fields[show][]=feed_url`.

Root

Get authenticated user

Retrieve details of the user account that is authenticating to the API.

GET /v1

Parameters

Your Dashboard →

Example

Example

REQUEST

```
$ curl https://api.transistor.fm/v1 -G \  
-H "x-api-key: your_api_key"
```

RESPONSE

```
$ {  
  "data": {  
    "id": "173455",  
    "type": "user",  
    "attributes": {  
      "created_at": "2020-01-01 00:00:00 UTC",  
      "image_url": null,  
      "name": "Jimmy Podcaster",  
      "time_zone": "UTC",  
      "updated_at": "2020-06-01 00:00:00 UTC"  
    }  
  }  
}
```

Analytics

Get show analytics

Retrieve analytics of downloads per day for an entire podcast. Defaults to the last 14 days.

```
GET /v1/analytics/:id
```

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Show ID or slug
start_date	String		Optional starting date for analytics (dd-mm-yyyy). Required if using an ending date.
end_date	String		Optional ending date for analytics (dd-mm-yyyy). Required if using a starting date.

Response

A single [ShowAnalytics resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/analytics/132543 -G \  
-H "x-api-key: your_api_key" \  
-d start_date=01-01-2020 \  
-d end_date=31-12-2020 \  
-d "include[]=show" \  
-d "fields[show][]=title"
```

Your Dashboard →

```
{
  "type": "show_analytics",
  "attributes": {
    "downloads": [
      {
        "date": "22-12-2024",
        "downloads": 0
      },
      {
        "date": "23-12-2024",
        "downloads": 0
      },
      {
        "date": "24-12-2024",
        "downloads": 0
      },
      {
        "date": "25-12-2024",
        "downloads": 0
      }
    ]
  }
}
```

Get all episode analytics

Retrieve analytics of downloads per day for all episodes of a podcast. Defaults to the last 7 days.

GET /v1/analytics/:id/episodes

Parameters

Field	Type	Details	Description
id Required	String		Show ID or slug
start_date	String		Optional starting date for analytics (dd-mm-yyyy). Required if using an ending date.
end_date	String		Optional ending date for analytics (dd-mm-yyyy). Required if using a starting date.

Response

A single [EpisodesAnalytics](#) resource

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/analytics/132543/episodes -G \
-H "x-api-key: your_api_key" \
-d start_date=01-01-2021 \
-d end_date=07-01-2021 \
-d "include[]=show" \
-d "fields[show][]=title"
```

RESPONSE

Your Dashboard →

```
      "type": "show"
    }
  }
},
"included": [
  {
    "id": "132543",
    "type": "show",
    "attributes": {
      "title": "The Caffeine Show"
    },
    "relationships": {}
  }
]
}
```

Get single episode analytics

Retrieve analytics of downloads per day for a single episode. Defaults to the last 14 days.

GET /v1/analytics/episodes/:id

Parameters

Field	Type	Details	Description
id	String	Required	Episode ID or slug
start_date	String		Optional starting date for analytics (dd-mm-yyyy). Required if using an ending date.
end_date	String		Optional ending date for analytics (dd-mm-yyyy). Required if using a starting date.

Response

A single [EpisodeAnalytics resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/analytics/episodes/3056098 -G \
-H "x-api-key: your_api_key" \
-d start_date=01-01-2020 \
-d end_date=31-12-2020 \
-d "include[]=episode" \
-d "fields[episode][]=title"
```

RESPONSE

Your Dashboard →

```
"downloads": [  
  {  
    "date": "22-12-2024",  
    "downloads": 0  
  },  
  {  
    "date": "23-12-2024",  
    "downloads": 0  
  },  
  {  
    "date": "24-12-2024",  
    "downloads": 0  
  },  
  {  
    "date": "25-12-2024",  
    "downloads": 0  
  }  
].
```

Episodes

Get episodes

Retrieve a paginated list of episodes ordered by published date.

