

# Festigram API Documentation

## Version 2

Festigram's API allows you to bring festival information directly to your applications. Available data includes:

- A list of festivals and their venues
- Each festival's lineup, with dates and times adjusted for time zones
- A profile of every artist in every lineup of every festival

Use the API with the models and methods listed below.

### Table of Contents

Artist.....	2
ArtistAlias.....	4
ArtistGenre.....	5
Date.....	6
Day .....	7
Festival .....	8
Genre .....	10
Lineup.....	11
Venue .....	12

## Artist

PATCH	/Artists	Patch an existing model instance or insert a new one into the data source.
GET	/Artists	Find all instances of the model matched by filter from the data source.
PUT	/Artists	Replace an existing model instance or insert a new one into the data source.
POST	/Artists	Create a new instance of the model and persist it into the data source.
PATCH	/Artists/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Artists/{id}	Find a model instance by id from the data source.
READ	/Artists/{id}	Check whether a model instance exists in the data source.
PUT	/Artists/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Artists/{id}	Delete a model instance by id from the data source.
GET	/Artists/{id}/aliases	Queries aliases of artist.
POST	/Artists/{id}/aliases	Creates a new instance in aliases of this model.
DELETE	/Artists/{id}/aliases	Deletes all aliases of this model.
GET	/Artists/{id}/aliases/{fk}	Find a related item by id for aliases.
PUT	/Artists/{id}/aliases/{fk}	Update a related item by id for aliases.
DELETE	/Artists/{id}/aliases/{fk}	Delete a related item by id for aliases.
GET	/Artists/{id}/aliases/count	Counts aliases of artist.
GET	/Artists/{id}/exists	Check whether a model instance exists in the data source.
GET	/Artists/{id}/festivals	Queries festivals of artist.
POST	/Artists/{id}/festivals	Creates a new instance in festivals of this model.
DELETE	/Artists/{id}/festivals	Deletes all festivals of this model.
GET	/Artists/{id}/festivals/{fk}	Find a related item by id for festivals.
PUT	/Artists/{id}/festivals/{fk}	Update a related item by id for festivals.
DELETE	/Artists/{id}/festivals/{fk}	Delete a related item by id for festivals.
GET	/Artists/{id}/festivals/{nk}/artists	Queries artists of festival.
POST	/Artists/{id}/festivals/{nk}/artists	Creates a new instance in artists of this model.
DELETE	/Artists/{id}/festivals/{nk}/artists	Deletes all artists of this model.
GET	/Artists/{id}/festivals/{nk}/artists/{fk}	Find a related item by id for artists.

PUT	/Artists/{id}/festivals/{nk}/artists/{fk}	Update a related item by id for artists.
DELETE	/Artists/{id}/festivals/{nk}/artists/{fk}	Delete a related item by id for artists.
GET	/Artists/{id}/festivals/{nk}/artists/count	Counts artists of festival.
READ	/Artists/{id}/festivals/{nk}/artists/rel/{fk}	Check the existence of artists' relation to an item by id.
PUT	/Artists/{id}/festivals/{nk}/artists/rel/{fk}	Add a related item by id for artists.
DELETE	/Artists/{id}/festivals/{nk}/artists/rel/{fk}	Remove the artists' relation to an item by id.
GET	/Artists/{id}/festivals/{nk}/dates	Queries dates of festival.
POST	/Artists/{id}/festivals/{nk}/dates	Creates a new instance in dates of this model.
DELETE	/Artists/{id}/festivals/{nk}/dates/	Deletes all dates of this model.
GET	/Artists/{id}/festivals/{nk}/dates/{fk}	Find a related item by id for dates.
PUT	/Artists/{id}/festivals/{nk}/dates/{fk}	Update a related item by id for dates.
DELETE	/Artists/{id}/festivals/{nk}/dates/{fk}	Delete a related item by id for dates.
GET	/Artists/{id}/festivals/{nk}/dates/count	Counts dates of festival.
GET	/Artists/{id}/festivals/{nk}/lineup	Queries lineup of festival.
POST	/Artists/{id}/festivals/{nk}/lineup	Creates a new instance in lineup of this model.
DELETE	/Artists/{id}/festivals/{nk}/lineup	Deletes all lineups of this model.
GET	/Artists/{id}/festivals/{nk}/lineup/{fk}	Find a related item by id for lineup.
PUT	/Artists/{id}/festivals/{nk}/lineup/{fk}	Update a related item by id for lineup.
DELETE	/Artists/{id}/festivals/{nk}/lineup/{fk}	Delete a related item by id for lineup.
GET	/Artists/{id}/festivals/{nk}/lineup/count	Counts lineup of festival.
GET	/Artists/{id}/festivals/{nk}/series	Fetches belongsTo relation series.
GET	/Artists/{id}/festivals/count	Counts festivals of artist.
READ	/Artists/{id}/festivals/rel/{fk}	Check the existence of festivals relation to an item by id.
PUT	/Artists/{id}/festivals/rel/{fk}	Add a related item by id for festivals.
DELETE	/Artists/{id}/festivals/rel/{fk}	Remove the festivals' relation to an item by id.
GET	/Artists/{id}/lineups	Queries lineups of artist.
POST	/Artists/{id}/lineups	Creates a new instance in lineups of this model.
DELETE	/Artists/{id}/lineups	Deletes all lineups of this model.
GET	/Artists/{id}/lineups/{fk}	Find a related item by id for lineups.
PUT	/Artists/{id}/lineups/{fk}	Update a related item by id for lineups.
DELETE	/Artists/{id}/lineups/{fk}	Delete a related item by id for lineups.
GET	/Artists/{id}/lineups/count	Counts lineups of artist.
POST	/Artists/{id}/replace	Replace attributes for a model instance and persist it into the data source.
POST	/Artists/addToLineup/{festivalId}	Add an artist to a festival lineup by festivalId.

