
Pyrez

Release 1.0.6.4

Jun 06, 2019

Contents:

1	Getting Started	3
1.1	Registration	3
1.2	Credentials	3
1.3	Importing	4
1.4	Creating API object	4
1.5	Sessions	4
2	API Reference (pyrez.api)	5
2.1	PaladinsAPI	5
2.2	RealmRoyaleAPI	15
2.3	SmiteAPI	21
2.4	StatusPageAPI	30
3	Enums (pyrez.enumerations)	33
4	Exceptions (pyrez.exceptions)	45
5	Data classes (pyrez.models)	47
6	If you still can't find what you're looking for, try in one of the following pages:	51
	Python Module Index	53
	Index	55

Pyrez is an open-source wrapper for [Hi-Rez Studios](#) API that supports [Paladins](#), [Realm Royale](#) and [Smite](#), written in [Python](#).

CHAPTER 1

Getting Started

1.1 Registration

A *Credentials* that will provide access to [Hi-Rez Studios API](#).

If you don't already have a Credentials, [click here](#) to become developer.

Attention: If your application is accepted, you will receive an e-mail from Hi-Rez Studios containing your personal *Credentials* within a few days.

1.2 Credentials

To access the API you'll need your own set of Credentials which consist of a Developer ID (devId) and an Authentication Key (authKey).

Here are the Credentials for a sample account:

devId	authKey
1004	23DF3C7E9BD14D84BF892AD206B6755C

Note: The same devId and authKey combination should work for [Paladins API](#), [Smite API](#) and [Realm Royale API](#), across all supported platforms.

Do not request a new, if you already have a Credentials.

1.3 Importing

```
import pyrez
import pyrez.api
from pyrez.api import PaladinsAPI, SmiteAPI, RealmRoyaleAPI
import pyrez.enumerations
import pyrez.models
```

1.4 Creating API object

```
paladins = PaladinsAPI(options)

#or
smite = SmiteAPI(options)

#or
reamlRoyale = RealmRoyaleAPI(options)
```

Options can have the following fields:

- `devId (int)` – This is the Developer ID that you receive from Hi-Rez Studios.
- `authKey (str)` – This is the Authentication Key that you receive from Hi-Rez Studios.
- `responseFormat (Format)` – The response format that will be used by default when making requests.
- `sessionId (str)` – The response format that will be used by default when making requests.
- `storeSession (str)` – Allows Pyrez to read and store `sessionId` in a .json file.

1.5 Sessions

Sessions are created automatically and self-managed by Pyrez so you really don't need to initialise / call this method directly. However, you can set it manually or even request a new Session.

Manually:

```
paladins = PaladinsAPI(devId=1004,
                        authKey="23DF3C7E9BD14D84BF892AD206B6755C",
                        sessionId="1465AFCA32DBDB800CEF8C72F296C52C")
```

Requesting a new Session:

```
paladins = PaladinsAPI(devId=1004, authKey="23DF3C7E9BD14D84BF892AD206B6755C")
session = paladins._createSession()
print(session.sessionId)

>>> '1465AFCA32DBDB800CEF8C72F296C52C'
```

API Reference (pyrez.api)

The following section outlines the API of Pyrez.

2.1 PaladinsAPI

These methods are supported by PaladinsAPI object.

```
class pyrez.api.PaladinsAPI(devId, authKey, responseFormat=Format.JSON, sessionId=None,  
                           storeSession=True)
```

Represents a client that connects to [Paladins](#) API.

Note: Any player with `Privacy Mode` enabled in-game will return a null dataset from methods that require a `playerId` or `playerName`.

Keyword Arguments

- **devId** (`int`) – Used for authentication. This is the Developer ID that you receive from Hi-Rez Studios.
- **authKey** (`str`) – Used for authentication. This is the Authentication Key that you receive from Hi-Rez Studios.
- **responseFormat** (Optional `Format`) – The response format that will be used by default when making requests. Passing in `None` or an invalid value will use the default instead of the passed in value.
- **sessionId** (Optional `str`) – Manually sets an active sessionId. Passing in `None` or an invalid sessionId will use the default instead of the passed in value.
- **storeSession** (Optional `bool`) – Allows Pyrez to read and store sessionId in a .json file. Defaults to `False`.

Raises

- `pyrez.exceptions.IdOrAuthEmpty` – Raised when the Developer ID or Authentication Key is not specified.
- `pyrez.exceptions.InvalidArgument` – Raised when an invalid Credentials is passed.

authKey

`str` – This is the Authentication Key that you receive from Hi-Rez Studios.

devId

`int` – This is the Developer ID that you receive from Hi-Rez Studios.

onSessionCreated

`pyrez.events.Event` – A decorator that registers an event to listen to.

responseFormat

`Format` – The response format that will be used by default when making requests.

sessionId

`str` – The active sessionId.

statusPage

`StatusPageAPI` – An object that represents `StatusPageAPI` client.

storeSession

`bool` – Allows Pyrez to read and store sessionId in a .json file.

getChampionCards (*godId*, *language=Language.English*)

Returns all Champion cards.

Parameters

- **godId** (`int` or `Champions`) – The god ID to get their cards.
- **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a list of `ChampionCard` objects or `None`

Return type list of `pyrez.models.Paladins.ChampionCard`

getChampionLeaderboard (*godId*, *queueId=QueuePaladins.Live_Competitive_Keyboard*)

Returns the current season's leaderboard for a champion/queue combination.

Parameters

- **godId** (`int` or `Champions`) – The god ID.
- **queueId** (Optional `int` or `QueuePaladins`) – The id of the game mode. Passing in `None` will use `pyrez.enumerations.QueuePaladins.Live_Competitive_Keyboard` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a list of `pyrez.models.Smite.GodLeaderboard` objects or None

Return type list of `pyrez.models.Smite.GodLeaderboard`

getChampionRanks (*playerId*)

Returns the Rank and Worshippers value for each Champion a player has played.

Parameters `playerId` (*int*) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getChampionSkins (*godId*, *language=Language.English*)

Returns all available skins for a particular Champion.

Parameters

- `godId` (*int*) –

- `language` (Optional *int* or *Language*) – The language that you want results returned in. Passing in None will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getChampions (*language=Language.English*)

Returns all Champions and their various attributes.

Parameters `language` (Optional *int* or *Language*) – The language that you want results returned in. Passing in None will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getDataUsed ()

Returns API Developer daily usage limits and the current status against those limits.

Note: Getting your data usage does contribute to your daily API limits.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `pyrez.models.DataUsed` object containing resources used or None.

Return type `pyrez.models.DataUsed` or None

getDemoDetails (*matchId*)

Returns information regarding a particular match.

Note: Rarely used in lieu of *getMatch()*.

Parameters *matchId* (*int*) – The id of the match. Can be obtained from *getMatchHistory()*, *getTopMatches()* & *getMatchIds()*.

Raises *TypeError* – Raised when more (or less) than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

getEsportsProLeague ()

Returns the matchup information for each matchup for the current eSports Pro League season.

Raises *TypeError* – Raised when passing any parameters.

Note: This method raises *makeRequest()* exceptions.

getFriends (*playerId*)

Returns the User names of each of the player's friends of one player.

Parameters *playerId* (*int*) –

Note: This method is PC only.

Raises *TypeError* – Raised when more (or less) than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

Returns

Return type List of *pyrez.models.Friend* objects

getGodLeaderboard (*godId*, *queueId*)

Returns the current season's leaderboard for a god/queue combination.

Parameters

- *godId* (*int*) –
- *queueId* (*int*) – The id of the game mode

Raises *TypeError* – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

getGodRanks (*playerId*)

Returns the Rank and Worshippers value for each God a player has played.

Parameters `playerId (int)` –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type List of `pyrez.models.GodRank` objects

getGodSkins (*godId, language=Language.English*)

Returns all available skins for a particular God.

Parameters

- **godId** (`int`) –
- **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getGods (*language=Language.English*)

Returns all Gods and their various attributes.

Parameters **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type Returns a list of `pyrez.models.Paladins.Champion` objects

getItems (*language=Language.English*)

Returns all Items and their various attributes.

Parameters **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getLatestPatchNotes (*language=Language.English*)

Parameters **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

getLeagueLeaderboard (*queueId*, *tier*, *split*)

Returns the top players for a particular league (as indicated by the queue/tier/split parameters).

Parameters

- **queueId** (*int*) – The id of the game mode
- **tier** (*int*) –
- **split** (*int*) –

Raises `TypeError` – Raised when more than 3 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getLeagueSeasons (*queueId*)

Provides a list of seasons (including the single active season) for a match queue.

Parameters **queueId** (*int*) – The id of the game mode

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getMatch (*matchId*, *isLiveMatch=False*)

Returns the player information / statistics for a particular match.

There is three ways to call this method:

```
getMatch(matchId)
#or
getMatch([matchId, matchId, matchId])
#or
getMatch(matchId, True)
```

Parameters

- **matchId** (*int* or *list* of *int*) – The id of the match. Can be obtained from `getMatchHistory()`, `getTopMatches()` & `getMatchIds()`.
- **isLiveMatch** (Optional *bool*) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Warning: There is a byte limit to the amount of data returned.

Please limit the matchId parameter to 5-10 matches for DB Performance reasons.

getMatchHistory (*playerId*)

Gets recent matches and high level match statistics for a particular player.

Parameters **playerId** (*int*) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getMatchIds (*queueId*, *date=None*, *hour=-1*)

Lists all Match IDs for a particular Match Queue.

Useful for API developers interested in constructing data by Queue.

Parameters

- **queueId** (*int*) – The id of the game mode
- **date** (*int*) –
- **hour** (*int*) – Used to limit the data returned (valid values: 0 - 23).

An *hour* parameter of *-1* represents the entire day, but be warned that this may be more data than we can return for certain queues.