```
GET /v1/episodes
```

Parameters

Field	Type	Details	Description
show_id	String		Show ID or slug
query	String		Search query
status	String	One of <div>draft</div>	Publishing status: published, scheduled, or draft
order	String	One of <div>desc</div> Default: desc	Return order of episodes. Newest first (desc), or oldest first (asc)
pagination[page]	Integer	Default: 0	Page number
pagination[per]	Integer	Default: 10	Resources per page

Response

An array of [Episode resources](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/episodes -G \  
-H "x-api-key: your_api_key" \  
-d show_id=132543 \  
-d "pagination[page]=1" \  
-d "pagination[per]=5" \  
-d "fields[episode][]=title" \  
-d "fields[episode][]=published_at"
```

Your Dashboard →

```
{
  "id": "3056098",
  "type": "episode",
  "attributes": {
    "title": "How To Roast Coffee",
    "published_at": "2020-07-01 00:00:00 UTC"
  },
  "relationships": {}
},
{
  "id": "3056099",
  "type": "episode",
  "attributes": {
    "title": "The Effects of Caffeine",
    "published_at": "2020-07-01 00:00:00 UTC"
  },
  "relationships": {}
}
],
"meta": {
```

Get an episode

Retrieve a single podcast episode.

GET /v1/episodes/:id

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Episode ID

Response

A single [Episode resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/episodes/3056098 -G \
-H "x-api-key: your_api_key" \
-d "include[]=show" \
-d "fields[show][]=title" \
-d "fields[show][]=summary"
```

RESPONSE

Your Dashboard →

```
"title": "How To Roast Coffee",
"number": 1,
"season": 1,
"status": "published",
"published_at": "2020-07-01 00:00:00 UTC",
"duration": 568,
"explicit": false,
"keywords": "coffee,caffeine,beans",
"alternate_url": null,
"media_url": "https://media.transistor.fm/c2c63c67/1615808c.mp3",
"image_url": null,
"video_url": "https://www.youtube.com/watch?v=xcyHT1ZLd9Y",
"author": "Jimmy Podcaster",
"summary": "A primer on roasting coffee",
"description": "This podcast talks about some <strong>strong</strong> coffee!",
"slug": null,
"created at": "2020-03-01 00:00:00 UTC".
```

Authorize an episode audio upload

Authorize a URL for uploading a local audio file to be used when creating or updating an episode. If you already have a publicly available URL for your audio file, skip this step and use that URL in the `episode[audio_url]` field when creating or updating an episode.

GET /v1/episodes/authorize_upload

Parameters

Field	Type	Details	Description
filename <small>Required</small>	String		Filename of the audio file you wish to upload

Response

A single [AudioUpload](#) resource

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/episodes/authorize_upload -G \
-H "x-api-key: your_api_key" \
-d filename=Episode1.mp3
```

RESPONSE

```
$ {
  "data": {
    "id": "8bc229af-493f-4229-8067-e811097168e3",
    "type": "audio_upload",
    "attributes": {
      "upload_url": "https://transistorupload.s3.amazonaws.com/588ea179cb09e23eaf331d12f600ab9e.mp3?X-Amz-Algorithm=/"
      "content_type": "audio/mpeg",
      "expires_in": 600,
      "audio_url": "https://transistorupload.s3.amazonaws.com/588ea179cb09e23eaf331d12f600ab9e.mp3"
    }
  }
}
```

Your Dashboard →

CURL EXAMPLE

```
$ curl -v -X PUT \
  -H "Content-Type: audio/mpeg" \
  -T /path/to/your/audio/Episode1.mp3 \
  "upload_url_from_authorize_upload"
```

RUBY EXAMPLE

```
require "net/http"

file = "/path/to/your/audio/Episode1.mp3"
presigned_url = "upload_url_from_authorize_upload"
url = URI.parse(presigned_url)

Net::HTTP.start(url.host) do |http|
  http.send_request("PUT", url.request_uri, File.read(file), { "content-type" => "audio/mpeg" })
end
```

Create an episode

Create a new draft episode for the specified show. Note that publishing an episode involves a separate endpoint.