POST	/Artists/admin/merge/{artistId_1}{artistId_2}	Merge two artists together under the name artistId_1.
GET	/Artists/change-stream	Create a change stream.
POST	/Artists/change-stream	Create a change stream.
GET	/Artists/count	Count instances of the model matched within the data source.
GET	/Artists/discovery	Deprecated and will be removed in a later version.
POST	/Artists/festivalLineup/{festivalId}	Find an artist within a lineup by festivalId.
GET	/Artists/findOne	Find first instance of the model matched within the data source.
POST	/Artists/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
GET	/Artists/search/{pattern}	Find an artist by searching with a regular expression.
POST	/Artists/update	Update instances of the model matched within the data source.
POST	/Artists/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## ArtistAlias

PATCH	/ArtistAliases	Patch an existing model instance or insert a new one into the data source.
GET	/ArtistAliases	Find all instances of the model matched within the data source.
PUT	/ArtistAliases	Replace an existing model instance or insert a new one into the data source.
POST	/ArtistAliases	Create a new instance of the model and persist it into the data source.
PATCH	/ArtistAliases/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/ArtistAliases/{id}	Find a model instance by id from the data source.
READ	/ArtistAliases/{id}	Check whether a model instance exists in the data source.
PUT	/ArtistAliases/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/ArtistAliases/{id}	Delete a model instance by id from the data source.
GET	/ArtistAliases/{id}/exists	Check whether a model instance exists in the

		data source.
POST	/ArtistAliases/{id}/replace	Replace attributes for a model instance and persist it into the data source.
POST	/ArtistAliases/batchCreate	Create multiple artist aliases simultaneously.
POST	/ArtistAliases/batchDelete	Delete multiple artist aliases simultaneously.
GET	/ArtistAliases/change-stream	Create a change stream.
POST	/ArtistAliases/change-stream	Create a change stream.
GET	/ArtistAliases/count	Count instances of the model matched within the data source.
GET	/ArtistAliases/findOne	Find first instance of the model matched within the data source.
POST	/ArtistAliases/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/ArtistAliases/update	Update instances of the model matched within the data source.
POST	/ArtistAliases/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## ArtistGenre

PATCH	/ArtistGenres	Patch an existing model instance or insert a new one into the data source.
GET	/ArtistGenres	Find all instances of the model matched within the data source.
PUT	/ArtistGenres	Replace an existing model instance or insert a new one into the data source.
POST	/ArtistGenres	Create a new instance of the model and persist it into the data source.
PATCH	/ArtistGenres/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/ArtistGenres/{id}	Find a model instance by id from the data source.
READ	/ArtistGenres/{id}	Check whether a model instance exists in the data source.
PUT	/ArtistGenres/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/ArtistGenres/{id}	Delete a model instance by id from the data source.
GET	/ArtistGenres/{id}/exists	Check whether a model instance exists in the data source.
POST	/ArtistGenres/{id}/replace	Replace attributes for a model instance and persist it into the data source.
GET	/ArtistGenres/change-stream	Create a change stream.

