

# Brawl Stars API

Brawl Stars API

[Terms of service](#)

## **players**

Access player specific information

GET [/players/{playerTag}/battlelog](#)

Get log of recent battles for a player.

Get list of recent battle results for a player. NOTE: It may take up to 30 minutes for a new battle to appear in the battlelog.

### **Parameters**

Name	Description
playerTag * string  (path)	Tag of the player.

### **Responses**

Response content type

Code	Description
------	-------------

200

Successful response

```
•  
•  
BattleList[Battle{  
battleTime  
string  
event  
Event{  
modelId  
integer  
mode  
stringEnum:  
[ soloShowdown, duoShowdown,  
heist, bounty, siege, gemGrab,  
brawlBall, bigGame, bossFight,  
roboRumble, takedown, loneStar,  
presentPlunder, hotZone,  
superCityRampage, knockout,  
volleyBrawl, basketBrawl,  
holdTheTrophy, trophyThieves,  
duels, wipeout, payload, botDrop,  
hunters, lastStand, snowtelThieves,  
pumpkinPlunder, trophyEscape,  
wipeout5V5, knockout5V5,  
gemGrab5V5, brawlBall5V5,  
godzillaCitySmash, paintBrawl,  
trioShowdown, zombiePlunder,  
jellyfishing, unknown ]  
id  
integer  
map  
JsonLocalizedName{  
}  
}  
}  
battle  
BattleResult{  
}  
}]
```

400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>   .   . ClientError{ reason string message string type string detail { } } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>   .   . ClientError{ reason string message string type string detail { } } </pre>
battleTime	string

event	<pre> Event{   modelId   integer   mode   stringEnum:     [ soloShowdown, duoShowdown,     heist, bounty, siege, gemGrab,     brawlBall, bigGame, bossFight,     roboRumble, takedown, loneStar,     presentPlunder, hotZone,     superCityRampage, knockout,     volleyBrawl, basketBrawl,     holdTheTrophy, trophyThieves,     duels, wipeout, payload, botDrop,     hunters, lastStand, snowtelThieves,     pumpkinPlunder, trophyEscape,     wipeout5V5, knockout5V5,     gemGrab5V5, brawlBall5V5,     godzillaCitySmash, paintBrawl,     trioShowdown, zombiePlunder,     jellyfishing, unknown ]   id   integer   map   JsonLocalizedName{    } } </pre>
battle	<pre> BattleResult{   } } </pre>
modelId	integer

mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
id	integer
map	JsonLocalizedName{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

GET [/players/{playerTag}](#)

Get player information

Get information about a single player by player tag. Player tags can be found either in game or by from clan member list.

#### Parameters

Name	Description
playerTag * string  (path)	Tag of the player.

#### Responses

Response content type

Code	Description
200	Successful response • • Player{

```
club
PlayerClub{
tag
string
name
string
}
isQualifiedFromChampionshipChallenge
boolean
3vs3Victories
integer
icon
PlayerIcon{
id
integer
}
tag
string
name
string
trophies
integer
expLevel
integer
expPoints
integer
highestTrophies
integer
soloVictories
integer
duoVictories
integer
bestRoboRumbleTime
integer
bestTimeAsBigBrawler
integer
brawlers
BrawlerStatList[BrawlerStat{
gadgets
AccessoryList[Accessory{
name
```

```
JsonLocalizedName{  
}  
}  
id  
integer  
}  
starPowers  
StarPowerList[StarPower{  
name  
JsonLocalizedName{  
}  
}  
id  
integer  
}  
id  
integer  
}  
currentWinStreak  
integer  
rank  
integer  
trophies  
integer  
highestTrophies  
integer  
power  
integer  
gears  
GearStatList[GearStat{  
name  
JsonLocalizedName{  
}  
}  
id  
integer  
level  
integer  
}  
}  
maxWinStreak  
integer  
name  
JsonLocalizedName{
```

	<pre>     }   }]   nameColor   string } </pre>
400	<p>Client provided incorrect parameters for the request.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre> ClientError{   reason   string   message   string   type   string   detail   {     }   } } </pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre> ClientError{   reason   string   message   string   type   string   detail   {     }   } } </pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>   .   . ClientError{   reason   string   message   string   type   string   detail   {   }   } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>   .   . ClientError{   reason   string   message   string   type   string   detail   {   }   } </pre>
club	<pre> PlayerClub{   tag   string   name   string   } </pre>
isQualifiedFromChampionshipChallenge	boolean

3vs3Victories	integer
icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
expLevel	integer
expPoints	integer
highestTrophies	integer
soloVictories	integer
duoVictories	integer
bestRoboRumbleTime	integer
bestTimeAsBigBrawler	integer
brawlers	BrawlerStatList[BrawlerStat{ gadgets AccessoryList[Accessory{ name JsonLocalizedName{ } } id integer } ] starPowers StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer } ] id integer currentWinStreak integer rank

	integer trophies integer highestTrophies integer power integer gears GearStatList[GearStat{ name JsonLocalizedName{ } } } id integer level integer }] maxWinStreak integer name JsonLocalizedName{ } } } }]
nameColor	string
tag	string
name	string
id	integer
gadgets	AccessoryList[Accessory{ name JsonLocalizedName{ } } id integer }]

starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
id	integer
currentWinStreak	integer
rank	integer
trophies	integer
highestTrophies	integer
power	integer
gears	GearStatList[GearStat{ name JsonLocalizedName{ } } id integer level integer }]
maxWinStreak	integer
name	JsonLocalizedName{ }
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer



message	string
type	string
detail	{ }

## **clubs**

### **Access club specific information**

**GET**[/clubs/{clubTag}/members](#)

List club members.

List club members.

#### **Parameters**

Name	Description
clubTag * string  (path)	Tag of the club.
before string  (query)	Return only items that occur before this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
after string  (query)	Return only items that occur after this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
limit integer  (query)	Limit the number of items returned in the response.

#### **Responses**

Response content type

Code	Description
------	-------------

200	<p>Successful response</p> <p>•</p> <p>•</p> <pre>ClubMemberList[ClubMember{   icon   PlayerIcon{     id     integer   }   tag   string   name   string   trophies   integer   role   stringEnum:   [ notMember, member, president,   senior, vicePresident, unknown ]   nameColor   string }]</pre>
400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {     }   }</pre>

403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>.</p> <p>.</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
404	<p>Resource was not found.</p> <p>.</p> <p>.</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail {</pre> <p>}</p> <p>}</p>
500	<p>Unknown error happened when handling the request.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail {</pre> <p>}</p> <p>}</p>

503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>   .   . ClientError{   reason   string   message   string   type   string   detail   {   } } </pre>
icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
role	stringEnum: [ notMember, member, president, senior, vicePresident, unknown ]
nameColor	string
id	integer
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

GET [/clubs/{clubTag}](#)

Get club information.

Get information about a single clan by club tag. Club tags can be found in game. Note that clan tags start with hash character '#' and that needs to be URL-encoded properly to work in URL, so for example clan tag '#2ABC' would become '%232ABC' in the URL.

#### Parameters

Name	Description

clubTag *	Tag of the club.
string  (path)	

**Responses**

Response content type

Code	Description

200

Successful response

•

•

Club{  
tag  
string  
name  
string  
description  
string  
trophies  
integer  
requiredTrophies  
integer  
members  
ClubMemberList[ClubMember{  
icon  
PlayerIcon{  
id  
integer  
}  
tag  
string  
name  
string  
trophies  
integer  
role  
stringEnum:  
[ notMember, member, president,  
senior, vicePresident, unknown ]  
nameColor  
string  
}]  
type  
stringEnum:  
[ open, inviteOnly, closed, unknown ]  
badgeld  
integer  
}

400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>         .         . ClientError{ reason string message string type string detail { } } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>         .         . ClientError{ reason string message string type string detail { } } </pre>
tag	string
name	string
description	string
trophies	integer
requiredTrophies	integer

members	ClubMemberList[ClubMember{ icon PlayerIcon{ id integer } tag string name string trophies integer role stringEnum: [ notMember, member, president, senior, vicePresident, unknown ] nameColor string }]
type	stringEnum: [ open, inviteOnly, closed, unknown ]
badgeld	integer
icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
role	stringEnum: [ notMember, member, president, senior, vicePresident, unknown ]
nameColor	string
id	integer
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

## [\*\*rankings\*\*](#)

**Access global and local rankings**

GET [rankings/{countryCode}/clubs](#)

Get club rankings for a country or global rankings.