POST /v1/episodes

Parameters

Field	Type	Details	Description
episode[show_id]	String	Required	ID or Slug of the Show to add an episode to
episode[audio_url]	String		URL to an episode's new audio file
episode[transcript_text]	String		Full text of the episode transcript
episode[author]	String		Episode author
episode[description]	String		Longer episode description which may contain HTML and unformatted text
episode[explicit]	Boolean		Episode contains explicit content
episode[image_url]	String		Episode artwork image URL
episode[keywords]	String		Comma-separated list of keywords
episode[number]	Integer		Episode number
episode[season]	Integer		Season number
episode[summary]	String		Episode summary short description
episode[type]	String	One of <input type="text"/>	Full, trailer, or bonus episode
episode[title]	String		Episode title
episode[alternate_url]	String		Alternate episode URL overriding the share_url
episode[video_url]	String		YouTube video URL to be embedded on episode sharing pages and website
episode[email_notifications]	Boolean		Private podcast email notifications override (defaults to Show setting)
episode[increment_number]	Boolean		Automatically set the number to the next episode number of the current season

Your Dashboard →

REQUEST

```
$ curl https://api.transistor.fm/v1/episodes -X POST \
-H "x-api-key: your_api_key" \
-d "episode[show_id]=132543" \
-d "episode[title]=Awesome podcast" \
-d "episode[summary]=A podcast about awesome things" \
-d "episode[season]=2" \
-d "episode[number]=1"
```

RESPONSE

```
$ {
  "data": {
    "id": "3056098",
    "type": "episode",
    "attributes": {
      "title": "Awesome podcast",
      "number": 1,
      "season": 2,
      "status": "draft",
      "published_at": null,
      "duration": null,
      "explicit": false,
      "keywords": null,
      "alternate_url": null,
      "media_url": "https://media.transistor.fm/c024aa5d/5e7f426b.mp3",
      "image_url": null,
      "video_url": null,
      "author": null,
      "summary": "A podcast about awesome things",
      "description": null,
      "slug": null,
      "created at": "2025-01-04 05:41:38 UTC".
    }
  }
}
```

Update an episode

Update a single podcast episode. Note that publishing or unpublishing an episode involves a separate endpoint.

PATCH /v1/episodes/:id

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Episode ID
episode[audio_url]	String		URL to an episode's new audio file
episode[transcript_text]	String		Full text of the episode transcript
episode[author]	String		Episode author
episode[description]	String		Longer episode description which may contain HTML and unformatted text
episode[explicit]	Boolean		Episode contains explicit content
episode[image_url]	String		Episode artwork image URL
episode[keywords]	String		Comma-separated list of keywords
episode[number]	Integer		Episode number
episode[season]	Integer		Season number

Your Dashboard →

episode[title]	String	Episode title
episode[alternate_url]	String	Alternate episode URL overriding the share_url
episode[video_url]	String	YouTube video URL to be embedded on episode sharing pages and w
episode[email_notifications]	Boolean	Private podcast email notifications override (defaults to Show setting)

Response

A single [Episode resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/episodes/3056098 -X PATCH \  
-H "x-api-key: your_api_key" \  
-d "episode[title]=Updated podcast" \  
-d "fields[episode][]=title"
```

RESPONSE

```
$ {  
  "data": {  
    "id": "3056098",  
    "type": "episode",  
    "attributes": {  
      "title": "Updated podcast"  
    },  
    "relationships": {}  
  }  
}
```

Publish, schedule, or unpublish an episode

Publish a single episode now or in the past, schedule for the future, or revert to a draft.

PATCH /v1/episodes/:id/publish

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Episode ID
episode[status] <small>Required</small>	String	One of <input type="text" value="draft"/>	Publishing status: published, scheduled, or draft
episode[published_at]	String		Episode publishing date and time - in your podcast's time zone

Response

A single [Episode resource](#)

Example

REQUEST

Your Dashboard →

RESPONSE

```
$ {
  "data": {
    "id": "3056098",
    "type": "episode",
    "attributes": {
      "status": "published"
    },
    "relationships": {}
  }
}
```

Shows

Get shows

Retrieve a paginated list of shows in descending order by updated date.

GET /v1/shows

Parameters

Field	Type	Details	Description
private	Boolean		Filter for private shows
query	String		Search query
pagination[page]	Integer	Default: 0	Page number
pagination[per]	Integer	Default: 10	Resources per page

Response

An array of [Show resources](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/shows -G \
-H "x-api-key: your_api_key" \
-d "pagination[page]=1" \
-d "pagination[per]=5" \
-d "fields[show][]=title" \
-d "fields[show][]=description"
```

RESPONSE

Your Dashboard →

```
{
  "attributes": {
    "title": "The Caffeine Show",
    "description": "A podcast covering all things coffee and caffeine"
  },
  "relationships": {}
},
{
  "meta": {
    "currentPage": 1,
    "totalPages": 1,
    "totalCount": 1
  }
}
```

Get a show

Retrive a single show (podcast).