POST	/ArtistGenres/change-stream	Create a change stream.
GET	/ArtistGenres/count	Count instances of the model matched within the data source.
GET	/ArtistGenres/findOne	Find first instance of the model matched within the data source.
POST	/ArtistGenres/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/ArtistGenres/update	Update instances of the model matched within the data source.
POST	/ArtistGenres/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## Date

PATCH	/Dates	Patch an existing model instance or insert a new one into the data source.
GET	/Dates	Find all instances of the model matched within the data source.
PUT	/Dates	Replace an existing model instance or insert a new one into the data source.
POST	/Dates	Create a new instance of the model and persist it into the data source.
PATCH	/Dates/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Dates/{id}	Find a model instance by id from the data source.
READ	/Dates/{id}	Check whether a model instance exists in the data source.
PUT	/Dates/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Dates/{id}	Delete a model instance by id from the data source.
GET	/Dates/{id}/days	Queries days of date.
POST	/Dates/{id}/days	Creates a new instance in days of this model.
DELETE	/Dates/{id}/days	Deletes all days of this model.
GET	/Dates/{id}/days/{fk}	Find a related item by id for days.
PUT	/Dates/{id}/days/{fk}	Update a related item by id for days.
DELETE	/Dates/{id}/days/{fk}	Delete a related item by id for days.
GET	/Dates/{id}/days/count	Counts days of date.
GET	/Dates/{id}/exists	Check whether a model instance exists in the data source.
GET	/Dates/{id}/festival	Fetches belongsTo relation festival.
POST	/Dates/{id}/replace	Replace attributes for a model instance and

		persist it into the data source.
GET	/Dates/change-stream	Create a change stream.
POST	/Dates/change-stream	Create a change stream.
GET	/Dates/count	Count instances of the model matched within the data source.
POST	/Dates/createWithDays	Create a date with a day argument.
GET	/Dates/findOne	Find first instance of the model matched within the data source.
POST	/Dates/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/Dates/update	Update instances of the model matched within the data source.
POST	/Dates/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## Day

PATCH	/Days	Patch an existing model instance or insert a new one into the data source.
GET	/Days	Find all instances of the model matched within the data source.
PUT	/Days	Replace an existing model instance or insert a new one into the data source.
POST	/Days	Create a new instance of the model and persist it into the data source.
PATCH	/Days/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Days/{id}	Find a model instance by id from the data source.
READ	/Days/{id}	Check whether a model instance exists in the data source.
PUT	/Days/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Days/{id}	Delete a model instance by id from the data source.
GET	/Days/{id}/dates	Fetches belongsTo relation dates.
GET	/Days/{id}/exists	Check whether a model instance exists in the data source.
POST	/Days/{id}/replace	Replace attributes for a model instance and persist it into the data source.
GET	/Days/{id}/sets	Queries sets of day.
POST	/Days/{id}/sets	Creates a new instance in sets of this model.
DELETE	/Days/{id}/sets	Deletes all sets of this model.

GET	/Days/{id}/sets/{fk}	Find a related item by id for sets.
PUT	/Days/{id}/sets/{fk}	Update a related item by id for sets.
DELETE	/Days/{id}/sets/{fk}	Delete a related item by id for sets.
GET	/Days/{id}/sets/counts	Counts sets of day.
GET	/Days/change-stream	Create a change stream.
POST	/Days/change-stream	Create a change stream.
GET	/Days/count	Count instances of the model matched within the data source.
GET	/Days/findOne	Find first instance of the model matched within the data source.
POST	/Days/greet	Deprecated and to be removed in a later version.
POST	/Days/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/Days/update	Update instances of the model matched within the data source.
POST	/Days/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## Festival

PATCH	/Festivals	Patch an existing model instance or insert a new one into the data source.
GET	/Festivals	Find all instances of the model matched within the data source.
PUT	/Festivals	Replace an existing model instance or insert a new one into the data source.
POST	/Festivals	Create a new instance of the model and persist it into the data source.
PATCH	/Festivals/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Festivals/{id}	Find a model instance by id from the data source.
READ	/Festivals/{id}	Check whether a model instance exists in the data source.
PUT	/Festivals/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Festivals/{id}	Delete a model instance by id from the data source.
GET	/Festivals/{id}/artists	Queries artists of festival.
POST	/Festivals/{id}/artists	Creates a new instance in artists of this model.
DELETE	/Festivals/{id}/artists	Deletes all artists of this model.