Get club rankings for a country or global rankings.

#### Parameters

Name	Description
countryCode * string  (path)	Two letter country code, or 'global' for global rankings.
before string  (query)	Return only items that occur before this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
after string  (query)	Return only items that occur after this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
limit integer  (query)	Limit the number of items returned in the response.

#### Responses

Response content type

Code	Description

200	<p>Successful response</p> <p>•</p> <p>•</p> <p>ClubRankingList[ClubRanking{ tag string name string trophies integer rank integer memberCount integer badgeld integer }]</p>
400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <p>ClientError{ reason string message string type string detail { } }</p>

403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <pre> . . ClientError{   reason   string   message   string   type   string   detail { } } </pre>
404	<p>Resource was not found.</p> <pre> . . ClientError{   reason   string   message   string   type   string   detail { } } </pre>

429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail {</pre> <p>}</p> <p>}</p>
500	<p>Unknown error happened when handling the request.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail {</pre> <p>}</p> <p>}</p>

503	<p>Service is temporarily unavailable because of maintenance.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
tag	string
name	string
trophies	integer
rank	integer
memberCount	integer
badgeld	integer
reason	string
message	string
type	string
detail	<pre>{ }</pre>
reason	string
message	string
type	string
detail	<pre>{ }</pre>
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

GET [/rankings/{countryCode}/brawlers/{brawlerId}](#)

Get brawler rankings for a country or global rankings.

Get brawler rankings for a country or global rankings. Brawler identifiers can be found by using the /v1/brawlers API endpoint.

#### Parameters

Name	Description
countryCode * string (path)	Two letter country code, or 'global' for global rankings.
brawlerId * string (path)	Identifier of the brawler.

before string  (query)	Return only items that occur before this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
after string  (query)	Return only items that occur after this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
limit integer  (query)	Limit the number of items returned in the response.

### Responses

Response content type

Code	Description

200

Successful response

•  
•  
PlayerRankingList[PlayerRanking{  
club  
PlayerRankingClub{  
name  
string  
}  
trophies  
integer  
icon  
PlayerIcon{  
id  
integer  
}  
tag  
string  
name  
string  
rank  
integer  
nameColor  
string  
}]

400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>         .         . ClientError{ reason string message string type string detail { } } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>         .         . ClientError{ reason string message string type string detail { } } </pre>
club	PlayerRankingClub{ name string }
trophies	integer

icon	PlayerIcon{ id integer }
tag	string
name	string
rank	integer
nameColor	string
name	string
id	integer
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string

type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

GET [/rankings/{countryCode}/players](#)

Get player rankings for a country or global rankings.

Get player rankings for a country or global rankings.

#### Parameters

Name	Description
countryCode * string  (path)	Two letter country code, or 'global' for global rankings.
before string  (query)	Return only items that occur before this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
after string  (query)	Return only items that occur after this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
limit integer  (query)	Limit the number of items returned in the response.

#### Responses

Response content type

Code	Description
------	-------------

200

Successful response

•  
•  
PlayerRankingList[PlayerRanking{  
club  
PlayerRankingClub{  
name  
string  
}  
trophies  
integer  
icon  
PlayerIcon{  
id  
integer  
}  
tag  
string  
name  
string  
rank  
integer  
nameColor  
string  
}]

400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>   .   . ClientError{ reason string message string type string detail { } } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>   .   . ClientError{ reason string message string type string detail { } } </pre>
club	PlayerRankingClub{ name string }
trophies	integer

icon	PlayerIcon{ id integer }
tag	string
name	string
rank	integer
nameColor	string
name	string
id	integer
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string

type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

## [brawlers](#)

### Access general brawler information

GET [/brawlers](#)

Get list of available brawlers.

Get list of available brawlers.

#### Parameters

Name	Description
before string (query)	Return only items that occur before this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
after string (query)	Return only items that occur after this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
limit integer (query)	Limit the number of items returned in the response.

#### Responses

Response content type

Code	Description
------	-------------

200

Successful response

•

•

BrawlerList[Brawler{  
gadgets  
AccessoryList[Accessory{  
name  
JsonLocalizedName{  
}  
id  
integer  
}]  
name  
JsonLocalizedName{  
}  
id  
integer  
starPowers  
StarPowerList[StarPower{  
name  
JsonLocalizedName{  
}  
id  
integer  
}]  
}]

400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>   .   . ClientError{ reason string message string type string detail { } } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>   .   . ClientError{ reason string message string type string detail { } } </pre>
gadgets	<pre> AccessoryList[Accessory{ name JsonLocalizedName{  } id integer }] </pre>

name	JsonLocalizedName{ }
id	integer
starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer
reason	string
message	string
type	string
detail	{ } }
reason	string
message	string
type	string
detail	{ } }
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

GET [/brawlers/{brawlerId}](#)

Get information about a brawler.

Get information about a brawler.

#### Parameters

Name	Description
brawlerId * string (path)	Identifier of the brawler.

#### Responses

Response content type

Code	Description
------	-------------

200

Successful response

•

•

Brawler{  
gadgets  
AccessoryList[Accessory{  
name  
JsonLocalizedName{  
}  
id  
integer  
}]  
name  
JsonLocalizedName{  
}  
id  
integer  
starPowers  
StarPowerList[StarPower{  
name  
JsonLocalizedName{  
}  
id  
integer  
}]  
}

400	<p>Client provided incorrect parameters for the request.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>         .         . ClientError{ reason string message string type string detail { } } </pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>         .         . ClientError{ reason string message string type string detail { } } </pre>
gadgets	<pre> AccessoryList[Accessory{ name JsonLocalizedName{  } id integer }] </pre>

name	JsonLocalizedName{ }
id	integer
starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer
reason	string
message	string
type	string
detail	{ } }
reason	string
message	string
type	string
detail	{ } }
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

## events

### GET/[gamemodes](#)

Get list of all available game modes.

Get list of all available game modes.

#### Parameters

Name	Description
before string (query)	Return only items that occur before this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.

after string  (query)	Return only items that occur after this marker. Before marker can be found from the response, inside the 'paging' property. Note that only after or before can be specified for a request, not both.
limit integer  (query)	Limit the number of items returned in the response.

### Responses

Response content type

Code	Description
200	<p>Successful response</p> <p>•</p> <p>•</p> <p>EventTypeList[EventType{ name JsonLocalizedNames{ } } id integer }]</p>

400	<p>Client provided incorrect parameters for the request.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>
403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail   {   } }</pre>

404	<p>Resource was not found.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>

500	<p>Unknown error happened when handling the request.</p> <pre>   .   . ClientError{   reason   string   message   string   type   string   detail   {   }   }</pre>
503	<p>Service is temporarily unavailable because of maintenance.</p> <pre>   .   . ClientError{   reason   string   message   string   type   string   detail   {   }   }</pre>
name	JsonLocalizedName{ }
id	integer
reason	string
message	string
type	string

detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

GET [/events/rotation](#)

Get event rotation

Get event rotation for ongoing events.