GET /v1/shows/:id

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Show ID or slug

Response

A single [Show resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/shows/132543 -G \
-H "x-api-key: your_api_key"
```

RESPONSE

Your Dashboard →

```
"author": null,
"category": "Arts :: Food",
"copyright": null,
"created_at": "2020-02-01 00:00:00 UTC",
"description": "A podcast covering all things coffee and caffeine",
"explicit": false,
"image_url": null,
"keywords": "coffee,caffeine,beans",
"language": "en",
"multiple_seasons": false,
"owner_email": null,
"playlist_limit": 25,
"private": false,
"secondary_category": "Arts",
"show_type": "episodic",
"slug": "the-caffeine-show",
"time_zone": null.
```

Update a show

Update a show with any or all of the following attributes.

PATCH /v1/shows/:id

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Show ID or slug
show[author]	String		Podcast author
show[category]	String	One of <input type="text"/>	Primary category
show[copyright]	String		Copyright information
show[description]	String		Podcast description
show[explicit]	Boolean		Podcast contains explicit content
show[image_url]	String		Podcast artwork image URL
show[keywords]	String		Comma-separated list of keywords
show[language]	String	One of <input type="text"/>	Podcast's spoken language
show[owner_email]	String		Podcast owner email
show[secondary_category]	String	One of <input type="text"/>	Secondary category
show[show_type]	String	One of <input type="text"/>	Publishing type. Episodic displays newest episodes, serial displays older
show[title]	String		Podcast title
show[time_zone]	String	One of <input type="text"/>	Publishing time zone
show[website]	String		Podcast website

Response

Your Dashboard →

```
$ curl https://api.transistor.fm/v1/shows/132543 -X PATCH \
-H "x-api-key: your_api_key" \
-d "show[title]=Updated Title" \
-d "show[author]=Updated Author" \
-d "fields[show][]=title" \
-d "fields[show][]=author"
```

RESPONSE

```
$ {
  "data": {
    "id": "132543",
    "type": "show",
    "attributes": {
      "title": "Updated Title",
      "author": "Updated Author"
    },
    "relationships": {}
  }
}
```

Subscribers

Get a list of subscribers

Retrieve a list of all subscribers for a single private podcast.

GET /v1/subscribers

Parameters

Field	Type	Details	Description
show_id <small>Required</small>	String		Show ID or slug
query	String		Search query
pagination[page]	Integer	Default: 0	Page number
pagination[per]	Integer	Default: 10	Resources per page

Response

An array of [Subscriber resources](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/subscribers -G \
-H "x-api-key: your_api_key" \
-d show_id=132543 \
-d "pagination[page]=1" \
-d "pagination[per]=5" \
-d "fields[subscriber][]=email"
```

RESPONSE

Your Dashboard →

```
{
  "attributes": {
    "email": "arthur@example.com"
  },
  "relationships": {}
},
{
  "id": "709424",
  "type": "subscriber",
  "attributes": {
    "email": "beatrice@example.com"
  },
  "relationships": {}
}
],
"meta": {
  "currentPage": 1,
  "totalPages": 1.
}
```

Retrieve a single private podcast subscriber.

GET /v1/subscribers/:id

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Subscriber ID

Response

A single [Subscriber](#) resource

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/subscribers/709423 -G \
-H "x-api-key: your_api_key"
```

RESPONSE

Your Dashboard →

```
{
  "email": "arthur@example.com",
  "status": "default",
  "feed_url": "https://subscribers.transistor.fm/a99916cf162f22",
  "created_at": "2020-01-01 00:00:00 UTC",
  "updated_at": "2020-06-01 00:00:00 UTC",
  "last_notified_at": "2020-07-01 00:00:00 UTC",
  "has_downloads": false,
  "subscribe_url": "https://subscribe.transistor.fm/a99916cf162f22"
},
"relationships": {
  "show": {
    "data": {
      "id": "132543",
      "type": "show"
    }
  }
}
```

Create a single subscriber

Add a single subscriber to a private podcast, and send an optional instructional email.

POST /v1/subscribers

Parameters

Field	Type	Details	Description
show_id <small>Required</small>	String		Show ID or slug
email <small>Required</small>	String		Email address
skip_welcome_email	Boolean	Default: false	Do not send the instructional email

Response

A single [Subscriber resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/subscribers -X POST \
  -H "x-api-key: your_api_key" \
  -d show_id=132543 \
  -d email=carol@example.com \
  -d "fields[subscriber][]=email"
```

RESPONSE

Your Dashboard →

```
    "email": "carol@example.com"
  },
  "relationships": {}
}
```

Create multiple subscribers

Add a batch of multiple subscribers to a private podcast, and send them optional instructional emails.

