Package 'impectR'

October 27, 2026

Title Access Data from the 'Impect' API

Version 2.7.0

Maintainer Florian Schmitt <florian.schmitt@catapultsports.com>

Description Pull data from the 'Impect' Customer API <https://glossary.impect.com/api-design>. The package can retrieve data such as events or match sums.

License MIT + file LICENSE

SystemRequirements ffmpeg (optional, only required for getVideoClips())

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, httr, jsonlite, purrr, tidyr, utils, methods, tibble, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Catapult Sports [cph], Florian Schmitt [aut, cre], Thomas Walentin [ctb]

Repository CRAN

Date/Publication 2026-10-27 10:00:02 UTC

Contents

getAccessToken 2
getData 3
getEvents 4
getFormations 5
getIterations 6
getMatches 7
getMatchPredictions 8
getPlayerIterationAverages 9

getPlayerIterationScores 10
getPlayerMatchScores 11
getPlayerMatchsums 12
getPlayerProfileScores 13
getSeasonSimulations 14
getSetPieces 15
getSquadCoefficients 16
getSquadIterationAverages 17
getSquadIterationScores 18
getSquadIterationStyleOfPlay 19
getSquadMatchScores 20
getSquadMatchStyleOfPlay 21
getSquadMatchsums 22
getSquadRatings 23
getStartingPositions 24
getSubstitutions 25
getVideoClips 26
TokenBucket-class 27

Index **29**

`getAccessToken` *Get an access token from the 'Impect' Customer API*

Description

Get an access token from the 'Impect' Customer API

Usage

```
getAccessToken(
  username,
  password,
  token_url =
   "https://login.impect.com/auth/realms/production/protocol/openid-connect/token"
)
```

Arguments

`username`	your 'IMPECT' username
`password`	your 'IMPECT' password
`token_url`	host specific token url

Value

a string containing a bearer token

Examples

```
# Toy example: this will error quickly (no credentials)
try(getAccessToken(username = "invalidUser", password = "invalidPassword"))

# Real usage: requires valid credentials
## Not run:
  token <- getAccessToken(username = "yourUsername", password = "yourPassword")

## End(Not run)
```

`getData` *Send a request to any 'IMPECT' API endpoint and return the response as a flattened dataframe*

Description

Use this function to access endpoints that are not wrapped by a dedicated function of this package. The request respects the same rate limits as all other functions. Nested fields of the response are flattened into separate columns whose names are converted to camel case (e.g. `squadHome.id` becomes `squadHomeId`).

Usage

```
getData(
  url,
  token,
  method = "GET",
  data = NULL,
  host = "https://api.impect.com"
)
```

Arguments

`url`	endpoint path (e.g. `"/v5/customerapi/countries"`), which is prefixed with `host`, or full URL, which must start with `host`
`token`	bearer token
`method`	HTTP method
`data`	optional request body as a list, sent as JSON
`host`	host environment

Value

a dataframe containing the flattened `data` of the response, or an empty dataframe if the response has no content

Examples

```
# Toy example: this will error quickly (no API token)
try(countries <- getData(
  url = "/v5/customerapi/countries",
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
# get the list of countries, which is not wrapped by a dedicated function
countries <- getData(
  url = "/v5/customerapi/countries",
  token = "yourToken"
)

## End(Not run)
```

`getEvents` *Return a dataframe that contains all events and kpi values for a set of given match IDs*

Description

Return a dataframe that contains all events and kpi values for a set of given match IDs

Usage

```
getEvents(
  matches,
  token,
  include_kpis = TRUE,
  include_set_pieces = FALSE,
  host = "https://api.impect.com"
)
```

Arguments

`matches`	'IMPECT' match ID or a list of match IDs
`token`	bearer token
`include_kpis`	include KPIs in event data
`include_set_pieces`	include additional set piece data in event data
`host`	host environment

Value

a dataframe containing all events and kpi values for a set of given match IDs

Examples

```
# Toy example: this will error quickly (no API token)
try(events <- getEvents(
  matches = c(0, 1),
  token = "invalid",
  include_kpis = T,
  include_set_pieces = F
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
events <- getEvents(
  matches = c(84248, 158150),
  token = "yourToken",
  include_kpis = T,
  include_set_pieces = F
)

## End(Not run)
```

`getFormations` *Return a dataframe that contains all starting formations for a set of given match IDs*