#### Parameters

No parameters

## Responses

Response content type

Code	Description
200	<p>Successful response</p> <p>•</p> <p>•</p> <p>ScheduledEvents[ScheduledEvent{</p> <p>slotId integer</p> <p>event ScheduledEventLocation{</p> <p>modelId integer</p> <p>mode stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]</p> <p>modifiers</p> <p>EventModifierList[stringEnum: [ unknown, none, energyDrink, angryRobo, meteorShower, graveyardShift, healingMushrooms, bossFightRockets, takedownLasers, takedownChainLightning, takedownRockets, waves,</p>

	<pre> hauntedBall, superCharge, fastBrawlers, showdown+,  peekABoo, burningBall ]] id integer map JsonLocalizedName{  } } startTime string endTime string }] </pre>
400	<p>Client provided incorrect parameters for the request.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre> ClientError{ reason string message string type string detail { } } </pre>

403	<p>Access denied, either because of missing/incorrect credentials or used API token does not grant access to the requested resource.</p> <pre> . . ClientError{   reason   string   message   string   type   string   detail { } } </pre>
404	<p>Resource was not found.</p> <pre> . . ClientError{   reason   string   message   string   type   string   detail { } } </pre>

429	<p>Request was throttled, because amount of requests was above the threshold defined for the used API token.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail {</pre> <p>}</p> <p>}</p>
500	<p>Unknown error happened when handling the request.</p> <ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul> <pre>ClientError{   reason   string   message   string   type   string   detail {</pre> <p>}</p> <p>}</p>

503	<p>Service is temporarily unavailable because of maintenance.</p> <p>•</p> <p>•</p> <pre>ClientError{   reason   string   message   string   type   string   detail   {   }   } }</pre>
slotId	integer

event	<pre> ScheduledEventLocation{   modelId   integer   mode   stringEnum:     [ soloShowdown, duoShowdown,     heist, bounty, siege, gemGrab,     brawlBall, bigGame, bossFight,     roboRumble, takedown, loneStar,     presentPlunder, hotZone,     superCityRampage, knockout,     volleyBrawl, basketBrawl,     holdTheTrophy, trophyThieves,     duels, wipeout, payload, botDrop,     hunters, lastStand, snowtelThieves,     pumpkinPlunder, trophyEscape,     wipeout5V5, knockout5V5,     gemGrab5V5, brawlBall5V5,     godzillaCitySmash, paintBrawl,     trioShowdown, zombiePlunder,     jellyfishing, unknown ]   modifiers   EventModifierList[stringEnum:     [ unknown, none, energyDrink,     angryRobo, meteorShower,     graveyardShift, healingMushrooms,     bossFightRockets, takedownLasers,     takedownChainLightning,     takedownRockets, waves,     hauntedBall, superCharge,     fastBrawlers, showdown+,     peekABoo, burningBall ]]   id   integer   map   JsonLocalizedName{     }   } } </pre>
startTime	string

endTime	string
modelId	integer
mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
modifiers	EventModifierList[stringEnum: [ unknown, none, energyDrink, angryRobo, meteorShower, graveyardShift, healingMushrooms, bossFightRockets, takedownLasers, takedownChainLightning, takedownRockets, waves, hauntedBall, superCharge, fastBrawlers, showdown+, peekABoo, burningBall ]]
id	integer
map	JsonLocalizedName{ }
reason	string
message	string
type	string
detail	{ }

reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }
reason	string
message	string
type	string
detail	{ }

## Models

PlayerRankingList[PlayerRankingList{

club	PlayerRankingClub{ name string }
trophies	integer

icon	PlayerIcon{ id integer }
tag	string
name	string
rank	integer
nameColor	string
name	string
id	integer
}	
PlayerRanking{	
club	PlayerRankingClub{ name string }
trophies	integer
icon	PlayerIcon{ id integer }
tag	string
name	string
rank	integer
nameColor	string
name	string
id	integer
}	
PlayerIcon{	
id	integer
}	
PlayerRankingClub{	
name	string
}	
ServiceVersion{	
major	integer
minor	integer

content	integer
}	
Brawler{	
gadgets	AccessoryList[Accessory{ name JsonLocalizedName{ } id integer }]
name	JsonLocalizedName{ }
id	integer
starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
}	
StarPowerList[StarPowerList{	
name	JsonLocalizedName{ }
id	integer
}]	
StarPower{	

name	JsonLocalizedName{ }
id	integer
}	
JsonLocalizedName{ }	
}	
AccessoryList[AccessoryList{ name	JsonLocalizedName{ }
id	integer
}]	
Accessory{ name	JsonLocalizedName{ }
id	integer
}	
BattleRegionList[BattleRegionList{ id	integer
name	string
}]	
BattleRegion{ id	integer
name	string
}	
ClubRankingList[ClubRankingList{ tag	string
name	string
trophies	integer
rank	integer
memberCount	integer
badgeld	integer
}]	
ClubRanking{ tag	string

name	string
trophies	integer
rank	integer
memberCount	integer
badgeld	integer
}	
RegisterMatchRequest{	
mode	stringEnum: [ regular, powerMatch ]
players	RegisterMatchRequestPlayers[Playe rEntry{ tag string side integer }]
locationId	integer
winsRequired	integer
gadgetsAllowed	boolean
bannedBrawlers	BannedBrawlerList[BannedBrawlerE ntry{ id integer side integer }]
timerPreset	stringEnum: [ default, longer, longest ]
tag	string
side	integer
id	integer
side	integer
}	
BannedBrawlerList[BannedBrawlerList{	
id	integer
side	integer
}]	
BannedBrawlerEntry{	

id	integer
side	integer
}	
RegisterMatchRequestPlayers[RegisterMatchRequestPlayers{	
tag	string
side	integer
}]	
PlayerEntry{	
tag	string
side	integer
}	
RegisterMatchResponse{	
id	string
}	
MatchLocationList[MatchLocationList{	
id	integer
name	string
gameMode	string
}]	
MatchLocation{	
id	integer
name	string
gameMode	string
}	
SetEsportsNotificationRequest{	
type	stringEnum: [ hub, unknown ]
players	{ }
ttl	integer
}	
SetEsportsNotificationResponse{	

notification	<pre>SetEsportsNotificationRequest{   type   stringEnum:   [ hub, unknown ]   players   {     }   ttl   integer }</pre>
status	string
type	<pre>stringEnum: [ hub, unknown ]</pre>
players	<pre>{   }</pre>
ttl	integer

}

Club{	
tag	string
name	string
description	string
trophies	integer
requiredTrophies	integer

members	ClubMemberList[ClubMember{ icon PlayerIcon{ id integer } tag string name string trophies integer role stringEnum: [ notMember, member, president, senior, vicePresident, unknown ] nameColor string }]
type	stringEnum: [ open, inviteOnly, closed, unknown ]
badgeld	integer
icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
role	stringEnum: [ notMember, member, president, senior, vicePresident, unknown ]
nameColor	string
id	integer

}

ClubMemberList[ClubMemberList{

icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
role	stringEnum: [ notMember, member, president, senior, vicePresident, unknown ]
nameColor	string
id	integer

}]

ClubMember{

icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
role	stringEnum: [ notMember, member, president, senior, vicePresident, unknown ]
nameColor	string
id	integer