POST /v1/subscribers/batch

Parameters

Field	Type	Details	Description
show_id <small>Required</small>	String		Show ID or slug
emails <small>Required</small>	[String]		Array of email addresses
skip_welcome_email	Boolean	Default: false	Do not send the instructional emails

Response

An array of [Subscriber resources](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/subscribers/batch -X POST \
  -H "x-api-key: your_api_key" \
  -d show_id=132543 \
  -d "emails[]=carol@example.com" \
  -d "emails[]=derek@example.com" \
  -d "fields[subscriber][]=email"
```

RESPONSE

Your Dashboard →

```
{
  "attributes": {
    "email": "carol@example.com"
  },
  "relationships": {}
},
{
  "id": "709428",
  "type": "subscriber",
  "attributes": {
    "email": "derek@example.com"
  },
  "relationships": {}
}
]
```

Update a single subscriber

Update a single private podcast subscriber.

PATCH /v1/subscribers/:id

Parameters

Field	Type	Details	Description
id <small>Required</small>	String		Subscriber ID
subscriber[email] <small>Required</small>	String		Subscriber's email address

Response

A single [Subscriber resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/subscribers/709423 -X PATCH \
-H "x-api-key: your_api_key" \
-d "subscriber[email]=updated@example.com" \
-d "fields[subscriber][]=email"
```

RESPONSE

```
$ {
  "data": {
    "id": "709423",
    "type": "subscriber",
    "attributes": {
      "email": "updated@example.com"
    },
    "relationships": {}
  }
}
```

Your Dashboard →

Parameters

Field	Type	Details	Description
show_id Required	String		Show ID or slug
email Required	String		Email address

Response

A single [Subscriber resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/subscribers -X DELETE \
-H "x-api-key: your_api_key" \
-d show_id=132543 \
-d email=carol@example.com
```

RESPONSE

```
$ {
  "data": {
    "id": "709423",
    "type": "subscriber",
    "attributes": {
      "email": "carol@example.com",
      "status": "default",
      "feed_url": "https://subscribers.transistor.fm/57fc9aef3a2960",
      "created_at": "2020-01-01 00:00:00 UTC",
      "updated_at": "2020-06-01 00:00:00 UTC",
      "last_notified_at": "2020-07-01 00:00:00 UTC",
      "has_downloads": false,
      "subscribe_url": "https://subscribe.transistor.fm/57fc9aef3a2960"
    }
  },
  "relationships": {
    "show": {
      "data": {
        "id": "132543",
        "type": "show"
      }
    }
  }
}
```

Delete a single subscriber by ID

Remove a single private podcast subscriber and revoke their access to the podcast.

```
DELETE /v1/subscribers/:id
```

Parameters

Field	Type	Details	Description
id Required	String		Subscriber ID

Response

Your Dashboard →

```
$ curl https://api.transistor.fm/v1/subscribers/709423 -X DELETE \
-H "x-api-key: your_api_key"
```

RESPONSE

```
$ {
  "data": {
    "id": "709423",
    "type": "subscriber",
    "attributes": {
      "email": "arthur@example.com",
      "status": "default",
      "feed_url": "https://subscribers.transistor.fm/aafce64bf97f61",
      "created_at": "2020-01-01 00:00:00 UTC",
      "updated_at": "2020-06-01 00:00:00 UTC",
      "last_notified_at": "2020-07-01 00:00:00 UTC",
      "has_downloads": false,
      "subscribe_url": "https://subscribe.transistor.fm/aafce64bf97f61"
    },
    "relationships": {
      "show": {
        "data": {
          "id": "132543",
          "type": "show"
        }
      }
    }
  }
}
```

Webhooks

Get webhooks

Retrieve a list of webhooks for a show