Raises `TypeError` – Raised when more than 3 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Warning: To avoid HTTP timeouts in the `getMatchIds()` method, you can now specify a 10-minute window within the specified `{hour}` field to lessen the size of data returned by appending a “,mm” value to the end of `{hour}`.

For example, to get the match Ids for the first 10 minutes of hour 3, you would specify `{hour}` as “3,00”.

This would only return the Ids between the time 3:00 to 3:09. Rules below:

Only valid values for mm are “00”, “10”, “20”, “30”, “40”, “50”.

To get the entire third hour worth of Match Ids, call `getMatchIds()` 6 times, specifying the following values for `{hour}`: “3,00”, “3,10”, “3,20”, “3,30”, “3,40”, “3,50”.

getPatchInfo ()

Function returns information about current deployed patch.

Note: Currently, this information only includes patch version.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type Object of `pyrez.models.PatchInfo`

getPlayer (*player*, *portalId=None*)

Returns league and other high level data for a particular player.

Parameters

- **player** (*str* or *int*) – playerName or playerId of the player you want to get info on
- **portalId** (Optional *int* or *pyrez.enumerations.PortalId*) – The portalId that you want to looking for (Defaults to None)

Raises

- *pyrez.exceptions.PlayerNotFound* – Raised if the given player does not exist or it's hidden.
- *TypeError* – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

Returns *list* of *pyrez.models.Paladins.Player* objects with league and other high level data for a particular player.

Return type *list* of *pyrez.models.Paladins.Player*

getPlayerAchievements (*playerId*)

Returns select achievement totals for the specified playerId.

Parameters **playerId** (*int*) –

Raises *TypeError* – Raised when more (or less) than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

getPlayerId (*playerName*, *portalId=None*, *xboxOrSwitch=False*)

Function returns a list of Hi-Rez playerId values.

Parameters

- **playerName** (*str* or *int*) –
- **portalId** (Optional *int* or *pyrez.enumerations.PortalId*) – Only returns a list of Hi-Rez playerId values for portalId provided. (Defaults to None)
- **xboxOrSwitch** (*bool*) – Meaningful only for the Paladins Xbox and Switch API.

Therefore a Paladins Gamer Tag value could be the same as a Paladins Switch Gamer Tag value.

Additionally, there could be multiple identical Paladins Switch Gamer Tag values. The purpose of this parameter is to return all Player ID data associated with the playerName (gamer tag) parameter. The expectation is that the unique player_id returned could then be used in subsequent method calls.

Raises *TypeError* – Raised when more than 3 parameters or less than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

getPlayerLoadouts (*playerId*, *language=Language.English*)

Returns deck loadouts per Champion.

Parameters

- **playerId** (*int*) –

- **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getPlayerStatus (*playerId*)

Returns player status as follows:

- 0: Offline,
- 1: In Lobby,
- 2: God Selection,
- 3: In Game,
- 4: Online,
- 5: Player not found

Parameters `playerId` (`int`) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns Object of `pyrez.models.PlayerStatus` containing player status

Return type `pyrez.models.PlayerStatus`

getQueueStats (*playerId*, *queueId*)

Returns match summary statistics for a (player, queue) combination grouped by gods played.

Parameters

- `playerId` (`int`) –
- `queueId` (`int`) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getServerStatus ()

Function returns UP/DOWN status for the primary game/platform environments.

Note: Data is cached once a minute.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Object of `pyrez.models.HiRezServerStatus`

Return type `pyrez.models.HiRezServerStatus`

getWebsitePost (*language=Language.English, slug=None, query=None*)

Parameters **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 3 parameters or less than 1 parameter is passed.

makeRequest (*apiMethod=None, params=()*)

Parameters

- **apiMethod** (`str`) –
- **params** (Optional: `list` or `tuple`) –

Raises

- `pyrez.exceptions.DailyLimit` – Raised when the daily request limit is reached.
- `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.
- `pyrez.exceptions.WrongCredentials` – Raised when a wrong Credentials is passed.
- `pyrez.exceptions.RequestError` – Raised when the server encountered an error processing the request.
- `pyrez.exceptions.NotFound` – Raised when the requested endpoint is not found.
- `pyrez.exceptions.SessionLimit` – Raised when the maximum number of active sessions is reached.

ping ()

A quick way of validating access (establish connectivity) to the Hi-Rez API.

You do not need to authenticate your ID or key to do this.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `pyrez.models.Ping` objects containing infos about the API.

Return type `pyrez.models.Ping`

searchPlayers (*playerName*)

Parameters **playerName** (`str`) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

testSession (*sessionId=None*)

A means of validating that a session is established.

Parameters **sessionId** (Optional *str*) – A sessionId to validate. Passing in *None* will use *sessionId* instead of the passed in value.

Raises *TypeError* – Raised when more (or less) than 1 parameter is passed.

Note: This method raises *makeRequest ()* exceptions.

Returns Returns a *bool* that means if the passed sessionId is valid.

Return type *bool*

2.2 RealmRoyaleAPI

These methods are supported by RealmRoyaleAPI object.

class `pyrez.api.RealmRoyaleAPI` (*devId*, *authKey*, *responseFormat=Format.JSON*, *sessionId=None*, *storeSession=True*)

Represents a client that connects to [Realm Royale](#) API.

Note: Any player with *Privacy Mode* enabled in-game will return a null dataset from methods that require a *playerId* or *playerName*.

Keyword Arguments

- **devId** (*int*) – Used for authentication. This is the Developer ID that you receive from Hi-Rez Studios.
- **authKey** (*str*) – Used for authentication. This is the Authentication Key that you receive from Hi-Rez Studios.
- **responseFormat** (Optional *Format*) – The response format that will be used by default when making requests. Passing in *None* or an invalid value will use the default instead of the passed in value.
- **sessionId** (Optional *str*) – Manually sets an active sessionId. Passing in *None* or an invalid sessionId will use the default instead of the passed in value.
- **storeSession** (Optional *bool*) – Allows Pyrez to read and store sessionId in a .json file. Defaults to *False*.

Raises

- *pyrez.exceptions.IdOrAuthEmpty* – Raised when the Developer ID or Authentication Key is not specified.
- *pyrez.exceptions.InvalidArgument* – Raised when an invalid Credentials is passed.

authKey

str – This is the Authentication Key that you receive from Hi-Rez Studios.

devId

int – This is the Developer ID that you receive from Hi-Rez Studios.

onSessionCreated

`pyrez.events.Event` – A decorator that registers an event to listen to.

responseFormat

`Format` – The response format that will be used by default when making requests.

sessionId

`str` – The active sessionId.

statusPage

`StatusPageAPI` – An object that represents `StatusPageAPI` client.

storeSession

`bool` – Allows Pyrez to read and store sessionId in a .json file.

getDataUsed()

Returns API Developer daily usage limits and the current status against those limits.

Note: Getting your data usage does contribute to your daily API limits.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `pyrez.models.DataUsed` object containing resources used or None.

Return type `pyrez.models.DataUsed` or None

getFriends(playerId)

Returns the User names of each of the player's friends of one player.

Parameters `playerId(int)` –

Note: This method is PC only.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns

Return type List of `pyrez.models.Friend` objects

getItems(language=Language.English)

Get all talents

Parameters `language` (Optional `int` or `Language`) – The language that you want results returned in. Passing in None will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getLeaderboard (*queueId*, *rankingCriteria*)

Parameters **rankingCriteria** (*int*) – Can be: - 1: team_wins, - 2: team_average_placement (shown below), - 3: individual_average_kills, - 4. win_rate, possibly/probably others as desired

Note:

- for duo and quad queues/modes the individual's placement results reflect their team/grouping; solo is self-explanatory
 - will limit results to the top 500 players (minimum 50 matches played per queue); we never like to expose weak/beginner players
 - players that select to be “private” will have their player_name and player_id values hidden
-

Warning: Expect this data to be cached on an hourly basis because the query to acquire the data will be expensive; don't spam the calls

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getMatch (*matchId*, *isLiveMatch=False*)

Returns the player information / statistics for a particular match.

There is three ways to call this method:

```
getMatch(matchId)
#or
getMatch([matchId, matchId, matchId])
#or
getMatch(matchId, True)
```

Parameters

- **matchId** (*int* or *list of int*) – The id of the match. Can be obtained from `getMatchHistory()`, `getTopMatches()` & `getMatchIds()`.
- **isLiveMatch** (Optional *bool*) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Warning: There is a byte limit to the amount of data returned.

Please limit the `matchId` parameter to 5-10 matches for DB Performance reasons.

getMatchHistory (*playerId*, *startDatetime=None*)

Gets recent matches and high level match statistics for a particular player.

Parameters `playerId` (*int*) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getMatchIds (*queueId*, *date=None*, *hour=-1*)

Lists all Match IDs for a particular Match Queue.

Useful for API developers interested in constructing data by Queue.

Parameters

- `queueId` (*int*) – The id of the game mode
- `date` (*int*) –
- `hour` (*int*) – Used to limit the data returned (valid values: 0 - 23).

An `hour` parameter of `-1` represents the entire day, but be warned that this may be more data than we can return for certain queues.

Raises `TypeError` – Raised when more than 3 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Warning: To avoid HTTP timeouts in the `getMatchIds()` method, you can now specify a 10-minute window within the specified `{hour}` field to lessen the size of data returned by appending a “,mm” value to the end of `{hour}`.

For example, to get the match Ids for the first 10 minutes of hour 3, you would specify `{hour}` as “3,00”.

This would only return the Ids between the time 3:00 to 3:09. Rules below:

Only valid values for mm are “00”, “10”, “20”, “30”, “40”, “50”.