}

ScheduledEvents[ScheduledEvents{

slotId	integer
--------	---------

event	<pre> ScheduledEventLocation{   modelId   integer   mode   stringEnum:     [ soloShowdown, duoShowdown,     heist, bounty, siege, gemGrab,     brawlBall, bigGame, bossFight,     roboRumble, takedown, loneStar,     presentPlunder, hotZone,     superCityRampage, knockout,     volleyBrawl, basketBrawl,     holdTheTrophy, trophyThieves,     duels, wipeout, payload, botDrop,     hunters, lastStand, snowtelThieves,     pumpkinPlunder, trophyEscape,     wipeout5V5, knockout5V5,     gemGrab5V5, brawlBall5V5,     godzillaCitySmash, paintBrawl,     trioShowdown, zombiePlunder,     jellyfishing, unknown ]   modifiers   EventModifierList[stringEnum:     [ unknown, none, energyDrink,     angryRobo, meteorShower,     graveyardShift, healingMushrooms,     bossFightRockets, takedownLasers,     takedownChainLightning,     takedownRockets, waves,     hauntedBall, superCharge,     fastBrawlers, showdown+,     peekABoo, burningBall ]]   id   integer   map   JsonLocalizedName{     }   } } </pre>
startTime	string

endTime	string
modelId	integer
mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
modifiers	EventModifierList[stringEnum: [ unknown, none, energyDrink, angryRobo, meteorShower, graveyardShift, healingMushrooms, bossFightRockets, takedownLasers, takedownChainLightning, takedownRockets, waves, hauntedBall, superCharge, fastBrawlers, showdown+, peekABoo, burningBall ]]
id	integer
map	JsonLocalizedName{ }

}

ScheduledEvent{

slotId	integer
--------	---------

event	<pre> ScheduledEventLocation{   modelId   integer   mode   stringEnum:     [ soloShowdown, duoShowdown,     heist, bounty, siege, gemGrab,     brawlBall, bigGame, bossFight,     roboRumble, takedown, loneStar,     presentPlunder, hotZone,     superCityRampage, knockout,     volleyBrawl, basketBrawl,     holdTheTrophy, trophyThieves,     duels, wipeout, payload, botDrop,     hunters, lastStand, snowtelThieves,     pumpkinPlunder, trophyEscape,     wipeout5V5, knockout5V5,     gemGrab5V5, brawlBall5V5,     godzillaCitySmash, paintBrawl,     trioShowdown, zombiePlunder,     jellyfishing, unknown ]   modifiers   EventModifierList[stringEnum:     [ unknown, none, energyDrink,     angryRobo, meteorShower,     graveyardShift, healingMushrooms,     bossFightRockets, takedownLasers,     takedownChainLightning,     takedownRockets, waves,     hauntedBall, superCharge,     fastBrawlers, showdown+,     peekABoo, burningBall ]]   id   integer   map   JsonLocalizedName{     }   } } </pre>
startTime	string

endTime	string
modelId	integer
mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
modifiers	EventModifierList[stringEnum: [ unknown, none, energyDrink, angryRobo, meteorShower, graveyardShift, healingMushrooms, bossFightRockets, takedownLasers, takedownChainLightning, takedownRockets, waves, hauntedBall, superCharge, fastBrawlers, showdown+, peekABoo, burningBall ]]
id	integer
map	JsonLocalizedName{ }

}

ScheduledEventLocation{

modelId	integer
---------	---------

mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
modifiers	EventModifierList[stringEnum: [ unknown, none, energyDrink, angryRobo, meteorShower, graveyardShift, healingMushrooms, bossFightRockets, takedownLasers, takedownChainLightning, takedownRockets, waves, hauntedBall, superCharge, fastBrawlers, showdown+, peekABoo, burningBall ]]]
id	integer
map	JsonLocalizedName{ }

}

EventModifierList[stringEnum:  
[ unknown, none, energyDrink, angryRobo, meteorShower, graveyardShift, healingMushrooms, bossFightRockets, takedownLasers, takedownChainLightning, takedownRockets, waves, hauntedBall, superCharge, fastBrawlers, showdown+, peekABoo, burningBall ]]]

EventModifier{

}

Player{

club	PlayerClub{ tag string name string }
isQualifiedFromChampionshipChallenge	boolean
3vs3Victories	integer
icon	PlayerIcon{ id integer }
tag	string
name	string
trophies	integer
expLevel	integer
expPoints	integer
highestTrophies	integer
soloVictories	integer
duoVictories	integer
bestRoboRumbleTime	integer
bestTimeAsBigBrawler	integer
brawlers	BrawlerStatList[BrawlerStat{ gadgets AccessoryList[Accessory{ name JsonLocalizedName{ } } id integer } ] starPowers StarPowerList[StarPower{ name JsonLocalizedName{ } }

	id integer }] id integer currentWinStreak integer rank integer trophies integer highestTrophies integer power integer gears GearStatList[GearStat{ name JsonLocalizedName{ } id integer level integer }] maxWinStreak integer name JsonLocalizedName{ } } ]
nameColor	string
tag	string
name	string
id	integer

gadgets	AccessoryList[Accessory{ name JsonLocalizedName{ } } id integer }]
starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
id	integer
currentWinStreak	integer
rank	integer
trophies	integer
highestTrophies	integer
power	integer
gears	GearStatList[GearStat{ name JsonLocalizedName{ } } id integer level integer }]
maxWinStreak	integer
name	JsonLocalizedName{ }

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer

}

BrawlerStatList[BrawlerStatList{

gadgets	AccessoryList[Accessory{ name JsonLocalizedName{ } } id integer }]
starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
id	integer
currentWinStreak	integer
rank	integer
trophies	integer
highestTrophies	integer
power	integer

gears	<pre> GearStatList[GearStat{   name   JsonLocalizedName{  }    id   integer   level   integer }]  </pre>
maxWinStreak	integer
name	<pre> JsonLocalizedName{  }  </pre>
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
level	integer
}]	
BrawlerStat{	
gadgets	<pre> AccessoryList[Accessory{   name   JsonLocalizedName{  }    id   integer }]  </pre>

starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
id	integer
currentWinStreak	integer
rank	integer
trophies	integer
highestTrophies	integer
power	integer
gears	GearStatList[GearStat{ name JsonLocalizedName{ } } id integer level integer }]
maxWinStreak	integer
name	JsonLocalizedName{ }
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer

name	JsonLocalizedName{ }
id	integer
level	integer
}	
GearStatList[GearStatList{	
name	JsonLocalizedName{ }
id	integer
level	integer
}]	
GearStat{	
name	JsonLocalizedName{ }
id	integer
level	integer
}	
PlayerClub{	
tag	string
name	string
}	
EventTypeList[EventTypeList{	
name	JsonLocalizedName{ }
id	integer
}]	
EventType{	
name	JsonLocalizedName{ }
id	integer
}	
Match{	
games	CompletedGameList[CompletedGame{

```
teams
CompletedGameTeamList[Complete
dGameTeam{
score
integer
isWinner
boolean
siege
SiegeStats{
botDamageToBase
integer
botLevelByRound
{
}

}
}
players
PlayerEntryCompletedGameList[Play
erEntryCompletedGame{
brawler
BrawlerInfo{
gadget
Accessory{
name
JsonLocalizedName{

}
id
integer
}
starPower
StarPower{
name
JsonLocalizedName{

}
id
integer
}
gears
GearInfoList[GearInfo{
name
```

```
JsonLocalizedNames{  
    }  
    id  
    integer  
    level  
    integer  
    }]  
    trophyChange  
    integer  
    trophies  
    integer  
    power  
    integer  
    name  
    JsonLocalizedNames{  
        }  
        id  
        integer  
    }  
    statistics  
    Stats{  
        siegeBoltsCollected  
        integer  
        brawlBallGoalsScored  
        integer  
        gemGrabGemsCollected  
        integer  
        gemGrabGemsLost  
        integer  
        bountyStarsGained  
        integer  
        bountyStarsLost  
        integer  
        superUsedCount  
        integer  
        gadgetUsedCount  
        integer  
        bountyPickedMiddleStar  
        boolean  
        matchEndKillStreak
```

```
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
integer
damageDealt
integer
kills
integer
averageLatency
integer
damageReceived
integer
totalDamageToSafe
integer
totalDamageToPets
integer
siegeDamageToRobot
integer
maxKillStreak
integer
}
tag
string
accountId
string
}]
}]
```