Description

Return a dataframe that contains all starting formations for a set of given match IDs

Usage

```
getFormations(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	'IMPECT' match ID or a list of match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing all starting formations for a set of given match IDs

Examples

```
# Toy example: this will error quickly (no API token)
try(events <- getFormations(
  matches = c(0, 1),
  token = "invalid"
```

```
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
formations <- getFormations(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getIterations` *Return a dataframe containing all iterations available to the user*

Description

Return a dataframe containing all iterations available to the user

Usage

```
getIterations(token, host = "https://api.impect.com")
```

Arguments

`token`	bearer token
`host`	host environment

Value

a dataframe containing all iterations available to the user

Examples

```
# Toy example: this will error quickly (no API token)
try(events <- getIterations(
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
iterations <- getIterations(
  token = "yourToken"
)

## End(Not run)
```

`getMatches` *Return a dataframe with basic information for all matches for a given iteration ID*

Description

Return a dataframe with basic information for all matches for a given iteration ID

Usage

```
getMatches(iteration, token, host = "https://api.impect.com")
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`host`	host environment

Value

a dataframe containing all matches for a given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(matchplan <- getMatches(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
matchplan <- getMatches(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

`getMatchPredictions` *Return a dataframe that contains match predictions for a given iteration ID*

Description

Return a dataframe that contains match predictions for a given iteration ID

Usage

```
getMatchPredictions(iteration, token, host = "https://api.impect.com")
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`host`	host environment

Value

a dataframe containing the match predictions for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(match_predictions <- getMatchPredictions(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
match_predictions <- getMatchPredictions(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

`getPlayerIterationAverages` *Return a dataframe that contains all player averages for a given iteration ID*

Description

Return a dataframe that contains all player averages for a given iteration ID

Usage

```
getPlayerIterationAverages(iteration, token, host = "https://api.impect.com")
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`host`	host environment

Value

a dataframe containing the KPI averages aggregated per player and position for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(player_avgs <- getPlayerIterationAverages(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
player_avgs <- getPlayerIterationAverages(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

`getPlayerIterationScores` *Return a dataframe that contains all player scores for a given iteration ID*

Description

Return a dataframe that contains all player scores for a given iteration ID

Usage

```
getPlayerIterationScores(
  iteration,
  token,
  positions = NULL,
  host = "https://api.impect.com"
)
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`positions`	optional list of position names. Must be one of: "GOALKEEPER", "LEFT_WINGBACK_DEFENDER", "RIGHT_WINGBACK_DEFENDER", "CENTRAL_DEFENDER", "DEFENSE_MIDFIELD", "CENTRAL_MIDFIELD", "ATTACKING_MIDFIELD", "LEFT_WINGER", "RIGHT_WINGER", "CENTER_FORWARD". If not submitted, function will return all positions individually.
`host`	host environment

Value

a dataframe containing the player scores aggregated per player for the given iteration ID and list of positions

Examples

```
# Toy example: this will error quickly (no API token)
try(player_scores <- getPlayerIterationScores(
  iteration = 0,
  positions = c("INVALID_POSITION_1", "INVALID_POSITION_2"),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
player_scores <- getPlayerIterationScores(
  iteration = 1004,
```

```
  positions = c("CENTRAL_DEFENDER", "DEFENSE_MIDFIELD"),
  token = "yourToken"
)

## End(Not run)
```

`getPlayerMatchScores` *Return a dataframe that contains all player scores for a given match ID and list of positions*

Description

Return a dataframe that contains all player scores for a given match ID and list of positions

Usage

```
getPlayerMatchScores(
  matches,
  token,
  positions = NULL,
  host = "https://api.impect.com"
)
```

Arguments

`matches`	'IMPECT' match IDs
`token`	bearer token
`positions`	optional list of position names. Must be one of: "GOALKEEPER", "LEFT_WINGBACK_DEFENDER", "RIGHT_WINGBACK_DEFENDER", "CENTRAL_DEFENDER", "DEFENSE_MIDFIELD", "CENTRAL_MIDFIELD", "ATTACKING_MIDFIELD", "LEFT_WINGER", "RIGHT_WINGER", "CENTER_FORWARD". If not submitted, function will return all positions individually.
`host`	host environment

Value

a dataframe containing the scores aggregated per player and position for the given match ID and list of positions