To get the entire third hour worth of Match Ids, call `getMatchIds()` 6 times, specifying the following values for `{hour}`: “3,00”, “3,10”, “3,20”, “3,30”, “3,40”, “3,50”.

getPatchInfo ()

Function returns information about current deployed patch.

Note: Currently, this information only includes patch version.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type Object of `pyrez.models.PatchInfo`

getPlayer (*player*, *platform=None*)

Returns league and other high level data for a particular player.

Parameters **player** (`int` or `str`) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getPlayerAchievements (*playerId*)

Returns select achievement totals for the specified playerId.

Parameters **playerId** (`int`) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getPlayerId (*playerName*, *portalId=None*)

Function returns a list of Hi-Rez playerId values.

Parameters

- **playerName** (`int` or `str`) – Function returns a list of Hi-Rez playerId values (expected list size = 1) for playerName provided.
- **portalId** (Optional `int` or `pyrez.enumerations.PortalId`) – Only returns a list of Hi-Rez playerId values for portalId provided. (Defaults to None)

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getPlayerStats (*playerId*)

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getPlayerStatus (*playerId*)

Returns player status as follows:

- 0: Offline,
- 1: In Lobby,
- 2: God Selection,

- 3: In Game,
- 4: Online,
- 5: Player not found

Parameters `playerId(int)` –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns Object of `pyrez.models.PlayerStatus` containing player status

Return type `pyrez.models.PlayerStatus`

getQueueStats (`playerId, queueId`)

Returns match summary statistics for a (player, queue) combination grouped by gods played.

Parameters

- `playerId(int)` –
- `queueId(int)` –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getServerStatus ()

Function returns UP/DOWN status for the primary game/platform environments.

Note: Data is cached once a minute.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Object of `pyrez.models.HiRezServerStatus`

Return type `pyrez.models.HiRezServerStatus`

makeRequest (`apiMethod=None, params=()`)

Parameters

- `apiMethod(str)` –
- `params` (Optional: `list` or `tuple`) –

Raises

- `pyrez.exceptions.DailyLimit` – Raised when the daily request limit is reached.
- `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

- `pyrez.exceptions.WrongCredentials` – Raised when a wrong Credentials is passed.
- `pyrez.exceptions.RequestError` – Raised when the server encountered an error processing the request.
- `pyrez.exceptions.NotFound` – Raised when the requested endpoint is not found.
- `pyrez.exceptions.SessionLimit` – Raised when the maximum number of active sessions is reached.

ping()

A quick way of validating access (establish connectivity) to the Hi-Rez API.

You do not need to authenticate your ID or key to do this.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `pyrez.models.Ping` objects containing infos about the API.

Return type `pyrez.models.Ping`

searchPlayers(playerName)

Parameters `playerName (str)` –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

testSession(sessionId=None)

A means of validating that a session is established.

Parameters `sessionId` (Optional `str`) – A sessionId to validate. Passing in `None` will use `sessionId` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `bool` that means if the passed sessionId is valid.

Return type `bool`

2.3 SmiteAPI

These methods are supported by SmiteAPI object.

class `pyrez.api.SmiteAPI` (`devId`, `authKey`, `responseFormat=Format.JSON`, `sessionId=None`, `storeSession=True`)

Represents a client that connects to Smite API.

Note: Any player with `Privacy Mode` enabled in-game will return a null dataset from methods that require a `playerId` or `playerName`.

Keyword Arguments

- **devId** (`int`) – Used for authentication. This is the Developer ID that you receive from Hi-Rez Studios.
- **authKey** (`str`) – Used for authentication. This is the Authentication Key that you receive from Hi-Rez Studios.
- **responseFormat** (Optional `Format`) – The response format that will be used by default when making requests. Passing in `None` or an invalid value will use the default instead of the passed in value.
- **sessionId** (Optional `str`) – Manually sets an active sessionId. Passing in `None` or an invalid sessionId will use the default instead of the passed in value.
- **storeSession** (Optional `bool`) – Allows Pyrez to read and store sessionId in a .json file. Defaults to `False`.

Raises

- `pyrez.exceptions.IdOrAuthEmpty` – Raised when the Developer ID or Authentication Key is not specified.
- `pyrez.exceptions.InvalidArgument` – Raised when an invalid Credentials is passed.

authKey

`str` – This is the Authentication Key that you receive from Hi-Rez Studios.

devId

`int` – This is the Developer ID that you receive from Hi-Rez Studios.

onSessionCreated

`pyrez.events.Event` – A decorator that registers an event to listen to.

responseFormat

`Format` – The response format that will be used by default when making requests.

sessionId

`str` – The active sessionId.

statusPage

`StatusPageAPI` – An object that represents `StatusPageAPI` client.

storeSession

`bool` – Allows Pyrez to read and store sessionId in a .json file.

getDataUsed()

Returns API Developer daily usage limits and the current status against those limits.

Note: Getting your data usage does contribute to your daily API limits.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises *makeRequest()* exceptions.

Returns Returns a *pyrez.models.DataUsed* object containing resources used or None.

Return type *pyrez.models.DataUsed* or None

getDemoDetails (*matchId*)

Returns information regarding a particular match.

Note: Rarely used in lieu of *getMatch()*.

Parameters *matchId* (*int*) – The id of the match. Can be obtained from *getMatchHistory()*, *getTopMatches()* & *getMatchIds()*.

Raises *TypeError* – Raised when more (or less) than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

getEsportsProLeague ()

Returns the matchup information for each matchup for the current eSports Pro League season.

Raises *TypeError* – Raised when passing any parameters.

Note: This method raises *makeRequest()* exceptions.

getFriends (*playerId*)

Returns the User names of each of the player's friends of one player.

Parameters *playerId* (*int*) –

Note: This method is PC only.

Raises *TypeError* – Raised when more (or less) than 1 parameter is passed.

Note: This method raises *makeRequest()* exceptions.

Returns

Return type List of *pyrez.models.Friend* objects

getGodLeaderboard (*godId*, *queueId*)

Returns the current season's leaderboard for a god/queue combination.

Parameters

- *godId* (*int*) –
- *queueId* (*int*) – The id of the game mode

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getGodRanks (*playerId*)

Returns the Rank and Worshippers value for each God a player has played.

Parameters `playerId` (*int*) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type List of `pyrez.models.GodRank` objects

getGodRecommendedItems (*godId*, *language=Language.English*)

Returns the Recommended Items for a particular God.

Parameters

- `godId` (*int*) –
- `language` (Optional *int* or *Language*) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getGodSkins (*godId*, *language=Language.English*)

Returns all available skins for a particular God.

Parameters

- `godId` (*int*) –
- `language` (Optional *int* or *Language*) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getGods (*language=Language.English*)

Returns all Gods and their various attributes.

Parameters `language` (Optional *int* or *Language*) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Returns List of pyrez.models.God or pyrez.models.Champion objects

getItems (*language=Language.English*)

Returns all Items and their various attributes.

Parameters **language** (Optional `int` or `Language`) – The language that you want results returned in. Passing in `None` will use `Language.English` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getLeagueLeaderboard (*queueId, tier, split*)

Returns the top players for a particular league (as indicated by the queue/tier/split parameters).

Parameters

- **queueId** (`int`) – The id of the game mode
- **tier** (`int`) –
- **split** (`int`) –

Raises `TypeError` – Raised when more than 3 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getLeagueSeasons (*queueId*)

Provides a list of seasons (including the single active season) for a match queue.

Parameters **queueId** (`int`) – The id of the game mode

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

getMatch (*matchId, isLiveMatch=False*)

Returns the player information / statistics for a particular match.

There is three ways to call this method:

```
getMatch(matchId)
#or
getMatch([matchId, matchId, matchId])
#or
getMatch(matchId, True)
```

Parameters

- **matchId** (`int` or `list of int`) – The id of the match. Can be obtained from `getMatchHistory()`, `getTopMatches()` & `getMatchIds()`.
- **isLiveMatch** (Optional `bool`) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Warning: There is a byte limit to the amount of data returned.

Please limit the `matchId` parameter to 5-10 matches for DB Performance reasons.

getMatchHistory (*playerId*)

Gets recent matches and high level match statistics for a particular player.

Parameters `playerId` (*int*) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getMatchIds (*queueId*, *date=None*, *hour=-1*)

Lists all Match IDs for a particular Match Queue.

Useful for API developers interested in constructing data by Queue.

Parameters

- **queueId** (*int*) – The id of the game mode
- **date** (*int*) –
- **hour** (*int*) – Used to limit the data returned (valid values: 0 - 23).

An `hour` parameter of `-1` represents the entire day, but be warned that this may be more data than we can return for certain queues.

Raises `TypeError` – Raised when more than 3 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Warning: To avoid HTTP timeouts in the `getMatchIds()` method, you can now specify a 10-minute window within the specified `{hour}` field to lessen the size of data returned by appending a “,mm” value to the end of `{hour}`.

For example, to get the match Ids for the first 10 minutes of hour 3, you would specify `{hour}` as “3,00”.

This would only return the Ids between the time 3:00 to 3:09. Rules below:

Only valid values for mm are “00”, “10”, “20”, “30”, “40”, “50”.

To get the entire third hour worth of Match Ids, call `getMatchIds()` 6 times, specifying the following values for `{hour}`: “3,00”, “3,10”, “3,20”, “3,30”, “3,40”, “3,50”.

getMotd ()

Returns information about the 20 most recent Match-of-the-Days.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest ()` exceptions.

getPatchInfo ()

Function returns information about current deployed patch.

Note: Currently, this information only includes patch version.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type Object of `pyrez.models.PatchInfo`

getPlayer (player, portalId=None)

Returns league and other high level data for a particular player.