	duration integer location MatchLocation{ id integer name string gameMode string } replayId string }]
phase	stringEnum: [ init, banHeroes, pickHeroes, finalPreparation, battle, matchResult, ending ]
initiativeSide	integer
round	integer
teams	MatchTeamList[MatchTeam{ bans BrawlerInfoList[BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } id integer }

```
gears
GearInfoList[GearInfo{
name
JsonLocalizedName{

}

id
integer
level
integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}

id
integer
}]
players
MatchTeamPlayerList[MatchTeamPlayer{
causedTermination
boolean
tag
string
isLeader
boolean
brawler
BrawlerInfo{
gadget
Accessory{
name
JsonLocalizedName{

}

id
```

```
    integer
    }
    starPower
    StarPower{
        name
        JsonLocalizedName{

    }
    id
    integer
    }
    gears
    GearInfoList[GearInfo{
        name
        JsonLocalizedName{

    }
    id
    integer
    level
    integer
    }]
    trophyChange
    integer
    trophies
    integer
    power
    integer
    name
    JsonLocalizedName{

    }
    id
    integer
    }
    }]
    side
    integer
    }]
```

terminationReason	stringEnum: [ none, playerDisconnected, playerNotResponding, technicalError, matchTooLong, other ]
players	PlayerMatchStatusList[PlayerMatchStatus{   isOnline   boolean   isInBattle   boolean   hasJoined   boolean   isReady   boolean   brawler   BrawlerInfo{     gadget     Accessory{       name       JsonLocalizedName{         }       id       integer     }     starPower     StarPower{       name       JsonLocalizedName{         }       id       integer     }     gears     GearInfoList[GearInfo{       name       JsonLocalizedName{         }     }   } } ]

	<pre> id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } tag string }] </pre>
state	stringEnum: [ open, cancelled, completed ]
id	string
teams	<pre> CompletedGameTeamList[Complete dGameTeam{ score integer isWinner boolean siege SiegeStats{ botDamageToBase integer botLevelByRound { }  } } players PlayerEntryCompletedGameList[Play </pre>

```
erEntryCompletedGame{
brawler
BrawlerInfo{
gadget
Accessory{
name
JsonLocalizedName{

}
id
integer
}
starPower
StarPower{
name
JsonLocalizedName{

}
id
integer
}
gears
GearInfoList[GearInfo{
name
JsonLocalizedName{

}
id
integer
level
integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}
}
```

```
id
integer
}
statistics
Stats{
siegeBoltsCollected
integer
brawlBallGoalsScored
integer
gemGrabGemsCollected
integer
gemGrabGemsLost
integer
bountyStarsGained
integer
bountyStarsLost
integer
superUsedCount
integer
gadgetUsedCount
integer
bountyPickedMiddleStar
boolean
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
```

	<pre> integer damageDealt integer kills integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer } tag string accountId string }] }] </pre>
duration	integer
location	<pre> MatchLocation{ id integer name string gameMode string } </pre>
replayId	string
score	integer
isWinner	boolean

siege	SiegeStats{ botDamageToBase integer botLevelByRound { } } }
players	PlayerEntryCompletedGameList[Play erEntryCompletedGame{ brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }]

```
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}

id
integer
}
statistics
Stats{
siegeBoltsCollected
integer
brawlBallGoalsScored
integer
gemGrabGemsCollected
integer
gemGrabGemsLost
integer
bountyStarsGained
integer
bountyStarsLost
integer
superUsedCount
integer
gadgetUsedCount
integer
bountyPickedMiddleStar
boolean
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
```

	integer objectivesStolen integer brawlBallShotsOnGoal integer brawlBallShotsSaved integer healingDone integer deaths integer damageDealt integer kills integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer } tag string accountId string }]
botDamageToBase	integer
botLevelByRound	{ }
brawler	BrawlerInfo{ gadget Accessory{ name

	JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{ } id integer } 
statistics	Stats{ siegeBoltsCollected }

```
integer
brawlBallGoalsScored
integer
gemGrabGemsCollected
integer
gemGrabGemsLost
integer
bountyStarsGained
integer
bountyStarsLost
integer
superUsedCount
integer
gadgetUsedCount
integer
bountyPickedMiddleStar
boolean
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
integer
damageDealt
integer
kills
integer
averageLatency
```

	integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer }
tag	string
accountId	string
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } id integer level integer }]

trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer
gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer

brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer
siegeDamageToRobot	integer
maxKillStreak	integer
id	integer
name	string
gameMode	string
bans	<pre>BrawlerInfoList[BrawlerInfo{     gadget     Accessory{         name         JsonLocalizedName{      }     id     integer } starPower StarPower{     name     JsonLocalizedName{  } id integer } gears GearInfoList[GearInfo{     name     JsonLocalizedName{  } }</pre>

	<pre> id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  } id integer }] </pre>
players	<pre> MatchTeamPlayerList[MatchTeamPlayer{ causedTermination boolean tag string isLeader boolean brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{  } id integer } starPower StarPower{ name JsonLocalizedName{ </pre>

	<pre> } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{  } id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  } id integer } ] </pre>
side	integer
gadget	<pre> Accessory{ name JsonLocalizedName{  } id integer } </pre>

starPower	StarPower{ name JsonLocalizedName{ } } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } ] id integer level integer ] ]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer
level	integer

causedTermination	boolean
tag	string
isLeader	boolean
brawler	<pre> BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{  }  id integer } starPower StarPower{ name JsonLocalizedName{  }  id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{  }  id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{ </pre>

	<pre>     }     id     integer   } </pre>
gadget	<pre> Accessory{   name   JsonLocalizedName{     }     id     integer   } </pre>
starPower	<pre> StarPower{   name   JsonLocalizedName{     }     id     integer   } </pre>
gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }     id     integer     level     integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
isOnline	boolean
isInBattle	boolean
hasJoined	boolean
isReady	boolean
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name

	<pre> JsonLocalizedName{  }  id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
tag	string
gadget	<pre> Accessory{ name JsonLocalizedName{  }  id integer } </pre>
starPower	<pre> StarPower{ name JsonLocalizedName{  }  id integer } </pre>

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }   id   integer   level   integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
level	integer
}	
PlayerMatchStatusList[PlayerMatchStatusList{	
isOnline	boolean
isInBattle	boolean
hasJoined	boolean
isReady	boolean
brawler	BrawlerInfo{   gadget }

```
Accessory{
  name
  JsonLocalizedName{

  }
  id
  integer
}
starPower
StarPower{
  name
  JsonLocalizedName{

  }
  id
  integer
}
gears
GearInfoList[GearInfo{
  name
  JsonLocalizedName{

  }
  id
  integer
  level
  integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}
id
integer
}
```

tag	string
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
}	
PlayerMatchStatus{	
isOnline	boolean
isInBattle	boolean
hasJoined	boolean
isReady	boolean
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } }

	<pre> id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  } id integer } </pre>
tag	string
gadget	<pre> Accessory{ name JsonLocalizedName{  } id integer } </pre>
starPower	<pre> StarPower{ name JsonLocalizedName{  } id integer } </pre>

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }   id   integer   level   integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
level	integer
}	
BrawlerInfo{	

gadget	Accessory{ name JsonLocalizedName{ } } id integer }
starPower	StarPower{ name JsonLocalizedName{ } } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
}	
GearInfoList[GearInfoList{	
name	JsonLocalizedName{ }
id	integer
level	integer
}]	
GearInfo{	
name	JsonLocalizedName{ }
id	integer
level	integer
}	
MatchTeamList[MatchTeamList{	
bans	BrawlerInfoList[BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{

	<pre> } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{  } id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  } id integer }] </pre>
players	<pre> MatchTeamPlayerList[MatchTeamPlayer{ causedTermination boolean tag string isLeader boolean brawler BrawlerInfo{ gadget Accessory{ </pre>

```
name
JsonLocalizedName{

}

id
integer
}
starPower
StarPower{
name
JsonLocalizedName{

}

id
integer
}
gears
GearInfoList[GearInfo{
name
JsonLocalizedName{

}

id
integer
level
integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}

id
integer
}
}]
```

side	integer
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
causedTermination	boolean
tag	string
isLeader	boolean
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } } ] id integer level