```
GET /v1/webhooks
```

Parameters

Field	Type	Details	Description
show_id Required	String		Show ID or slug

Response

An array of [Webhook resources](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/webhooks -G \
-H "x-api-key: your_api_key" \
-d show_id=132543
```

RESPONSE

Your Dashboard →

```
"attributes": {
  "event_name": "episode_created",
  "url": "http://example.com/incoming_web_hooks",
  "created_at": null,
  "updated_at": null
},
"relationships": {
  "user": {
    "data": {
      "id": "173455",
      "type": "user"
    }
  },
  "show": {
    "data": {
      "id": "132543",
      "type": "show"
    }
  }
}
```

Subscribe to webhook

Subscribe to a webhook with the given event name and show.

POST /v1/webhooks

Parameters

Field	Type	Details	Description
event_name <small>Required</small>	String	One of <div>episode_created</div>	Name of webhook event
show_id <small>Required</small>	String		Show ID or slug
url <small>Required</small>	String		Target URL for webhook delivery

Response

A single [Webhook resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/webhooks -X POST \
  -H "x-api-key: your_api_key" \
  -d event_name=episode_created \
  -d show_id=1 \
  -d url=http://example.com/incoming_web_hooks
```

RESPONSE

Your Dashboard →

```
{
  "event_name": "episode_created",
  "url": "http://example.com/incoming_web_hooks",
  "created_at": null,
  "updated_at": null
},
"relationships": {
  "user": {
    "data": {
      "id": "173455",
      "type": "user"
    }
  },
  "show": {
    "data": {
      "id": "132543",
      "type": "show"
    }
  }
}
```

Unsubscribe from webhook

Unsubscribe from a webhook.

DELETE /v1/webhooks/:id

Parameters

Field	Type	Details	Description
id Required	String		Webhook ID

Response

A single [Webhook resource](#)

Example

REQUEST

```
$ curl https://api.transistor.fm/v1/webhooks/104325 -X DELETE \
-H "x-api-key: your_api_key"
```

RESPONSE

Your Dashboard →