Parameters

- **player** (`int` or `str`) – playerName or playerId of the player you want to get info on
- **portalId** (Optional `int` or `pyrez.enumerations.PortalId`) – The portalId that you want to looking for (Defaults to None)

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Returns

Return type `pyrez.models.PlayerSmite` | `pyrez.models.PlayerPaladins` object with league and other high level data for a particular player.

getPlayerAchievements (playerId)

Returns select achievement totals for the specified playerId.

Parameters **playerId** (`int`) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getPlayerId (playerName, portalId=None)

Function returns a list of Hi-Rez playerId values.

Parameters

- **playerName** (`int` or `str`) – Function returns a list of Hi-Rez playerId values (expected list size = 1) for playerName provided.

- **portalId** (Optional `int` or `pyrez.enumerations.PortalId`) – Only returns a list of Hi-Rez playerId values for portalId provided. (Defaults to None)

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getPlayerStatus (*playerId*)

Returns player status as follows:

- 0: Offline,
- 1: In Lobby,
- 2: God Selection,
- 3: In Game,
- 4: Online,
- 5: Player not found

Parameters **playerId** (`int`) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

Returns Object of `pyrez.models.PlayerStatus` containing player status

Return type `pyrez.models.PlayerStatus`

getQueueStats (*playerId, queueId*)

Returns match summary statistics for a (player, queue) combination grouped by gods played.

Parameters

- **playerId** (`int`) –
- **queueId** (`int`) –

Raises `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getServerStatus ()

Function returns UP/DOWN status for the primary game/platform environments.

Note: Data is cached once a minute.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest ()` exceptions.

Returns Object of `pyrez.models.HiRezServerStatus`

Return type `pyrez.models.HiRezServerStatus`

getTeamDetails (*clanId*)

Lists the number of players and other high level details for a particular clan.

Parameters `clanId` (*int*) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getTeamPlayers (*clanId*)

Lists the players for a particular clan.

Parameters `clanId` (*int*) –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest ()` exceptions.

getTopMatches ()

Lists the 50 most watched / most recent recorded matches.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest ()` exceptions.

makeRequest (*apiMethod=None, params=()*)

Parameters

- **apiMethod** (*str*) –
- **params** (Optional: *list* or *tuple*) –

Raises

- `pyrez.exceptions.DailyLimit` – Raised when the daily request limit is reached.
- `TypeError` – Raised when more than 2 parameters or less than 1 parameter is passed.
- `pyrez.exceptions.WrongCredentials` – Raised when a wrong Credentials is passed.
- `pyrez.exceptions.RequestError` – Raised when the server encountered an error processing the request.
- `pyrez.exceptions.NotFound` – Raised when the requested endpoint is not found.
- `pyrez.exceptions.SessionLimit` – Raised when the maximum number of active sessions is reached.

ping()

A quick way of validating access (establish connectivity) to the Hi-Rez API.

You do not need to authenticate your ID or key to do this.

Raises `TypeError` – Raised when passing any parameters.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `pyrez.models.Ping` objects containing infos about the API.

Return type `pyrez.models.Ping`

searchPlayers(playerName)

Parameters `playerName(str)` –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

searchTeams(searchTeam)

Returns high level information for Clan names containing the `searchTeam` string.

Parameters `searchTeam(str)` –

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

testSession(sessionId=None)

A means of validating that a session is established.

Parameters `sessionId` (Optional `str`) – A `sessionId` to validate. Passing in `None` will use `sessionId` instead of the passed in value.

Raises `TypeError` – Raised when more (or less) than 1 parameter is passed.

Note: This method raises `makeRequest()` exceptions.

Returns Returns a `bool` that means if the passed `sessionId` is valid.

Return type `bool`

2.4 StatusPageAPI

These methods are supported by `StatusPageAPI` object.

class pyrez.api.StatusPageAPI

A wrapper for the [Status Page](#) API, exposing convenient actions useful for embedding your status anywhere.

getComponents ()

Get the components for the [Status Page](#).

Each component is listed along with its status - one of operational, degraded_performance, partial_outage, or major_outage.

getHistory (_format=Format.JSON)

Get the history for the [Status Page](#).

Parameters **_format** (Optional *Format*) – Passing in None will use the default instead of the passed in value.

getIncidents (unresolvedOnly=False)

Get a list of the 50 most recent incidents. This includes all unresolved incidents (Investigating, Identified, Monitoring, Resolved, or Postmortem).

Parameters **unresolvedOnly** (Optional *bool*) – Only returns a list of unresolved incidents state (Investigating, Identified, or Monitoring).

getScheduledMaintenances (activeOnly=False, upcomingOnly=False)

Get a list of the 50 most recent scheduled maintenances. This includes all scheduled maintenances (Scheduled, In Progress, Verifying, or Completed).

Parameters

- **activeOnly** (Optional *bool*) – Only returns a list of active maintenances. (In Progress or Verifying state)
- **upcomingOnly** (Optional *bool*) – Only returns a list of upcoming maintenances. (scheduled maintenances still in the Scheduled state)

getStatus ()

Get the status rollup for the whole [Status Page](#).

This endpoint includes an indicator - one of none, minor, major, or critical, as well as a human description of the blended component status.

Examples of the blended status include All Systems Operational, Partial System Outage, and Major Service Outage.

getSummary ()

Get a summary of the [Status Page](#), including a status indicator, component statuses, unresolved incidents, and any upcoming or in-progress scheduled maintenances.

Enums (pyrez.enumerations)

There are several enums used within Pyrez to make it easier for you to configure the library for use how you like.

class `pyrez.enumerations.Enum`

Represents a generic enum object. This is a sub-class of `enum.Enum`.

Supported Operations:

Operation	Description
<code>x == y</code>	Checks if two Enum are equal.
<code>x != y</code>	Checks if two Enum are not equal.
<code>hash(x)</code>	Return the Enum's hash.
<code>str(x)</code>	Returns the Enum's name with discriminator.
<code>int(x)</code>	Return the Enum's value as int.

equal (*other*)

getId ()

getName ()

class `pyrez.enumerations.Champions`

Represents a Paladins Champion. This is a sub-class of `Enum`.

Supported Operations:

Operation	Description
<code>x == y</code>	Checks if two Champions are equal.
<code>x != y</code>	Checks if two Champions are not equal.
<code>hash(x)</code>	Return the Champion's hash.
<code>str(x)</code>	Returns the Champion's name with discriminator.
<code>int(x)</code>	Return the Champion's value as int.

getHeader ()

```
getIcon()  
isDamage()  
isFlank()  
isFrontline()  
isSupport()  
Androxus = 2205  
Ash = 2404  
Atlas = 2512  
Barik = 2073  
Bomb_King = 2281  
Buck = 2147  
Cassie = 2092  
Dredge = 2495  
Drogoz = 2277  
Evie = 2094  
Fernando = 2071  
Furia = 2491  
Grohk = 2093  
Grover = 2254  
Imani = 2509  
Inara = 2348  
Jenos = 2431  
Khan = 2479  
Kinessa = 2249  
Koga = 2493  
Lex = 2362  
Lian = 2417  
Maeve = 2338  
Makoa = 2288  
Mal_Damba = 2303  
Moji = 2481  
Pip = 2056  
Ruckus = 2149  
Seris = 2372  
Sha_Lin = 2307  
Skye = 2057
```

```

Strix = 2438
Talus = 2472
Terminus = 2477
Torvald = 2322
Tyra = 2314
Viktor = 2285
Vivian = 2480
Willo = 2393
Ying = 2267
Zhin = 2420

```

```
class pyrez.enumerations.Classes.Classes
```

```
    An enumeration.
```

```

Assassin = 2496
Engineer = 2495
Hunter = 2493
Mage = 2494
Warrior = 2285

```

```
class pyrez.enumerations.Endpoint
```

```
    The endpoint that you want to access to retrieve information from the Hi-Rez Studios API.
```

```
    getEndpoint (_endpoint=None)
```

```
    HAND_OF_THE_GODS = 'http://api.handofthegods.com/handofthegodsapi.svc'
```

```
    HIREZ = 'https://api.hirezstudios.com'
```

```
    PALADINS = 'http://api.paladins.com/paladinsapi.svc'
```

```
    PALADINS_STRIKE = 'http://api.paladinsstrike.com/paladinsstrike.svc'
```

```
    REALM_ROYALE = 'http://api.realmroyale.com/realmapi.svc'
```

```
    SMITE = 'http://api.smitegame.com/smiteapi.svc'
```

```
    STATUS_PAGE = 'https://stk4xr7rly0r.statuspage.io'
```

```
class pyrez.enumerations.Format
```

```
    An enumeration.
```

```
    ATOM = 'atom'
```

```
    JSON = 'json'
```

```
    RSS = 'rss'
```

```
    XML = 'xml'
```

```
class pyrez.enumerations.Gods
```

```
    An enumeration.
```

```
    getCard ()
```

```
    getIcon ()
```

```
isAssassin()
isGuardian()
isHunter()
isMage()
isWarrior()
Achilles = 3492
Agni = 1737
Ah_Muzen_Cab = 1956
Ah_Puch = 2056
Amaterasu = 2110
Anhur = 1773
Anubis = 1668
Ao_Kuang = 2034
Aphrodite = 1898
Apollo = 1899
Arachne = 1699
Ares = 1782
Artemis = 1748
Artio = 3336
Athena = 1919
Awilix = 2037
Bacchus = 1809
Bakasura = 1755
Baron_Samedi = 3518
Bastet = 1678
Bellona = 2047
Cabranken = 2008
Camazotz = 2189
Cerberus = 3419
Cernunnos = 2268
Chaac = 1966
Change = 1921
Chernobog = 3509
Chiron = 2075
Chronos = 1920
Cu_Chulainn = 2319
```