	<pre> integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
gadget	<pre> Accessory{ name JsonLocalizedName{  }  id integer } </pre>
starPower	<pre> StarPower{ name JsonLocalizedName{  }  id integer } </pre>

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{  }    id   integer   level   integer }]  </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
name	<pre> JsonLocalizedName{  }  </pre>
id	integer
level	integer
}]	
MatchTeam{	
bans	<pre> BrawlerInfoList[BrawlerInfo{   gadget   Accessory{     name     JsonLocalizedName{  }  </pre>

	<pre> id integer } starPower StarPower{ name JsonLocalizedName{  }  id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{  }  id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer }] </pre>
players	<pre> MatchTeamPlayerList[MatchTeamPlayer{ causedTermination boolean tag </pre>

```
    string
    isLeader
    boolean
    brawler
    BrawlerInfo{
        gadget
        Accessory{
            name
            JsonLocalizedName{

        }
        id
        integer
    }
    starPower
    StarPower{
        name
        JsonLocalizedName{

    }
    id
    integer
}
gears
GearInfoList[GearInfo{
    name
    JsonLocalizedName{

}
    id
    integer
    level
    integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{
```

	<code>}</code> <code>id</code> <code>integer</code> <code>}</code> <code>]</code>
side	<code>integer</code>
gadget	<code>Accessory{</code> <code>name</code> <code>JsonLocalizedName{</code> <code>}</code> <code>id</code> <code>integer</code> <code>}</code>
starPower	<code>StarPower{</code> <code>name</code> <code>JsonLocalizedName{</code> <code>}</code> <code>id</code> <code>integer</code> <code>}</code>
gears	<code>GearInfoList[GearInfo{</code> <code>name</code> <code>JsonLocalizedName{</code> <code>}</code> <code>id</code> <code>integer</code> <code>level</code> <code>integer</code> <code>}]</code>
trophyChange	<code>integer</code>
trophies	<code>integer</code>
power	<code>integer</code>
name	<code>JsonLocalizedName{</code> <code>}</code>

id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
causedTermination	boolean
tag	string
isLeader	boolean
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name

	<pre> JsonLocalizedName{  }  id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
gadget	<pre> Accessory{ name JsonLocalizedName{  }  id integer } </pre>
starPower	<pre> StarPower{ name JsonLocalizedName{  }  id integer } </pre>

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }   id   integer   level   integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
level	integer
}	
MatchTeamPlayerList[MatchTeamPlayerList{	
causedTermination	boolean
tag	string
isLeader	boolean
brawler	<pre> BrawlerInfo{   gadget   Accessory{     name   } } </pre>

```
JsonLocalizedName{  
}  
}  
id  
integer  
}  
starPower  
StarPower{  
name  
JsonLocalizedName{  
}  
}  
id  
integer  
}  
gears  
GearInfoList[GearInfo{  
name  
JsonLocalizedName{  
}  
}  
id  
integer  
level  
integer  
}]  
trophyChange  
integer  
trophies  
integer  
power  
integer  
name  
JsonLocalizedName{  
}  
}  
id  
integer  
}
```

gadget	Accessory{ name JsonLocalizedName{ } } id integer }
starPower	StarPower{ name JsonLocalizedName{ } } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
}]	
MatchTeamPlayer{	
causedTermination	boolean
tag	string
isLeader	boolean
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } } id

	<pre> integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
gadget	<pre> Accessory{ name JsonLocalizedName{  }  id integer } </pre>
starPower	<pre> StarPower{ name JsonLocalizedName{  }  id integer } </pre>

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }   id   integer   level   integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
level	integer
}	
BrawlerInfoList[BrawlerInfoList{	

gadget	Accessory{ name JsonLocalizedName{ } } id integer }
starPower	StarPower{ name JsonLocalizedName{ } } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
}]	
CompletedGameList[CompletedGameList{	
teams	CompletedGameTeamList[Complete dGameTeam{ score integer isWinner boolean siege SiegeStats{ botDamageToBase integer botLevelByRound { } } } players PlayerEntryCompletedGameList[Play erEntryCompletedGame{ brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower

```
StarPower{
  name
  JsonLocalizedName{

  }
  id
  integer
  }
  gears
  GearInfoList[GearInfo{
    name
    JsonLocalizedName{

    }
    id
    integer
    level
    integer
  }]
  trophyChange
  integer
  trophies
  integer
  power
  integer
  name
  JsonLocalizedName{

  }
  id
  integer
  }
  statistics
  Stats{
    siegeBoltsCollected
    integer
    brawlBallGoalsScored
    integer
    gemGrabGemsCollected
    integer
    gemGrabGemsLost
    integer
```

bountyStarsGained  
integer  
bountyStarsLost  
integer  
superUsedCount  
integer  
gadgetUsedCount  
integer  
bountyPickedMiddleStar  
boolean  
matchEndKillStreak  
integer  
hotZoneInsideZonePercentage  
integer  
healingDoneToSelf  
integer  
healingDoneToTeamMates  
integer  
objectivesRecovered  
integer  
objectivesStolen  
integer  
brawlBallShotsOnGoal  
integer  
brawlBallShotsSaved  
integer  
healingDone  
integer  
deaths  
integer  
damageDealt  
integer  
kills  
integer  
averageLatency  
integer  
damageReceived  
integer  
totalDamageToSafe  
integer  
totalDamageToPets  
integer

	siegeDamageToRobot integer maxKillStreak integer } tag string accountId string }] }]
duration	integer
location	MatchLocation{ id integer name string gameMode string }
replayId	string
score	integer
isWinner	boolean
siege	SiegeStats{ botDamageToBase integer botLevelByRound { } }
players	PlayerEntryCompletedGameList[PlayerEntryCompletedGame{ brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{

```
    }
    id
    integer
  }
  starPower
  StarPower{
    name
    JsonLocalizedName{

    }
    id
    integer
  }
  gears
  GearInfoList[GearInfo{
    name
    JsonLocalizedName{

    }
    id
    integer
    level
    integer
  }]
  trophyChange
  integer
  trophies
  integer
  power
  integer
  name
  JsonLocalizedName{

    }
    id
    integer
  }
  statistics
  Stats{
    siegeBoltsCollected
    integer
```

brawlBallGoalsScored  
integer  
gemGrabGemsCollected  
integer  
gemGrabGemsLost  
integer  
bountyStarsGained  
integer  
bountyStarsLost  
integer  
superUsedCount  
integer  
gadgetUsedCount  
integer  
bountyPickedMiddleStar  
boolean  
matchEndKillStreak  
integer  
hotZoneInsideZonePercentage  
integer  
healingDoneToSelf  
integer  
healingDoneToTeamMates  
integer  
objectivesRecovered  
integer  
objectivesStolen  
integer  
brawlBallShotsOnGoal  
integer  
brawlBallShotsSaved  
integer  
healingDone  
integer  
deaths  
integer  
damageDealt  
integer  
kills  
integer  
averageLatency  
integer

	<pre> damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer } tag string accountId string }] </pre>
botDamageToBase	integer
botLevelByRound	<pre>{ }</pre>
brawler	<pre> BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{  } id integer } starPower StarPower{ name JsonLocalizedName{  } id integer } gears GearInfoList[GearInfo{ </pre>