DELETE	/Festivals/{id}/artists	Deletes all artists of this model.
GET	/Festivals/{id}/artists/{fk}	Find a related item by id for artists.
PUT	/Festivals/{id}/artists/{fk}	Update a related item by id for artists.
DELETE	/Festivals/{id}/artists/{fk}	Delete a related item by id for artists.
GET	/Festivals/{id}/artists/{nk}/festivals	Queries festivals of artist.
POST	/Festivals/{id}/artists/{nk}/festivals	Creates a new instance in festivals of this model.
DELETE	/Festivals/{id}/artists/{nk}/festivals	Deletes all festivals of this model.
GET	/Festivals/{id}/artists/{nk}/festivals/{fk}	Find a related item by id for festivals.
PUT	/Festivals/{id}/artists/{nk}/festivals/{fk}	Update a related item by id for festivals.
DELETE	/Festivals/{id}/artists/{nk}/festivals/{fk}	Delete a related item by id for festivals.
GET	/Festivals/{id}/artists/{nk}/festivals/count	Counts festivals of artist.
READ	/Festivals/{id}/artists/{nk}/festivals/rel/{fk}	Check the existence of festivals relation to an item by id.
PUT	/Festivals/{id}/artists/{nk}/festivals/rel/{fk}	Add a related item by id for festivals.
DELETE	/Festivals/{id}/artists/{nk}/festivals/rel/{fk}	Remove the festivals' relation to an item by id.
GET	/Festivals/{id}/artists/{nk}/lineups	Queries lineups of artist.
POST	/Festivals/{id}/artists/{nk}/lineups	Creates a new instance in lineups of this model.
DELETE	/Festivals/{id}/artists/{nk}/lineups	Deletes all lineups of this model.
GET	/Festivals/{id}/artists/{nk}/lineups/{fk}	Find a related item by id for lineups.
PUT	/Festivals/{id}/artists/{nk}/lineups/{fk}	Update a related item by id for lineups.
DELETE	/Festivals/{id}/artists/{nk}/lineups/{fk}	Delete a related item by id for lineups.
GET	/Festivals/{id}/artists/{nk}/lineups/count	Counts lineups of artist.
GET	/Festivals/{id}/artists/count	Counts artists of festival.
READ	/Festivals/{id}/artists/rel/{fk}	Check the existence of artists' relation to an item by id.
PUT	/Festivals/{id}/artists/rel/{fk}	Add a related item by id for artists.
DELETE	/Festivals/{id}/artists/rel/{fk}	Remove the artists' relation to an item by id.
GET	/Festivals/{id}/dates	Queries dates of festival.
POST	/Festivals/{id}/dates	Creates a new instance in dates of this model.
DELETE	/Festivals/{id}/dates	Deletes all dates of this model.
GET	/Festivals/{id}/dates/{fk}	Find a related item by id for dates.
PUT	/Festivals/{id}/dates/{fk}	Update a related item by id for dates.
DELETE	/Festivals/{id}/dates/{fk}	Delete a related item by id for dates.
GET	/Festivals/{id}/dates/count	Counts dates of festival.
GET	/Festivals/{id}/exists	Check whether a model instance exists in the data source.
GET	/Festivals/{id}/lineup	Queries lineup of festival.
POST	/Festivals/{id}/lineup	Creates a new instance in lineup of this model.
DELETE	/Festivals/{id}/lineup	Deletes all lineups of this model.
GET	/Festivals/{id}/lineup/{fk}	Find a related item by id for lineup.
PUT	/Festivals/{id}/lineup/{fk}	Update a related item by id for lineup.
DELETE	/Festivals/{id}/lineup/{fk}	Delete a related item by id for lineup.