Cupid = 1778
Da_Ji = 2270
Discordia = 3377
Erlang_Shen = 2138
Fafnir = 2136
Fenrir = 1843
Freya = 1784
Ganesha = 2269
Geb = 1978
Guan_Yu = 1763
Hachiman = 3344
Hades = 1676
He_Bo = 1674
Hel = 1718
Hera = 3558
Hercules = 1848
Horus = 3611
Hou_Yi = 2040
Hun_Batz = 1673
Isis = 1918
Izanami = 2179
Janus = 1999
Jing_Wei = 2122
Jormungandr = 3585
Kali = 1649
Khepri = 2066
King_Arthur = 3565
Kukulcan = 1677
Kumbhakarna = 1993
Kuzenbo = 2260
Loki = 1797
Medusa = 2051
Mercury = 1941
Merlin = 3566
Ne_Zha = 1915
Neith = 1872

```
Nemesis = 1980
Nike = 2214
Nox = 2036
Nu_Wa = 1958
Odin = 1669
Osiris = 2000
Pele = 3543
Poseidon = 1881
Ra = 1698
Raijin = 2113
Rama = 2002
Ratatoskr = 2063
Ravana = 2065
Scylla = 1988
Serqet = 2005
Set = 3612
Skadi = 2107
Sobek = 1747
Sol = 2074
Sun_Wukong = 1944
Susano = 2123
Sylvanus = 2030
Terra = 2147
Thanatos = 1943
The_Morrigan = 2226
Thor = 1779
Thoth = 2203
Tyr = 1924
Ullr = 1991
Vamana = 1723
Vulcan = 1869
Xbalanque = 1864
Xing_Tian = 2072
Ymir = 1670
Zeus = 1672
Zhong_Kui = 1926
```

```
class pyrez.enumerations.Language
    An enumeration.

    Chinese = 5
    English = 1
    French = 3
    German = 2
    Polish = 12
    Portuguese = 10
    Russian = 11
    Spanish = 7
    Spanish_Latin_America = 9
    Turkish = 13

class pyrez.enumerations.PortalId
    An enumeration.

    Discord = 25
    HiRez = 1
    PS4 = 9
    Steam = 5
    Switch = 22
    Unknown = -1
    Xbox = 10

class pyrez.enumerations.Region
    An enumeration.

    AUSTRALIA = 'Australia'
    BRAZIL = 'Brazil'
    EUROPE = 'Europe'
    LATIN_AMERICA_NORTH = 'Latin America North'
    NORTH_AMERICA = 'North America'
    SOUTHEAST_ASIA = 'Southeast Asia'

class pyrez.enumerations.QueuePaladins
    An enumeration.

    Challenge_FP = 442
    Challenge_IP = 443
    Challenge_Match = 426
    Challenge_TP = 441
    Classic_Siege = 465
    Custom_Event_End_Times = 489
```

```
Custom_Onslaught_Foreman_Rise = 462
Custom_Onslaught_Magistrate_Archives = 464
Custom_Onslaught_Marauders_Port = 483
Custom_Onslaught_Primal_Court = 455
Custom_Onslaught_Snowfall_Junction = 454
Custom_Siege_Ascension_Peak = 473
Custom_Siege_Brightmarsh = 458
Custom_Siege_Fish_Market = 431
Custom_Siege_Frog_Isle = 433
Custom_Siege_Frozen_Guard = 432
Custom_Siege_Ice_Mines = 439
Custom_Siege_Jaguar_Falls = 438
Custom_Siege_Serpeant_Beach = 440
Custom_Siege_Shattered_Desert = 487
Custom_Siege_Splitstone_Quarry = 459
Custom_Siege_Stone_Keep = 423
Custom_Siege_Timber_Mill = 430
Custom_Siege_Warders_Gate = 485
Custom_Team_Deathmatch_Abyss = 479
Custom_Team_Deathmatch_Dragon_Arena = 484
Custom_Team_Deathmatch_Foreman_Rise = 471
Custom_Team_Deathmatch_Magistrates_Archives = 472
Custom_Team_Deathmatch_Throne = 480
Custom_Team_Deathmatch_Trade_District = 468
Live_Battlegrounds_Duo = 475
Live_Battlegrounds_Quad = 476
Live_Battlegrounds_Solo = 474
Live_Competitive_GamePad = 428
Live_Competitive_Keyboard = 486
Live_Event_Ascension_Peak = 477
Live_Event_End_Times = 488
Live_Event_Rise_Of_Furia = 478
Live_Onslaught = 452
Live_Practice_Onslaught = 453
Live_Practice_Siege = 425
Live_Practice_Team_Deathmatch = 470
```

```

Live_Siege = 424
Live_Team_DeathMatch = 469
Live_Test_Maps = 445
Multi_Queue = 999
Payload = 437
Perf_Capture_Map = 435
Practice = 427
PvE_Hands_That_Bind = 446
PvE_HnS = 449
PvE_Survival = 451
Shooting_Range = 434
Tencent_Alpha_Test_Queue_Coop = 436
Tutorial = 444
WIPpVE_High_Rollers = 448
WIPpVE_Leap_Frogs = 450
WIPpVE_Los_Pollos_Fernandos = 447
zzRETIRED = 429
class pyrez.enumerations.QueueRealmRoyale
    An enumeration.
    Live_Duo = 475
    Live_Duo_Low_Level = 484
    Live_Duo_Mid_Level = 483
    Live_Solo = 474
    Live_Solo_Low_Level = 480
    Live_Solo_Mid_Level = 479
    Live_Squad = 476
    Live_Squad_Low_Level = 482
    Live_Squad_Mid_Level = 481
    Live_Tutorial = 478
    Live_Wars = 477
class pyrez.enumerations.QueueSmite
    For Smite, queue_id's 426, 435, 440, 445, 448, 451, 459, & 450 are the only ones considered for player win/loss
    stats from /getplayer.
    Adventure_CH10 = 500
    Adventure_Horde = 495
    Adventure_Joust = 499
    Arena_League = 452

```

```
Arena_Practice_Easy = 443
Arena_Practice_Medium = 472
Arena_Queue = 435
Arena_Training = 483
Arena_Tutorial = 462
Arena_vs_AI_Easy = 457
Arena_vs_AI_Medium = 468
Assault = 445
Assault_Practice_Easy = 479
Assault_Practice_Medium = 480
Assault_vs_AI_Easy = 481
Assault_vs_AI_Medium = 454
Basic_Tutorial = 436
Clash = 466
Clash_Practice_Easy = 470
Clash_Practice_Medium = 477
Clash_Tutorial = 471
Clash_vs_AI_Easy = 478
Clash_vs_AI_Medium = 469
Conquest = 426
Conquest_5v5 = 423
Conquest_Practice_Easy = 458
Conquest_Practice_Medium = 475
Conquest_Ranked_GamePad = 504
Conquest_Ranked_Keyboard = 451
Conquest_Tutorial = 463
Conquest_vs_AI_Easy = 476
Conquest_vs_AI_Medium = 461
Custom_Arena = 438
Custom_Assault = 446
Custom_Clash = 467
Custom_Conquest = 429
Custom_Joust = 441
Custom_Siege = 460
Domination = 433
Domination_Challenge = 439
```

```
Joust_1v1_Ranked_GamePad = 502
Joust_1v1_Ranked_Keyboard = 440
Joust_3v3_Ranked_GamePad = 503
Joust_3v3_Ranked_Keyboard = 450
Joust_3v3_Training = 482
Joust_Practice_Easy = 464
Joust_Practice_Medium = 473
Joust_Queue_3v3 = 448
Joust_vs_AI_Easy = 474
Joust_vs_AI_Medium = 456
Jungle_Practice = 444
Jungle_Practice_Presele_ = 496
Loki_Dungeon = 501
MOTD = 434
Novice_Queue = 424
Practice = 427
Siege_4v4 = 459

class pyrez.enumerations.Status
    An enumeration.

    isInGame()
    isOnline()
    God_Selection = 2
    In_Game = 3
    In_Lobby = 1
    Not_Found = 5
    Offline = 0
    Online = 4

class pyrez.enumerations.Tier
    An enumeration.

    Bronze_I = 5
    Bronze_II = 4
    Bronze_III = 3
    Bronze_IV = 2
    Bronze_V = 1
    Diamond_I = 25
    Diamond_II = 24
    Diamond_III = 23
```

```
Diamond_IV = 22
Diamond_V = 21
Gold_I = 15
Gold_II = 14
Gold_III = 13
Gold_IV = 12
Gold_V = 11
Grandmaster = 27
Master = 26
Platinum_I = 20
Platinum_II = 19
Platinum_III = 18
Platinum_IV = 17
Platinum_V = 16
Silver_I = 10
Silver_II = 9
Silver_III = 8
Silver_IV = 7
Silver_V = 6
Unranked = 0
```

Exceptions (pyrez.exceptions)

The following exceptions are thrown by the library.

```
exception pyrez.exceptions.PyrezException (*args, **kwargs)
    Base class for all other Pyrez exceptions.

exception pyrez.exceptions.DailyLimit (*args, **kwargs)

exception pyrez.exceptions.Deprecated (*args, **kwargs)

exception pyrez.exceptions.IdOrAuthEmpty (*args, **kwargs)

exception pyrez.exceptions.InvalidArgument (*args, **kwargs)

exception pyrez.exceptions.MatchException (*args, **kwargs)

exception pyrez.exceptions.NoResult (*args, **kwargs)

exception pyrez.exceptions.NotFound (*args, **kwargs)

exception pyrez.exceptions.NotSupported (*args, **kwargs)

exception pyrez.exceptions.PaladinsOnly (*args, **kwargs)

exception pyrez.exceptions.PlayerNotFound (*args, **kwargs)

exception pyrez.exceptions.RealmRoyaleOnly (*args, **kwargs)

exception pyrez.exceptions.RequestError (*args, **kwargs)

exception pyrez.exceptions.SessionLimit (*args, **kwargs)

exception pyrez.exceptions.SmiteOnly (*args, **kwargs)

exception pyrez.exceptions.UnexpectedException (*args, **kwargs)

exception pyrez.exceptions.WrongCredentials (*args, **kwargs)
```

Data classes (pyrez.models)

These are the classes created by API responses. They hold the data that is responded from the API server in an easy to use way. You shouldn't create these yourself.

```
class pyrez.models.Ability (**kwargs)
```

```
class pyrez.models.APIResponse (**kwargs)
```

Represents a generic Pyrez object. This is a sub-class of *APIResponseBase*.

errorMsg [str] The message returned from the API request.

hasError ()

```
class pyrez.models.APIResponseBase (**kwargs)
```

Represents a generic Pyrez object.