	<pre> name JsonLocalizedName{  }  id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
statistics	<pre> Stats{ siegeBoltsCollected integer brawlBallGoalsScored integer gemGrabGemsCollected integer gemGrabGemsLost integer bountyStarsGained integer bountyStarsLost integer superUsedCount integer gadgetUsedCount integer bountyPickedMiddleStar boolean } </pre>

	matchEndKillStreak integer hotZoneInsideZonePercentage integer healingDoneToSelf integer healingDoneToTeamMates integer objectivesRecovered integer objectivesStolen integer brawlBallShotsOnGoal integer brawlBallShotsSaved integer healingDone integer deaths integer damageDealt integer kills integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer }
tag	string
accountId	string

gadget	Accessory{ name JsonLocalizedName{ } } id integer }
starPower	StarPower{ name JsonLocalizedName{ } } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer
gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer
siegeDamageToRobot	integer

maxKillStreak	integer
id	integer
name	string
gameMode	string
}]	
CompletedGame{	
teams	CompletedGameTeamList[Complete dGameTeam{ score integer isWinner boolean siege SiegeStats{ botDamageToBase integer botLevelByRound { } } } players PlayerEntryCompletedGameList[Play erEntryCompletedGame{ brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } }

```
id
integer
}
gears
GearInfoList[GearInfo{
name
JsonLocalizedName{

}
id
integer
level
integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}
id
integer
}
statistics
Stats{
siegeBoltsCollected
integer
brawlBallGoalsScored
integer
gemGrabGemsCollected
integer
gemGrabGemsLost
integer
bountyStarsGained
integer
bountyStarsLost
integer
superUsedCount
```

```
integer
gadgetUsedCount
integer
bountyPickedMiddleStar
boolean
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
integer
damageDealt
integer
kills
integer
averageLatency
integer
damageReceived
integer
totalDamageToSafe
integer
totalDamageToPets
integer
siegeDamageToRobot
integer
maxKillStreak
integer
}
```

	tag string accountId string }] }
duration	integer
location	MatchLocation{ id integer name string gameMode string }
replayId	string
score	integer
isWinner	boolean
siege	SiegeStats{ botDamageToBase integer botLevelByRound { } }
players	PlayerEntryCompletedGameList[Play erEntryCompletedGame{ brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower

```
StarPower{
  name
  JsonLocalizedName{

  }
  id
  integer
  }
  gears
  GearInfoList[GearInfo{
    name
    JsonLocalizedName{

    }
    id
    integer
    level
    integer
  }]
  trophyChange
  integer
  trophies
  integer
  power
  integer
  name
  JsonLocalizedName{

  }
  id
  integer
  }
  statistics
  Stats{
    siegeBoltsCollected
    integer
    brawlBallGoalsScored
    integer
    gemGrabGemsCollected
    integer
    gemGrabGemsLost
    integer
```

bountyStarsGained  
integer  
bountyStarsLost  
integer  
superUsedCount  
integer  
gadgetUsedCount  
integer  
bountyPickedMiddleStar  
boolean  
matchEndKillStreak  
integer  
hotZoneInsideZonePercentage  
integer  
healingDoneToSelf  
integer  
healingDoneToTeamMates  
integer  
objectivesRecovered  
integer  
objectivesStolen  
integer  
brawlBallShotsOnGoal  
integer  
brawlBallShotsSaved  
integer  
healingDone  
integer  
deaths  
integer  
damageDealt  
integer  
kills  
integer  
averageLatency  
integer  
damageReceived  
integer  
totalDamageToSafe  
integer  
totalDamageToPets  
integer

	siegeDamageToRobot integer maxKillStreak integer } tag string accountId string }]
botDamageToBase	integer
botLevelByRound	{ } }
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } id integer }

	<pre> level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
statistics	<pre> Stats{ siegeBoltsCollected integer brawlBallGoalsScored integer gemGrabGemsCollected integer gemGrabGemsLost integer bountyStarsGained integer bountyStarsLost integer superUsedCount integer gadgetUsedCount integer bountyPickedMiddleStar boolean matchEndKillStreak integer hotZoneInsideZonePercentage integer healingDoneToSelf integer </pre>

	healingDoneToTeamMates integer objectivesRecovered integer objectivesStolen integer brawlBallShotsOnGoal integer brawlBallShotsSaved integer healingDone integer deaths integer damageDealt integer kills integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer }
tag	string
accountId	string

gadget	Accessory{ name JsonLocalizedName{ } } id integer }
starPower	StarPower{ name JsonLocalizedName{ } } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } ] id integer level integer ]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer

name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer
gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer
siegeDamageToRobot	integer

maxKillStreak	integer
id	integer
name	string
gameMode	string
}	
CompletedGameTeamList[CompletedGameTeamList{	
score	integer
isWinner	boolean
siege	SiegeStats{ botDamageToBase integer botLevelByRound { } } }
players	PlayerEntryCompletedGameList[Play erEntryCompletedGame{ brawler BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } id integer } gears GearInfoList[GearInfo{ }

```
name
JsonLocalizedName{

}

id
integer
level
integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}

id
integer
}
statistics
Stats{
siegeBoltsCollected
integer
brawlBallGoalsScored
integer
gemGrabGemsCollected
integer
gemGrabGemsLost
integer
bountyStarsGained
integer
bountyStarsLost
integer
superUsedCount
integer
gadgetUsedCount
integer
bountyPickedMiddleStar
boolean
```

```
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
integer
damageDealt
integer
kills
integer
averageLatency
integer
damageReceived
integer
totalDamageToSafe
integer
totalDamageToPets
integer
siegeDamageToRobot
integer
maxKillStreak
integer
}
tag
string
accountId
string
}]
```

botDamageToBase	integer
botLevelByRound	{ }
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } } ] id integer level integer } ] trophyChange integer trophies integer power integer name

	JsonLocalizedname{ } } id integer } }
statistics	Stats{ siegeBoltsCollected integer brawlBallGoalsScored integer gemGrabGemsCollected integer gemGrabGemsLost integer bountyStarsGained integer bountyStarsLost integer superUsedCount integer gadgetUsedCount integer bountyPickedMiddleStar boolean matchEndKillStreak integer hotZoneInsideZonePercentage integer healingDoneToSelf integer healingDoneToTeamMates integer objectivesRecovered integer objectivesStolen integer brawlBallShotsOnGoal integer brawlBallShotsSaved integer

	healingDone integer deaths integer damageDealt integer kills integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer }
tag	string
accountId	string
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }   id   integer   level   integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer

gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer
siegeDamageToRobot	integer
maxKillStreak	integer

}]

CompletedGameTeam{

score	integer
isWinner	boolean
siege	SiegeStats{ botDamageToBase integer botLevelByRound { } }
players	PlayerEntryCompletedGameList[PlayerEntryCompletedGame{ brawler BrawlerInfo{ gadget}}

```
Accessory{
  name
  JsonLocalizedName{

  }
  id
  integer
}
starPower
StarPower{
  name
  JsonLocalizedName{

  }
  id
  integer
}
gears
GearInfoList[GearInfo{
  name
  JsonLocalizedName{

  }
  id
  integer
  level
  integer
}]
trophyChange
integer
trophies
integer
power
integer
name
JsonLocalizedName{

}
id
integer
}
statistics
```

```
Stats{  
    siegeBoltsCollected  
    integer  
    brawlBallGoalsScored  
    integer  
    gemGrabGemsCollected  
    integer  
    gemGrabGemsLost  
    integer  
    bountyStarsGained  
    integer  
    bountyStarsLost  
    integer  
    superUsedCount  
    integer  
    gadgetUsedCount  
    integer  
    bountyPickedMiddleStar  
    boolean  
    matchEndKillStreak  
    integer  
    hotZoneInsideZonePercentage  
    integer  
    healingDoneToSelf  
    integer  
    healingDoneToTeamMates  
    integer  
    objectivesRecovered  
    integer  
    objectivesStolen  
    integer  
    brawlBallShotsOnGoal  
    integer  
    brawlBallShotsSaved  
    integer  
    healingDone  
    integer  
    deaths  
    integer  
    damageDealt  
    integer  
    kills
```

	integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer } tag string accountId string }]
botDamageToBase	integer
botLevelByRound	{ } }
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } id integer }

```
    }
    gears
    GearInfoList[GearInfo{
        name
        JsonLocalizedName{

    }
    id
    integer
    level
    integer
    }]
    trophyChange
    integer
    trophies
    integer
    power
    integer
    name
    JsonLocalizedName{

}
    id
    integer
}
```