Examples

```
# Toy example: this will error quickly (no API token)
try(player_match_scores <- getPlayerMatchScores(
  matches = c(0, 1),
  positions = c("INVALID_POSITION_1", "INVALID_POSITION_2"),
  token = "invalid"
))
```

```
# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
player_match_scores <- getPlayerMatchScores(
  matches = c(84248, 158150),
  positions = c("CENTRAL_DEFENDER", "DEFENSE_MIDFIELD"),
  token = "yourToken"
)

## End(Not run)
```

`getPlayerMatchsums` *Return a dataframe that contains all player matchsums for a given match ID*

Description

Return a dataframe that contains all player matchsums for a given match ID

Usage

```
getPlayerMatchsums(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	'IMPECT' match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing the matchsums aggregated per player and position for the given match ID

Examples

```
# Toy example: this will error quickly (no API token)
try(player_match_sums <- getPlayerMatchsums(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
player_match_sums <- getPlayerMatchsums(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getPlayerProfileScores` *Return a dataframe that contains all player profile scores for a given iteration ID*

Description

Return a dataframe that contains all player profile scores for a given iteration ID

Usage

```
getPlayerProfileScores(
  iteration,
  positions,
  token,
  host = "https://api.impect.com"
)
```

Arguments

`iteration`	'IMPECT' iteration ID
`positions`	list of position names. Must be one of: "GOALKEEPER", "LEFT_WINGBACK_DEFENDER", "RIGHT_WINGBACK_DEFENDER", "CENTRAL_DEFENDER", "DEFENSE_MIDFIELD", "CENTRAL_MIDFIELD", "ATTACKING_MIDFIELD", "LEFT_WINGER", "RIGHT_WINGER", "CENTER_FORWARD"
`token`	bearer token
`host`	host environment

Value

a dataframe containing the player profilescores aggregated per player for the given iteration ID and list of positions

Examples

```
# Toy example: this will error quickly (no API token)
try(player_profile_scores <- getPlayerProfileScores(
  iteration = 0,
  positions = c("INVALID_POSITION_1", "INVALID_POSITION_2"),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
player_profile_scores <- getPlayerProfileScores(
  iteration = 1004,
  positions = c("CENTRAL_DEFENDER", "DEFENSE_MIDFIELD"),
```

```
  token = "yourToken"
)

## End(Not run)
```

`getSeasonSimulations` *Return a dataframe that contains season simulations for a given iteration ID*

Description

For every simulation date, a season simulation shows the actual table, the expected final points and goals and the probability of every finishing position of each squad. The rank probabilities are expanded into one column per finishing position (`rank1Probability`, `rank2Probability`, ...) and refer to the squad's own group.

Usage

```
getSeasonSimulations(
  iteration,
  token,
  date = NULL,
  host = "https://api.impect.com"
)
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`date`	optional date (`"YYYY-MM-DD"`) to only return the latest simulation on or before that date
`host`	host environment

Value

a dataframe containing the season simulations for the given iteration ID, or an empty dataframe if no simulations are available

Examples

```
# Toy example: this will error quickly (no API token)
try(season_simulations <- getSeasonSimulations(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
```

```
season_simulations <- getSeasonSimulations(
  iteration = 1004,
  token = "yourToken"
)

# get the current simulation only
current_simulation <- getSeasonSimulations(
  iteration = 1004,
  token = "yourToken",
  date = base::format(base::Sys.Date())
)

## End(Not run)
```

`getSetPieces` *Return a dataframe that contains all set pieces and aggregated kpi values per set piece sub phase for a set of given list of match IDs*

Description

Return a dataframe that contains all set pieces and aggregated kpi values per set piece sub phase for a set of given list of match IDs

Usage

```
getSetPieces(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	list fo 'IMPECT' match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing all set pieces and aggregated kpi values per set piece sub phase for a set of given list of match IDs

Examples

```
# Toy example: this will error quickly (no API token)
try(set_pieces <- getSetPieces(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
set_pieces <- getSetPieces(
```

```
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getSquadCoefficients` *Return a dataframe that contains squad ratings for a given iteration ID*

Description

Return a dataframe that contains squad ratings for a given iteration ID

Usage

```
getSquadCoefficients(iteration, token, host = "https://api.impect.com")
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`host`	host environment

Value

a dataframe containing the squad ratings for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_ratings <- getSquadRatings(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_ratings <- getSquadRatings(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

getSquadIterationAverages *Return a dataframe that contains all squads averages for a given iteration ID*