Keyword Arguments **json** (dict or list) – The request as JSON, if you prefer.

```
class pyrez.models.BaseMatchDetail (**kwargs)
```

```
class pyrez.models.DataUsed (**kwargs)
```

concurrentSessionsLeft ()

requestsLeft ()

sessionsLeft ()

```
class pyrez.models.DemoDetails (**kwargs)
```

```
class pyrez.models.EsportProLeague (**kwargs)
```

An important return value is “matchStatus” which represents a match being:

- scheduled (1),
- in-progress (2),

- complete (3)

```
class pyrez.models.Friend (**kwargs)
class pyrez.models.God (**kwargs)
class pyrez.models.InGameItem (itemID, itemName, itemLevel)
class pyrez.models.Item (**kwargs)
class pyrez.models.ItemDescription (**kwargs)
class pyrez.models.LeagueLeaderboard (**kwargs)
class pyrez.models.LeagueSeason (**kwargs)
class pyrez.models.LiveMatch (**kwargs)
```

getMapName (_clear=False)

```
class pyrez.models.Match (**kwargs)
class pyrez.models.MatchBase (**kwargs)
class pyrez.models.MatchHistory (**kwargs)
class pyrez.models.MatchId (**kwargs)
class pyrez.models.Menuitem (**kwargs)
class pyrez.models.MergedPlayer (**kwargs)
class pyrez.models.MOTD (**kwargs)
class pyrez.models.PatchInfo (**kwargs)
class pyrez.models.Ping (kwargs)
class pyrez.models.Player (**kwargs)
class pyrez.models.PlayerAchievements (**kwargs)
class pyrez.models.PlayerBase (**kwargs)
class pyrez.models.PlayerId (**kwargs)
class pyrez.models.PlayerPS (**kwargs)
class pyrez.models.PlayerStatus (**kwargs)
class pyrez.models.QueueStats (**kwargs)
class pyrez.models.Ranked (**kwargs)
```

hasPlayedRanked ()

```
class pyrez.models.ServerStatus (**kwargs)
class pyrez.models.Session (**kwargs)
```

isApproved ()

```
class pyrez.models.Skin (**kwargs)
class pyrez.models.TestSession (kwargs)
class pyrez.models.StatusPage (**kwargs)
```

```

class pyrez.models.HiRez.AccountInfo (**kwargs)
class pyrez.models.HiRez.ContactInfo (**kwargs)
class pyrez.models.HiRez.Game (**kwargs)
class pyrez.models.HiRez.LinkedAccount (**kwargs)
class pyrez.models.HiRez.PortalAccount (**kwargs)
class pyrez.models.HiRez.Transaction (**kwargs)
class pyrez.models.HiRez.UserInfo (**kwargs)
class pyrez.models.Paladins.Champion (**kwargs)
class pyrez.models.Paladins.ChampionAbility (**kwargs)
class pyrez.models.Paladins.ChampionCard (**kwargs)

    getCardURL ()
    getIconURL ()
class pyrez.models.Paladins.ChampionSkin (**kwargs)
class pyrez.models.Paladins.Item (**kwargs)
class pyrez.models.Paladins.Loadout (**kwargs)
class pyrez.models.Paladins.LoadoutItem (**kwargs)

    getCard ()
    getFrame ()
class pyrez.models.Paladins.Player (**kwargs)
class pyrez.models.Paladins.Post (**kwargs)
class pyrez.models.RealmRoyale.Leaderboard (**kwargs)
class pyrez.models.RealmRoyale.LeaderboardDetails (**kwargs)
class pyrez.models.RealmRoyale.Match (**kwargs)
class pyrez.models.RealmRoyale.MatchHistory (**kwargs)
class pyrez.models.RealmRoyale.Player (**kwargs)
class pyrez.models.RealmRoyale.Talent (**kwargs)
class pyrez.models.Smite.God (**kwargs)
class pyrez.models.Smite.GodLeaderboard (**kwargs)
class pyrez.models.Smite.GodRank (**kwargs)
class pyrez.models.Smite.GodRecommendedItem (**kwargs)
class pyrez.models.Smite.GodSkin (**kwargs)
class pyrez.models.Smite.Item (**kwargs)
class pyrez.models.Smite.Player (**kwargs)
class pyrez.models.Smite.TopMatch (**kwargs)
class pyrez.models.StatusPage.AffectedComponents (**kwargs)

```

```
class pyrez.models.StatusPage.Base (**kwargs)
class pyrez.models.StatusPage.Component (**kwargs)
class pyrez.models.StatusPage.ComponentMixin (**kwargs)
class pyrez.models.StatusPage.IncidentInfo (**kwargs)
class pyrez.models.StatusPage.Page (**kwargs)
class pyrez.models.StatusPage.Status (**kwargs)
class pyrez.models.StatusPage.StatusPage (**kwargs)
class pyrez.models.StatusPage.Incidents (**kwargs)
class pyrez.models.StatusPage.IncidentUpdates (**kwargs)
class pyrez.models.StatusPage.ScheduledMaintenances (**kwargs)
```

CHAPTER 6

If you still can't find what you're looking for, try in one of the following pages:

- [genindex](#)
- [search](#)

p

- `pyrez.enumerations.Classes`, 35
- `pyrez.exceptions`, 45
- `pyrez.models`, 47
 - `pyrez.models.HiRez`, 48
 - `pyrez.models.Paladins`, 49
 - `pyrez.models.RealmRoyale`, 49
 - `pyrez.models.Smite`, 49
 - `pyrez.models.StatusPage`, 49

A

- Ability (class in pyrez.models), 47
- AccountInfo (class in pyrez.models.HiRez), 48
- Achilles (pyrez.enumerations.Gods attribute), 36
- Adventure_CH10 (pyrez.enumerations.QueueSmite attribute), 41
- Adventure_Horde (pyrez.enumerations.QueueSmite attribute), 41
- Adventure_Joust (pyrez.enumerations.QueueSmite attribute), 41
- AffectedComponents (class in pyrez.models.StatusPage), 49
- Agni (pyrez.enumerations.Gods attribute), 36
- Ah_Muzen_Cab (pyrez.enumerations.Gods attribute), 36
- Ah_Puch (pyrez.enumerations.Gods attribute), 36
- Amaterasu (pyrez.enumerations.Gods attribute), 36
- Androxus (pyrez.enumerations.Champions attribute), 34
- Anhur (pyrez.enumerations.Gods attribute), 36
- Anubis (pyrez.enumerations.Gods attribute), 36
- Ao_Kuang (pyrez.enumerations.Gods attribute), 36
- Aphrodite (pyrez.enumerations.Gods attribute), 36
- APIResponse (class in pyrez.models), 47
- APIResponseBase (class in pyrez.models), 47
- Apollo (pyrez.enumerations.Gods attribute), 36
- Arachne (pyrez.enumerations.Gods attribute), 36
- Arena_League (pyrez.enumerations.QueueSmite attribute), 41
- Arena_Practice_Easy (pyrez.enumerations.QueueSmite attribute), 41
- Arena_Practice_Medium (pyrez.enumerations.QueueSmite attribute), 42
- Arena_Queue (pyrez.enumerations.QueueSmite attribute), 42
- Arena_Training (pyrez.enumerations.QueueSmite attribute), 42
- Arena_Tutorial (pyrez.enumerations.QueueSmite attribute), 42
- Arena_vs_AI_Easy (pyrez.enumerations.QueueSmite attribute), 42
- Arena_vs_AI_Medium (pyrez.enumerations.QueueSmite attribute), 42
- Ares (pyrez.enumerations.Gods attribute), 36
- Artemis (pyrez.enumerations.Gods attribute), 36
- Artio (pyrez.enumerations.Gods attribute), 36
- Ash (pyrez.enumerations.Champions attribute), 34
- Assassin (pyrez.enumerations.Classes.Classes attribute), 35
- Assault (pyrez.enumerations.QueueSmite attribute), 42
- Assault_Practice_Easy (pyrez.enumerations.QueueSmite attribute), 42
- Assault_Practice_Medium (pyrez.enumerations.QueueSmite attribute), 42
- Assault_vs_AI_Easy (pyrez.enumerations.QueueSmite attribute), 42
- Assault_vs_AI_Medium (pyrez.enumerations.QueueSmite attribute), 42
- Athena (pyrez.enumerations.Gods attribute), 36
- Atlas (pyrez.enumerations.Champions attribute), 34
- ATOM (pyrez.enumerations.Format attribute), 35
- AUSTRALIA (pyrez.enumerations.Region attribute), 39
- authKey (pyrez.api.PaladinsAPI attribute), 6
- authKey (pyrez.api.RealmRoyaleAPI attribute), 15
- authKey (pyrez.api.SmiteAPI attribute), 22
- Awilix (pyrez.enumerations.Gods attribute), 36

B

- Bacchus (pyrez.enumerations.Gods attribute), 36
- Bakasura (pyrez.enumerations.Gods attribute), 36
- Barik (pyrez.enumerations.Champions attribute), 34
- Baron_Samedi (pyrez.enumerations.Gods attribute), 36
- Base (class in pyrez.models.StatusPage), 49
- BaseMatchDetail (class in pyrez.models), 47
- Basic_Tutorial (pyrez.enumerations.QueueSmite attribute), 42
- Bastet (pyrez.enumerations.Gods attribute), 36

