

# Package ‘impectR’

September 8, 2025

**Title** Access Data from the Impect API

**Version** 2.4.3

**Maintainer** Florian Schmitt <florian.schmitt@impect.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

**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** Impect GmbH [cph],  
Florian Schmitt [aut, cre],  
Thomas Walentin [ctb]

**Repository** CRAN

**Date/Publication** 2025-09-08 18:40:17 UTC

## Contents

getAccessToken . . . . .	2
getEvents . . . . .	3
getFormations . . . . .	4
getIterations . . . . .	4
getMatches . . . . .	5
getPlayerIterationAverages . . . . .	6
getPlayerIterationScores . . . . .	7
getPlayerMatchScores . . . . .	8
getPlayerMatchsums . . . . .	9
getPlayerProfileScores . . . . .	9

getSetPieces . . . . .	10
getSquadIterationAverages . . . . .	11
getSquadIterationScores . . . . .	12
getSquadMatchScores . . . . .	13
getSquadMatchsums . . . . .	14
getSquadRatings . . . . .	14
getStartingPositions . . . . .	15
getSubstitutions . . . . .	16
TokenBucket-class . . . . .	17

<b>Index</b>	<b>18</b>
--------------	-----------

---

getAccessToken	<i>Get an access token from the 'Impect' Customer API</i>
----------------	---

---

## Description

Get an access token from the 'Impect' Customer API

## Usage

```
getAccessToken(username, password)
```

## Arguments

username	your 'IMPECT' username
password	your 'IMPECT' password

## 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)
```

---

getEvents	<i>Return a dataframe that contains all events and kpi values for a set of given match IDs</i>
-----------	--

---

### 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)
```

### 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

### 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	<i>Return a dataframe that contains all starting formations for a set of given match IDs</i>
---------------	--

---

**Description**

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

**Usage**

```
getFormations(matches, token)
```

**Arguments**

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

**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	<i>Return a dataframe containing all iterations available to the user</i>
---------------	---

---

**Description**

Return a dataframe containing all iterations available to the user

**Usage**

```
getIterations(token)
```

**Arguments**

token            bearer token

**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	<i>Return a dataframe with basic information for all matches for a given iteration ID</i>
------------	---

---

**Description**

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

**Usage**

```
getMatches(iteration, token)
```

**Arguments**

iteration        'IMPECT' iteration ID  
token            bearer token

**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)
```

---

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)
```

**Arguments**

iteration	'IMPECT' iteration ID
token	bearer token

**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, positions, token)
```

### 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

### 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, positions, token)
```

### Arguments

<code>matches</code>	'IMPECT' match IDs
<code>positions</code>	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"
<code>token</code>	bearer token

### 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)
```

**Arguments**

matches	'IMPECT' match IDs
token	bearer token

**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)
```

**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

**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)
```

---

getSetPieces	<i>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</i>
--------------	---

---

**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)
```

**Arguments**

matches	list fo 'IMPECT' match IDs
token	bearer token

**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)
```

---

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)
```

**Arguments**

iteration	'IMPECT' iteration ID
token	bearer token

**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)
```

### Arguments

iteration	'IMPCET' iteration ID
token	bearer token

### 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)
```

---

getSquadMatchScores	<i>Return a dataframe that contains squad level scores and ratios for a given match ID</i>
---------------------	--

---

### Description

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

### Usage

```
getSquadMatchScores(matches, token)
```

### Arguments

matches	'IMPECT' match IDs
token	bearer token

### 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)
```

---

getSquadMatchsums	<i>Return a dataframe that contains squad level matchsums for a given match ID</i>
-------------------	--

---

**Description**

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

**Usage**

```
getSquadMatchsums(matches, token)
```

**Arguments**

matches	'IMPECT' match IDs
token	bearer token

**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	<i>Return a dataframe that contains squad ratings for a given iteration ID</i>
-----------------	--

---

**Description**

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

**Usage**

```
getSquadRatings(iteration, token)
```

**Arguments**

```
iteration      'IMPECT' iteration ID  
token         bearer token
```

**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)
```

**Arguments**

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

**Value**

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

**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	<i>Return a dataframe that contains all line-up changes for a set of given match IDs</i>
------------------	--

---

**Description**

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

**Usage**

```
getSubstitutions(matches, token)
```

**Arguments**

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

**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)
```

---

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,  
    interval = 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, [17](#)

[getAccessToken](#), [2](#)

[getEvents](#), [3](#)

[getFormations](#), [4](#)

[getIterations](#), [4](#)

[getMatches](#), [5](#)

[getPlayerIterationAverages](#), [6](#)

[getPlayerIterationScores](#), [7](#)

[getPlayerMatchScores](#), [8](#)

[getPlayerMatchsums](#), [9](#)

[getPlayerProfileScores](#), [9](#)

[getSetPieces](#), [10](#)

[getSquadIterationAverages](#), [11](#)

[getSquadIterationScores](#), [12](#)

[getSquadMatchScores](#), [13](#)

[getSquadMatchsums](#), [14](#)

[getSquadRatings](#), [14](#)

[getStartingPositions](#), [15](#)

[getSubstitutions](#), [16](#)

TokenBucket (TokenBucket-class), [17](#)

TokenBucket-class, [17](#)