Description

Return a dataframe that contains all squads averages for a given iteration ID

Usage

```
getSquadIterationAverages(iteration, token, host = "https://api.impect.com")
```

Arguments

iteration	'IMPECT' iteration ID
token	bearer token
host	host environment

Value

a dataframe containing the KPI averages aggregated per squad for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_avgs <- getSquadIterationAverages(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_avgs <- getSquadIterationAverages(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

`getSquadIterationScores` *Return a dataframe that contains all squads scores for a given iteration ID*

Description

Return a dataframe that contains all squads scores for a given iteration ID

Usage

```
getSquadIterationScores(iteration, token, host = "https://api.impect.com")
```

Arguments

`iteration`	'IMPCET' iteration ID
`token`	bearer token
`host`	host environment

Value

a dataframe containing the squad scores aggregated per squad for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_scores <- getSquadIterationScores(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_scores <- getSquadIterationScores(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

getSquadIterationStyleOfPlay *Return a dataframe that contains the squad style of play values for a given iteration ID*

Description

Return a dataframe that contains the squad style of play values for a given iteration ID

Usage

```
getSquadIterationStyleOfPlay(iteration, token, host = "https://api.impect.com")
```

Arguments

iteration	'IMPECT' iteration ID
token	bearer token
host	host environment

Value

a dataframe containing the style of play values per squad for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_style_of_play <- getSquadIterationStyleOfPlay(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_style_of_play <- getSquadIterationStyleOfPlay(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

`getSquadMatchScores` *Return a dataframe that contains squad level scores and ratios for a given match ID*

Description

Return a dataframe that contains squad level scores and ratios for a given match ID

Usage

```
getSquadMatchScores(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	'IMPECT' match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing the scores and rations aggregated per squad for the given match ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_match_scores <- getSquadMatchScores(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_match_scores <- getSquadMatchScores(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getSquadMatchStyleOfPlay` *Return a dataframe that contains the squad style of play values for the given match IDs*

Description

Return a dataframe that contains the squad style of play values for the given match IDs

Usage

```
getSquadMatchStyleOfPlay(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	'IMPECT' match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing the style of play values per squad for the given match IDs

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_match_style_of_play <- getSquadMatchStyleOfPlay(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_match_style_of_play <- getSquadMatchStyleOfPlay(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getSquadMatchsums` *Return a dataframe that contains squad level matchsums for a given match ID*

Description

Return a dataframe that contains squad level matchsums for a given match ID

Usage

```
getSquadMatchsums(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	'IMPECT' match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing the matchsums aggregated per squad for the given match ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_match_sums <- getSquadMatchsums(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_match_sums <- getSquadMatchsums(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getSquadRatings` *Return a dataframe that contains squad ratings for a given iteration ID*

Description

Return a dataframe that contains squad ratings for a given iteration ID

Usage

```
getSquadRatings(iteration, token, host = "https://api.impect.com")
```

Arguments

`iteration`	'IMPECT' iteration ID
`token`	bearer token
`host`	host environment

Value

a dataframe containing the squad ratings for the given iteration ID

Examples

```
# Toy example: this will error quickly (no API token)
try(squad_ratings <- getSquadRatings(
  iteration = 0,
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
squad_ratings <- getSquadRatings(
  iteration = 1004,
  token = "yourToken"
)

## End(Not run)
```

`getStartingPositions` *Return a dataframe that contains all starting positions for a set of given match IDs*

Description

Return a dataframe that contains all starting positions for a set of given match IDs

Usage

```
getStartingPositions(
  matches,
  token,
  host = "https://api.impect.com",
  include_bench = FALSE
)
```

Arguments

`matches`	'IMPECT' match ID or a list of match IDs
`token`	bearer token
`host`	host environment
`include_bench`	include players from the matchday squad who did not start, with `position = "BENCH"` and `positionSide = NA`

Value

a dataframe containing all starting positions for a set of given match IDs, optionally including bench players

Examples

```
# Toy example: this will error quickly (no API token)
try(starting_pos <- getStartingPositions(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
starting_pos <- getStartingPositions(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getSubstitutions` *Return a dataframe that contains all line-up changes for a set of given match IDs*

Description

Return a dataframe that contains all line-up changes for a set of given match IDs

Usage

```
getSubstitutions(matches, token, host = "https://api.impect.com")
```

Arguments

`matches`	'IMPECT' match ID or a list of match IDs
`token`	bearer token
`host`	host environment