#### statistics

```
Stats{
    siegeBoltsCollected
    integer
    brawlBallGoalsScored
    integer
    gemGrabGemsCollected
    integer
    gemGrabGemsLost
    integer
    bountyStarsGained
    integer
    bountyStarsLost
    integer
    superUsedCount
    integer
    gadgetUsedCount
```

```
integer
bountyPickedMiddleStar
boolean
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
integer
damageDealt
integer
kills
integer
averageLatency
integer
damageReceived
integer
totalDamageToSafe
integer
totalDamageToPets
integer
siegeDamageToRobot
integer
maxKillStreak
integer
}
```

tag	string
accountId	string
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }

id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer
gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer

siegeDamageToRobot	integer
maxKillStreak	integer
}	
PlayerEntryCompletedGameList[PlayerEntryCompletedGameList{	
brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{ } id integer } starPower StarPower{ name JsonLocalizedName{ } } id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{ } } id integer level integer }] trophyChange integer trophies integer power integer name

	JsonLocalizedname{ } } id integer } }
statistics	Stats{ siegeBoltsCollected integer brawlBallGoalsScored integer gemGrabGemsCollected integer gemGrabGemsLost integer bountyStarsGained integer bountyStarsLost integer superUsedCount integer gadgetUsedCount integer bountyPickedMiddleStar boolean matchEndKillStreak integer hotZoneInsideZonePercentage integer healingDoneToSelf integer healingDoneToTeamMates integer objectivesRecovered integer objectivesStolen integer brawlBallShotsOnGoal integer brawlBallShotsSaved integer

	healingDone integer deaths integer damageDealt integer kills integer averageLatency integer damageReceived integer totalDamageToSafe integer totalDamageToPets integer siegeDamageToRobot integer maxKillStreak integer }
tag	string
accountId	string
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }

gears	<pre> GearInfoList[GearInfo{   name   JsonLocalizedName{     }   id   integer   level   integer   }] </pre>
trophyChange	integer
trophies	integer
power	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
name	<pre> JsonLocalizedName{   } </pre>
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer

gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer
siegeDamageToRobot	integer
maxKillStreak	integer

}]

PlayerEntryCompletedGame{

brawler	BrawlerInfo{ gadget Accessory{ name JsonLocalizedName{  } id integer } starPower StarPower{ name JsonLocalizedName{  }
---------	---

	<pre> id integer } gears GearInfoList[GearInfo{ name JsonLocalizedName{  }  id integer level integer }] trophyChange integer trophies integer power integer name JsonLocalizedName{  }  id integer } </pre>
statistics	<pre> Stats{ siegeBoltsCollected integer brawlBallGoalsScored integer gemGrabGemsCollected integer gemGrabGemsLost integer bountyStarsGained integer bountyStarsLost integer superUsedCount </pre>

```
integer
gadgetUsedCount
integer
bountyPickedMiddleStar
boolean
matchEndKillStreak
integer
hotZoneInsideZonePercentage
integer
healingDoneToSelf
integer
healingDoneToTeamMates
integer
objectivesRecovered
integer
objectivesStolen
integer
brawlBallShotsOnGoal
integer
brawlBallShotsSaved
integer
healingDone
integer
deaths
integer
damageDealt
integer
kills
integer
averageLatency
integer
damageReceived
integer
totalDamageToSafe
integer
totalDamageToPets
integer
siegeDamageToRobot
integer
maxKillStreak
integer
}
```

tag	string
accountId	string
gadget	Accessory{ name JsonLocalizedName{ } id integer }
starPower	StarPower{ name JsonLocalizedName{ } id integer }
gears	GearInfoList[GearInfo{ name JsonLocalizedName{ } id integer level integer }]
trophyChange	integer
trophies	integer
power	integer
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }

id	integer
name	JsonLocalizedName{ }
id	integer
name	JsonLocalizedName{ }
id	integer
level	integer
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer
gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer

siegeDamageToRobot	integer
maxKillStreak	integer
}	
Stats{	
siegeBoltsCollected	integer
brawlBallGoalsScored	integer
gemGrabGemsCollected	integer
gemGrabGemsLost	integer
bountyStarsGained	integer
bountyStarsLost	integer
superUsedCount	integer
gadgetUsedCount	integer
bountyPickedMiddleStar	boolean
matchEndKillStreak	integer
hotZoneInsideZonePercentage	integer
healingDoneToSelf	integer
healingDoneToTeamMates	integer
objectivesRecovered	integer
objectivesStolen	integer
brawlBallShotsOnGoal	integer
brawlBallShotsSaved	integer
healingDone	integer
deaths	integer
damageDealt	integer
kills	integer
averageLatency	integer
damageReceived	integer
totalDamageToSafe	integer
totalDamageToPets	integer
siegeDamageToRobot	integer
maxKillStreak	integer
}	
SiegeStats{	
botDamageToBase	integer

botLevelByRound	{ }
}	
CancelMatchResponse{	
success	boolean
}	
BattleList[BattleList{	
battleTime	string
event	Event{ modelId integer mode stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ] id integer map JsonLocalizedName{ } } }
battle	BattleResult{ }

modelId	integer
mode	<p>stringEnum:</p> <p>[ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]</p>
id	integer
map	JsonLocalizedName{ }
battleTime	string

}]  

Battle{

event	<pre> Event{   modelId   integer   mode   stringEnum:     [ soloShowdown, duoShowdown,     heist, bounty, siege, gemGrab,     brawlBall, bigGame, bossFight,     roboRumble, takedown, loneStar,     presentPlunder, hotZone,     superCityRampage, knockout,     volleyBrawl, basketBrawl,     holdTheTrophy, trophyThieves,     duels, wipeout, payload, botDrop,     hunters, lastStand, snowtelThieves,     pumpkinPlunder, trophyEscape,     wipeout5V5, knockout5V5,     gemGrab5V5, brawlBall5V5,     godzillaCitySmash, paintBrawl,     trioShowdown, zombiePlunder,     jellyfishing, unknown ]   id   integer   map   JsonLocalizedName{    } } </pre>
battle	<pre> BattleResult{   } } </pre>
modelId	integer

mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
id	integer
map	JsonLocalizedName{ }
}	
BattleResult{	
}	
Event{	
modelId	integer

mode	stringEnum: [ soloShowdown, duoShowdown, heist, bounty, siege, gemGrab, brawlBall, bigGame, bossFight, roboRumble, takedown, loneStar, presentPlunder, hotZone, superCityRampage, knockout, volleyBrawl, basketBrawl, holdTheTrophy, trophyThieves, duels, wipeout, payload, botDrop, hunters, lastStand, snowtelThieves, pumpkinPlunder, trophyEscape, wipeout5V5, knockout5V5, gemGrab5V5, brawlBall5V5, godzillaCitySmash, paintBrawl, trioShowdown, zombiePlunder, jellyfishing, unknown ]
id	integer
map	JsonLocalizedName{ }

}

BrawlerList[BrawlerList{

gadgets	AccessoryList[Accessory{ name JsonLocalizedName{ } } id integer }]
name	JsonLocalizedName{ }
id	integer

starPowers	StarPowerList[StarPower{ name JsonLocalizedName{ } } id integer }]
name	JsonLocalizedName{ } }
id	integer
name	JsonLocalizedName{ } }
id	integer

}]

ClientError{

reason	string
message	string
type	string
detail	{ }

}