GET	/Festivals/{id}/lineup/count	Counts lineups of festival.
POST	/Festivals/{id}/replace	Replace attributes for a model instance and persist it into the data source.
GET	/Festivals/{id}/series	Fetches belongsTo relation series.
GET	/Festivals/change-stream	Create a change stream.
POST	/Festivals/change-stream	Create a change stream.
GET	/Festivals/count	Count instances of the model matched within the data source.
GET	/Festivals/findOne	Find first instance of the model matched within the data source.
POST	/Festivals/greet	Deprecated and to be removed in a future version.
POST	/Festivals/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/Festivals/update	Update instances of the model matched within the data source.
POST	/Festivals/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## Genre

PATCH	/Genres	Patch an existing model instance or insert a new one into the data source.
GET	/Genres	Find all instances of the model matched within the data source.
PUT	/Genres	Replace an existing model instance or insert a new one into the data source.
POST	/Genres	Create a new instance of the model and persist it into the data source.
PATCH	/Genres/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Genres/{id}	Find a model instance by id from the data source.
READ	/Genres/{id}	Check whether a model instance exists in the data source.
PUT	/Genres/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Genres/{id}	Delete a model instance by id from the data source.
GET	/Genres/{id}/exists	Check whether a model instance exists in the data source.
POST	/Genres/{id}/replace	Replace attributes for a model instance and persist it into the data source.

GET	/Genres/change-stream	Create a change stream.
POST	/Genres/change-stream	Create a change stream.
GET	/Genres/count	Count instances of the model matched within the data source.
GET	/Genres/findOne	Find first instance of the model matched within the data source.
POST	/Genres/greet	Deprecated and to be removed in a later version.
POST	/Genres/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/Genres/update	Update instances of the model matched within the data source.
POST	/Genres/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## Lineup

PATCH	/Lineups	Patch an existing model instance or insert a new one into the data source.
GET	/Lineups	Find all instances of the model matched within the data source.
PUT	/Lineups	Replace an existing model instance or insert a new one into the data source.
POST	/Lineups	Create a new instance of the model and persist it into the data source.
PATCH	/Lineups/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Lineups/{id}	Find a model instance by id from the data source.
READ	/Lineups/{id}	Check whether a model instance exists in the data source.
PUT	/Lineups/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Lineups/{id}	Delete a model instance by id from the data source.
GET	/Lineups/{id}/exists	Check whether a model instance exists in the data source.
GET	/Lineups/{id}/festival	Fetches belongsTo relation festival.
POST	/Lineups/{id}/replace	Replace attributes for a model instance and persist it into the data source.
POST	/Lineups/batchDelete	Delete multiple lineups simultaneously.
POST	/Lineups/batchUpdate	Update multiple lineups simultaneously.
GET	/Lineups/change-stream	Create a change stream.

POST	/Lineups/change-stream	Create a change stream.
GET	/Lineups/count	Count instances of the model matched within the data source.
GET	/Lineups/findOne	Find first instance of the model matched within the data source.
POST	/Lineups/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/Lineups/update	Update instances of the model matched within the data source.
POST	/Lineups/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.

## Venue

PATCH	/Venues	Patch an existing model instance or insert a new one into the data source.
GET	/Venues	Find all instances of the model matched by filter from the data source.
PUT	/Venues	Replace an existing model instance or insert a new one into the data source.
POST	/Venues	Create a new instance of the model and persist it into the data source.
PATCH	/Venues/{id}	Patch attributes for a model instance and persist it into the data source.
GET	/Venues/{id}	Find a model instance by id from the data source.
READ	/Venues/{id}	Check whether a model instance exists in the data source.
PUT	/Venues/{id}	Replace attributes for a model instance and persist it into the data source.
DELETE	/Venues/{id}	Delete a model instance by id from the data source.
GET	/Venues/{id}/exists	Check whether a model instance exists in the data source.
POST	/Venues/{id}/replace	Replace attributes for a model instance and persist it into the data source.
GET	/Venues/change-stream	Create a change stream.
POST	/Venues/change-stream	Create a change stream.
GET	/Venues/count	Count instances of the model matched within the data source.
GET	/Venues/findOne	Find first instance of the model matched within the data source.
POST	/Venues/greet	Deprecated and to be removed from a later

		version.
POST	/Venues/replaceOrCreate	Replace an existing model instance or insert a new one into the data source.
POST	/Venues/update	Update instances of the model matched within the data source.
POST	/Venues/upsertWithWhere	Update an existing model instance or insert a new one into the data source based on the where criteria.