Value

a dataframe containing all line-up changes for a set of given match IDs

Examples

```
# Toy example: this will error quickly (no API token)
try(subs <- getSubstitutions(
  matches = c(0, 1),
  token = "invalid"
))

# Real usage: requires valid Bearer Token from `getAccessToken()`
## Not run:
subs <- getSubstitutions(
  matches = c(84248, 158150),
  token = "yourToken"
)

## End(Not run)
```

`getVideoClips` *Cut a video clip around each event of an events dataframe and merge them into a single video file*

Description

For every row of the (pre-filtered) `events` dataframe a clip window is built as `gameTimeInSec - lead` to `gameTimeInSec + duration + lag`, the corresponding match video is fetched from the 'IMPECT' API and cut with 'ffmpeg', and all clips are merged into a single file at `output_path`. Input row order is preserved and every clip is normalized to a common resolution and frame rate before merging.

Usage

```
getVideoClips(
  events,
  output_path,
  token,
  lead = 3,
  lag = 3,
  target_width = 1920,
  target_height = 1080,
  target_fps = 25,
  warn_above_seconds = 600,
  host = "https://api.impect.com"
)
```

Arguments

`events`	dataframe of events, e.g. a filtered output of `getEvents()`, containing the columns `matchId`, `gameTimeInSec` and `duration`
`output_path`	path of the merged output video file, must end in `.mp4`
`token`	bearer token
`lead`	seconds to include before each event
`lag`	seconds to include after each event
`target_width`	width of the output video in pixels
`target_height`	height of the output video in pixels
`target_fps`	frame rate of the output video
`warn_above_seconds`	warn if the expected total output length exceeds this number of seconds
`host`	host environment

Details

Events whose match video is not available to the user (HTTP 403) are skipped with one warning per match. Clips that contain no footage or fail for any other reason (e.g. API or 'ffmpeg' errors) are skipped with a warning. An error is only raised if the token is invalid or none of the requested clips could be created.

This is an interim solution: the API currently serves full match videos that have to be cut client-side. It will most likely be replaced by an API endpoint that returns already-cut clips.

Requires the 'ffmpeg' binary to be installed and available on the `PATH` (see https://ffmpeg.org/download.html).

Value

the path of the created video file

Examples

```
# Real usage: requires valid Bearer Token from `getAccessToken()` and
# 'ffmpeg' on the PATH
## Not run:
events <- getEvents(matches = c(84248), token = "yourToken")
shots <- events[events$actionType == "SHOT", ]
getVideoClips(
  events = shots,
  output_path = "shots.mp4",
  token = "yourToken"
)

## End(Not run)
```

`TokenBucket-class` *TokenBucket Class*

Description

This class represents a token bucket, which is a rate limiting mechanism used to control the frequency of certain actions or requests based on available tokens. Tokens are added to the bucket at a specified rate and can be consumed when needed.

Fields

`capacity` The maximum number of tokens in the bucket.

`tokens` The number of tokens currently available in the bucket.

`interval` The duration in seconds after which the bucket gets refilled.

`last_update` The timestamp of the last update to the bucket.

Examples

```
try({
# create token bucket
bucket <- TokenBucket(
  capacity = 10,
  tokens = 10,
  intervall = 1,
  last_update = Sys.Time()
)

# add tokens to bucket
bucket$addTokens()

# check if a token is available
bucket$isTokenAvailable

# consume token
bucket$consumeToken
})
```

Index

∗ **classes**
TokenBucket-class, 27

getAccessToken, 2
getData, 3
getEvents, 4
getFormations, 5
getIterations, 6
getMatches, 7
getMatchPredictions, 8
getPlayerIterationAverages, 9
getPlayerIterationScores, 10
getPlayerMatchScores, 11
getPlayerMatchsums, 12
getPlayerProfileScores, 13
getSeasonSimulations, 14
getSetPieces, 15
getSquadCoefficients, 16
getSquadIterationAverages, 17
getSquadIterationScores, 18
getSquadIterationStyleOfPlay, 19
getSquadMatchScores, 20
getSquadMatchStyleOfPlay, 21
getSquadMatchsums, 22
getSquadRatings, 23
getStartingPositions, 24
getSubstitutions, 25
getVideoClips, 26

TokenBucket *(*TokenBucket-class*)*, 27
TokenBucket-class, 27