Bellona (*pyrez.enumerations.Gods attribute*), 36
Bomb_King (*pyrez.enumerations.Champions attribute*), 34
BRAZIL (*pyrez.enumerations.Region attribute*), 39
Bronze_I (*pyrez.enumerations.Tier attribute*), 43
Bronze_II (*pyrez.enumerations.Tier attribute*), 43
Bronze_III (*pyrez.enumerations.Tier attribute*), 43
Bronze_IV (*pyrez.enumerations.Tier attribute*), 43
Bronze_V (*pyrez.enumerations.Tier attribute*), 43
Buck (*pyrez.enumerations.Champions attribute*), 34

C

Cabrakan (*pyrez.enumerations.Gods attribute*), 36
Camazotz (*pyrez.enumerations.Gods attribute*), 36
Cassie (*pyrez.enumerations.Champions attribute*), 34
Cerberus (*pyrez.enumerations.Gods attribute*), 36
Cernunnos (*pyrez.enumerations.Gods attribute*), 36
Chaac (*pyrez.enumerations.Gods attribute*), 36
Challenge_FP (*pyrez.enumerations.QueuePaladins attribute*), 39
Challenge_IP (*pyrez.enumerations.QueuePaladins attribute*), 39
Challenge_Match (*pyrez.enumerations.QueuePaladins attribute*), 39
Challenge_TP (*pyrez.enumerations.QueuePaladins attribute*), 39
Champion (*class in pyrez.models.Paladins*), 49
ChampionAbility (*class in pyrez.models.Paladins*), 49
ChampionCard (*class in pyrez.models.Paladins*), 49
Champions (*class in pyrez.enumerations*), 33
ChampionSkin (*class in pyrez.models.Paladins*), 49
Change (*pyrez.enumerations.Gods attribute*), 36
Chernobog (*pyrez.enumerations.Gods attribute*), 36
Chinese (*pyrez.enumerations.Language attribute*), 39
Chiron (*pyrez.enumerations.Gods attribute*), 36
Chronos (*pyrez.enumerations.Gods attribute*), 36
Clash (*pyrez.enumerations.QueueSmite attribute*), 42
Clash_Practice_Easy (*pyrez.enumerations.QueueSmite attribute*), 42
Clash_Practice_Medium (*pyrez.enumerations.QueueSmite attribute*), 42
Clash_Tutorial (*pyrez.enumerations.QueueSmite attribute*), 42
Clash_vs_AI_Easy (*pyrez.enumerations.QueueSmite attribute*), 42
Clash_vs_AI_Medium (*pyrez.enumerations.QueueSmite attribute*), 42
Classes (*class in pyrez.enumerations.Classes*), 35
Classic_Siege (*pyrez.enumerations.QueuePaladins attribute*), 39
Component (*class in pyrez.models.StatusPage*), 50
ComponentMixin (*class in pyrez.models.StatusPage*), 50

concurrentSessionsLeft () (*pyrez.models.DataUsed method*), 47
Conquest (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_5v5 (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_Practice_Easy (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_Practice_Medium (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_Ranked_GamePad (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_Ranked_Keyboard (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_Tutorial (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_vs_AI_Easy (*pyrez.enumerations.QueueSmite attribute*), 42
Conquest_vs_AI_Medium (*pyrez.enumerations.QueueSmite attribute*), 42
ContactInfo (*class in pyrez.models.HiRez*), 49
Cu_Chulainn (*pyrez.enumerations.Gods attribute*), 36
Cupid (*pyrez.enumerations.Gods attribute*), 36
Custom_Arena (*pyrez.enumerations.QueueSmite attribute*), 42
Custom_Assault (*pyrez.enumerations.QueueSmite attribute*), 42
Custom_Clash (*pyrez.enumerations.QueueSmite attribute*), 42
Custom_Conquest (*pyrez.enumerations.QueueSmite attribute*), 42
Custom_Event_End_Times (*pyrez.enumerations.QueuePaladins attribute*), 39
Custom_Joust (*pyrez.enumerations.QueueSmite attribute*), 42
Custom_Onslaught_Foreman_Rise (*pyrez.enumerations.QueuePaladins attribute*), 39
Custom_Onslaught_Magistrate_Archives (*pyrez.enumerations.QueuePaladins attribute*), 40
Custom_Onslaught_Marauders_Port (*pyrez.enumerations.QueuePaladins attribute*), 40
Custom_Onslaught_Primal_Court (*pyrez.enumerations.QueuePaladins attribute*), 40
Custom_Onslaught_Snowfall_Junction (*pyrez.enumerations.QueuePaladins attribute*), 40
Custom_Siege (*pyrez.enumerations.QueueSmite attribute*), 42
Custom_Siege_Ascension_Peak

- (*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Brightmarsh
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Fish_Market
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Frog_Isle
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Frozen_Guard
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Ice_Mines
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Jaguar_Falls
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Serpeant_Beach
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Shattered_Desert
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Splitstone_Quarry
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Stone_Keep
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Timber_Mill
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Siege_Warders_Gate
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Team_Deathmatch_Abyss
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Team_Deathmatch_Dragon_Arena
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Team_Deathmatch_Foreman_Rise
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Team_Deathmatch_Magistrates_Archives
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Team_Deathmatch_Throne
(*pyrez.enumerations.QueuePaladins* attribute), 40
- Custom_Team_Deathmatch_Trade_District
(*pyrez.enumerations.QueuePaladins* attribute), 40
- D**
- Da_Ji (*pyrez.enumerations.Gods* attribute), 37
- DailyLimit, 45
- DataUsed (class in *pyrez.models*), 47
- DemoDetails (class in *pyrez.models*), 47
- Deprecated, 45
- devId (*pyrez.api.PaladinsAPI* attribute), 6
- devId (*pyrez.api.RealmRoyaleAPI* attribute), 15
- devId (*pyrez.api.SmiteAPI* attribute), 22
- Diamond_I (*pyrez.enumerations.Tier* attribute), 43
- Diamond_II (*pyrez.enumerations.Tier* attribute), 43
- Diamond_III (*pyrez.enumerations.Tier* attribute), 43
- Diamond_IV (*pyrez.enumerations.Tier* attribute), 43
- Diamond_V (*pyrez.enumerations.Tier* attribute), 44
- Discord (*pyrez.enumerations.PortalId* attribute), 39
- Discordia (*pyrez.enumerations.Gods* attribute), 37
- Domination (*pyrez.enumerations.QueueSmite* attribute), 42
- Domination_Challenge
(*pyrez.enumerations.QueueSmite* attribute), 42
- Dredge (*pyrez.enumerations.Champions* attribute), 34
- Drogoz (*pyrez.enumerations.Champions* attribute), 34
- E**
- Endpoint (class in *pyrez.enumerations*), 35
- Engineer (*pyrez.enumerations.Classes.Classes* attribute), 35
- English (*pyrez.enumerations.Language* attribute), 39
- Enum (class in *pyrez.enumerations*), 33
- equal() (*pyrez.enumerations.Enum* method), 33
- Erlang_Shen (*pyrez.enumerations.Gods* attribute), 37
- EsportProLeague (class in *pyrez.models*), 47
- EUROPE (*pyrez.enumerations.Region* attribute), 39
- Evie (*pyrez.enumerations.Champions* attribute), 34
- F**
- Fafnir (*pyrez.enumerations.Gods* attribute), 37
- Fenrir (*pyrez.enumerations.Gods* attribute), 37
- Fernando (*pyrez.enumerations.Champions* attribute), 34
- Format (class in *pyrez.enumerations*), 35
- French (*pyrez.enumerations.Language* attribute), 39
- Freya (*pyrez.enumerations.Gods* attribute), 37
- Friend (class in *pyrez.models*), 48
- Furia (*pyrez.enumerations.Champions* attribute), 34
- G**
- Game (class in *pyrez.models.HiRez*), 49
- Ganesha (*pyrez.enumerations.Gods* attribute), 37
- Geb (*pyrez.enumerations.Gods* attribute), 37
- German (*pyrez.enumerations.Language* attribute), 39