```
{
  "event_name": "episode_created",
  "url": "http://example.com/incoming_web_hooks",
  "created_at": null,
  "updated_at": null
},
"relationships": {
  "user": {
    "data": {
      "id": "173455",
      "type": "user"
    }
  },
  "show": {
    "data": {
      "id": "132543",
      "type": "show"
    }
  }
}
```

Resources

API Resources are data objects returned by successful API requests, representing Users, Shows, Episodes, Subscribers, Analytics, and Webhooks.

User

The current Transistor user account authenticated for the API.

Fields

Name	Type	Description
created_at	Datetime	Timestamp of creation
image_url	String	Avatar image URL
name	String	Full name
time_zone	String	Current time zone
updated_at	Datetime	Timestamp of last update

Relationships

None.

Show

An individual Transistor show (podcast).

Fields

Name	Type	Description
amazon_music	String	Amazon Music URL
anghami	String	Anghami URL

Your Dashboard →

castbox	String	Castbox URL
castro	String	Castro URL
category	String	Primary category
copyright	String	Copyright information
created_at	Datetime	Timestamp of creation
deezer	String	Deezer URL
description	String	Podcast description
email_notifications	Boolean	Private podcast email notifications enabled or disabled
explicit	Boolean	Podcast contains explicit content
feed_url	String	Podcast RSS Feed URL
fountain	String	Fountain URL
gaana	String	Gaana URL
goodpods	String	Goodpods URL
iHeartRadio	String	iHeartRadio URL
image_url	String	Podcast artwork image URL
jiosaavn	String	JioSaavn URL
keywords	String	Comma-separated list of keywords
language	String	Podcast's spoken language
multiple_seasons	Boolean	Podcast has multiple seasons
overcast	String	Overcast URL
owner_email	String	Podcast owner email
pandora	String	Pandora URL
player_FM	String	Player FM URL
playlist_limit	Integer	Playlist embed player episode limit
pocket_casts	String	Pocket Casts URL
podcast_addict	String	Podcast Addict URL
private	Boolean	Podcast is private and subscribers are managed by admins
secondary_category	String	Secondary category
show_type	String	Publishing type. Episodic displays newest episodes, serial displays oldest first
slug	String	Podcast slug (used for API and RSS Feed URL)
soundcloud	String	Soundcloud URL
spotify	String	Spotify URL
time_zone	String	Publishing time zone
title	String	Podcast title
tuneln	String	Tuneln URL
updated_at	Datetime	Timestamp of last update
website	String	Podcast website

[Your Dashboard](#) →

subscribers

An array of [Subscriber resources](#)

Episode

An individual Transistor podcast episode record.

Fields

Name	Type	Description
alternate_url	String	Alternate episode URL overriding the share_url
audio_processing	Boolean	Denotes processing of audio after creating or updating the audio_url
author	String	Episode author
created_at	Datetime	Timestamp of creation
description	String	Longer episode description which may contain HTML and unformatted tags for chapters, people, supporters, etc
duration	Integer	Duration of episode in seconds
duration_in_mmss	String	Duration of episode in minutes and seconds. e.g. 34:12
email_notifications	String	Private podcast email notifications override (defaults to Show setting)
embed_html	String	Embeddable audio player HTML
embed_html_dark	String	Dark theme of the embeddable audio player HTML
explicit	Boolean	Episode contains explicit content
formatted_description	String	HTML episode description including dynamic content like chapters, people, supporters, etc
formatted_published_at	String	Formatted version of the published_at datetime field
formatted_summary	String	Formatted episode summary short description
image_url	String	Episode artwork image URL
keywords	String	Comma-separated list of keywords
media_url	String	Trackable audio MP3 URL
number	Integer	Episode number
published_at	Datetime	Episode publishing date and time - in your podcast's time zone
season	Integer	Season number
share_url	String	Social media sharing page URL
slug	String	Slugified episode title used in Transistor websites. e.g. my-first-episode
status	String	Publishing status: published, scheduled, or draft
summary	String	Episode summary short description
title	String	Episode title
transcript_url	String	Shareable URL for the episode transcript
transcripts	Array	An array of URLs to AI transcription formats
type	String	Full, trailer, or bonus episode
updated_at	Datetime	Timestamp of last update

Your Dashboard →

Relationships

Name	Type
show	A single Show resource

Subscriber

An individual subscriber record for a private podcast.

Fields

Name	Type	Description
created_at	Datetime	Timestamp of creation
email	String	Subscriber's email address
feed_url	String	URL for subscriber's unique RSS Feed
has_downloads	Boolean	Subscriber has downloaded at least one episode
last_notified_at	Datetime	Timestamp of when subscriber was last emailed
status	String	Email notification status - default, subscribed, or unsubscribed
subscribe_url	String	URL for subscriber's private landing page
updated_at	Datetime	Timestamp of last update

Relationships

Name	Type
show	A single Show resource

ShowAnalytics

Download analytics per day for a single podcast.

Fields

Name	Type	Description
downloads	Array	An array of download counts per day
end_date	String	Ending date of the analytics date range
start_date	String	Starting date of the analytics date range

Relationships

Name	Type
show	A single Show resource

Your Dashboard →

Fields

Name	Type	Description
end_date	String	
episodes	Array	
start_date	String	

Relationships

Name	Type
show	A single Show resource

EpisodeAnalytics

Download analytics per day for a single episode.

Fields

Name	Type	Description
downloads	Array	An array of download counts per day
end_date	String	Ending date of the analytics date range
start_date	String	Starting date of the analytics date range

Relationships

Name	Type
episode	A single Episode resource

AudioUpload

Authorized audio upload resource, including the upload_url to upload the audio file to, and the content_type to use when uploading.

Fields

Name	Type	Description
audio_url	String	URL of your audio file after uploading is complete. To be used when creating or updating an episode
content_type	String	Content type of the file to upload. audio/mpeg, audio/wav, etc
expires_in	Integer	Amount of time in seconds before the authorized upload URL expires
upload_url	String	Endpoint URL to upload your audio file to using HTTP PUT

Relationships

None.

Your Dashboard →

Fields

Name	Type	Description
created_at	String	Timestamp of creation
event_name	String	Name of webhook event
updated_at	String	Timestamp of last update
url	String	Target URL for webhook delivery

Relationships

Name	Type
user	A single User resource
show	A single Show resource

[↑](#) Back to the top