[getCard\(\) \(pyrez.enumerations.Gods method\), 35](#)
[getCard\(\) \(pyrez.models.Paladins.LoadoutItem method\), 49](#)
[getCardURL\(\) \(pyrez.models.Paladins.ChampionCard method\), 49](#)
[getChampionCards\(\) \(pyrez.api.PaladinsAPI method\), 6](#)
[getChampionLeaderboard\(\) \(pyrez.api.PaladinsAPI method\), 6](#)
[getChampionRanks\(\) \(pyrez.api.PaladinsAPI method\), 7](#)
[getChampions\(\) \(pyrez.api.PaladinsAPI method\), 7](#)
[getChampionSkins\(\) \(pyrez.api.PaladinsAPI method\), 7](#)
[getComponents\(\) \(pyrez.api.StatusPageAPI method\), 30](#)
[getDataUsed\(\) \(pyrez.api.PaladinsAPI method\), 7](#)
[getDataUsed\(\) \(pyrez.api.RealmRoyaleAPI method\), 16](#)
[getDataUsed\(\) \(pyrez.api.SmiteAPI method\), 22](#)
[getDemoDetails\(\) \(pyrez.api.PaladinsAPI method\), 7](#)
[getDemoDetails\(\) \(pyrez.api.SmiteAPI method\), 23](#)
[getEndpoint\(\) \(pyrez.enumerations.Endpoint method\), 35](#)
[getEsportsProLeague\(\) \(pyrez.api.PaladinsAPI method\), 8](#)
[getEsportsProLeague\(\) \(pyrez.api.SmiteAPI method\), 23](#)
[getFrame\(\) \(pyrez.models.Paladins.LoadoutItem method\), 49](#)
[getFriends\(\) \(pyrez.api.PaladinsAPI method\), 8](#)
[getFriends\(\) \(pyrez.api.RealmRoyaleAPI method\), 16](#)
[getFriends\(\) \(pyrez.api.SmiteAPI method\), 23](#)
[getGodLeaderboard\(\) \(pyrez.api.PaladinsAPI method\), 8](#)
[getGodLeaderboard\(\) \(pyrez.api.SmiteAPI method\), 23](#)
[getGodRanks\(\) \(pyrez.api.PaladinsAPI method\), 8](#)
[getGodRanks\(\) \(pyrez.api.SmiteAPI method\), 24](#)
[getGodRecommendedItems\(\) \(pyrez.api.SmiteAPI method\), 24](#)
[getGods\(\) \(pyrez.api.PaladinsAPI method\), 9](#)
[getGods\(\) \(pyrez.api.SmiteAPI method\), 24](#)
[getGodSkins\(\) \(pyrez.api.PaladinsAPI method\), 9](#)
[getGodSkins\(\) \(pyrez.api.SmiteAPI method\), 24](#)
[getHeader\(\) \(pyrez.enumerations.Champions method\), 33](#)
[getHistory\(\) \(pyrez.api.StatusPageAPI method\), 31](#)
[getIcon\(\) \(pyrez.enumerations.Champions method\), 33](#)
[getIcon\(\) \(pyrez.enumerations.Gods method\), 35](#)
[getIconURL\(\) \(pyrez.models.Paladins.ChampionCard method\), 49](#)
[getId\(\) \(pyrez.enumerations.Enum method\), 33](#)
[getIncidents\(\) \(pyrez.api.StatusPageAPI method\), 31](#)
[getItems\(\) \(pyrez.api.PaladinsAPI method\), 9](#)
[getItems\(\) \(pyrez.api.RealmRoyaleAPI method\), 16](#)
[getItems\(\) \(pyrez.api.SmiteAPI method\), 25](#)
[getLatestPatchNotes\(\) \(pyrez.api.PaladinsAPI method\), 9](#)
[getLeaderboard\(\) \(pyrez.api.RealmRoyaleAPI method\), 17](#)
[getLeagueLeaderboard\(\) \(pyrez.api.PaladinsAPI method\), 9](#)
[getLeagueLeaderboard\(\) \(pyrez.api.SmiteAPI method\), 25](#)
[getLeagueSeasons\(\) \(pyrez.api.PaladinsAPI method\), 10](#)
[getLeagueSeasons\(\) \(pyrez.api.SmiteAPI method\), 25](#)
[getMapName\(\) \(pyrez.models.LiveMatch method\), 48](#)
[getMatch\(\) \(pyrez.api.PaladinsAPI method\), 10](#)
[getMatch\(\) \(pyrez.api.RealmRoyaleAPI method\), 17](#)
[getMatch\(\) \(pyrez.api.SmiteAPI method\), 25](#)
[getMatchHistory\(\) \(pyrez.api.PaladinsAPI method\), 10](#)
[getMatchHistory\(\) \(pyrez.api.RealmRoyaleAPI method\), 18](#)
[getMatchHistory\(\) \(pyrez.api.SmiteAPI method\), 26](#)
[getMatchIds\(\) \(pyrez.api.PaladinsAPI method\), 11](#)
[getMatchIds\(\) \(pyrez.api.RealmRoyaleAPI method\), 18](#)
[getMatchIds\(\) \(pyrez.api.SmiteAPI method\), 26](#)
[getMotd\(\) \(pyrez.api.SmiteAPI method\), 26](#)
[getName\(\) \(pyrez.enumerations.Enum method\), 33](#)
[getPatchInfo\(\) \(pyrez.api.PaladinsAPI method\), 11](#)
[getPatchInfo\(\) \(pyrez.api.RealmRoyaleAPI method\), 18](#)
[getPatchInfo\(\) \(pyrez.api.SmiteAPI method\), 27](#)
[getPlayer\(\) \(pyrez.api.PaladinsAPI method\), 11](#)
[getPlayer\(\) \(pyrez.api.RealmRoyaleAPI method\), 19](#)
[getPlayer\(\) \(pyrez.api.SmiteAPI method\), 27](#)
[getPlayerAchievements\(\) \(pyrez.api.PaladinsAPI method\), 12](#)
[getPlayerAchievements\(\) \(pyrez.api.RealmRoyaleAPI method\), 19](#)
[getPlayerAchievements\(\) \(pyrez.api.SmiteAPI method\), 27](#)
[getPlayerId\(\) \(pyrez.api.PaladinsAPI method\), 12](#)
[getPlayerId\(\) \(pyrez.api.RealmRoyaleAPI method\), 19](#)
[getPlayerId\(\) \(pyrez.api.SmiteAPI method\), 27](#)
[getPlayerLoadouts\(\) \(pyrez.api.PaladinsAPI method\), 12](#)

getPlayerStats() (*pyrez.api.RealmRoyaleAPI method*), 19
 getPlayerStatus() (*pyrez.api.PaladinsAPI method*), 13
 getPlayerStatus() (*pyrez.api.RealmRoyaleAPI method*), 19
 getPlayerStatus() (*pyrez.api.SmiteAPI method*), 28
 getQueueStats() (*pyrez.api.PaladinsAPI method*), 13
 getQueueStats() (*pyrez.api.RealmRoyaleAPI method*), 20
 getQueueStats() (*pyrez.api.SmiteAPI method*), 28
 getScheduledMaintenances() (*pyrez.api.StatusPageAPI method*), 31
 getServerStatus() (*pyrez.api.PaladinsAPI method*), 13
 getServerStatus() (*pyrez.api.RealmRoyaleAPI method*), 20
 getServerStatus() (*pyrez.api.SmiteAPI method*), 28
 getStatus() (*pyrez.api.StatusPageAPI method*), 31
 getSummary() (*pyrez.api.StatusPageAPI method*), 31
 getTeamDetails() (*pyrez.api.SmiteAPI method*), 29
 getTeamPlayers() (*pyrez.api.SmiteAPI method*), 29
 getTopMatches() (*pyrez.api.SmiteAPI method*), 29
 getWebsitePost() (*pyrez.api.PaladinsAPI method*), 14
 God (*class in pyrez.models*), 48
 God (*class in pyrez.models.Smite*), 49
 God_Selection (*pyrez.enumerations.Status attribute*), 43
 GodLeaderboard (*class in pyrez.models.Smite*), 49
 GodRank (*class in pyrez.models.Smite*), 49
 GodRecommendedItem (*class in pyrez.models.Smite*), 49
 Gods (*class in pyrez.enumerations*), 35
 GodSkin (*class in pyrez.models.Smite*), 49
 Gold_I (*pyrez.enumerations.Tier attribute*), 44
 Gold_II (*pyrez.enumerations.Tier attribute*), 44
 Gold_III (*pyrez.enumerations.Tier attribute*), 44
 Gold_IV (*pyrez.enumerations.Tier attribute*), 44
 Gold_V (*pyrez.enumerations.Tier attribute*), 44
 Grandmaster (*pyrez.enumerations.Tier attribute*), 44
 Grohk (*pyrez.enumerations.Champions attribute*), 34
 Grover (*pyrez.enumerations.Champions attribute*), 34
 Guan_Yu (*pyrez.enumerations.Gods attribute*), 37
H
 Hachiman (*pyrez.enumerations.Gods attribute*), 37
 Hades (*pyrez.enumerations.Gods attribute*), 37
 HAND_OF_THE_GODS (*pyrez.enumerations.Endpoint attribute*), 35
 hasError() (*pyrez.models.APIResponse method*), 47
 hasPlayedRanked() (*pyrez.models.Ranked method*), 48
 He_Bo (*pyrez.enumerations.Gods attribute*), 37
 Hel (*pyrez.enumerations.Gods attribute*), 37
 Hera (*pyrez.enumerations.Gods attribute*), 37
 Hercules (*pyrez.enumerations.Gods attribute*), 37
 HIREZ (*pyrez.enumerations.Endpoint attribute*), 35
 HiRez (*pyrez.enumerations.PortalId attribute*), 39
 Horus (*pyrez.enumerations.Gods attribute*), 37
 Hou_Yi (*pyrez.enumerations.Gods attribute*), 37
 Hun_Batz (*pyrez.enumerations.Gods attribute*), 37
 Hunter (*pyrez.enumerations.Classes.Classes attribute*), 35
I
 IdOrAuthEmpty, 45
 Imani (*pyrez.enumerations.Champions attribute*), 34
 In_Game (*pyrez.enumerations.Status attribute*), 43
 In_Lobby (*pyrez.enumerations.Status attribute*), 43
 Inara (*pyrez.enumerations.Champions attribute*), 34
 IncidentInfo (*class in pyrez.models.StatusPage*), 50
 Incidents (*class in pyrez.models.StatusPage*), 50
 IncidentUpdates (*class in pyrez.models.StatusPage*), 50
 InGameItem (*class in pyrez.models*), 48
 InvalidArgument, 45
 isApproved() (*pyrez.models.Session method*), 48
 isAssassin() (*pyrez.enumerations.Gods method*), 35
 isDamage() (*pyrez.enumerations.Champions method*), 34
 isFlank() (*pyrez.enumerations.Champions method*), 34
 isFrontline() (*pyrez.enumerations.Champions method*), 34
 isGuardian() (*pyrez.enumerations.Gods method*), 36
 isHunter() (*pyrez.enumerations.Gods method*), 36
 isInGame() (*pyrez.enumerations.Status method*), 43
 Isis (*pyrez.enumerations.Gods attribute*), 37
 isMage() (*pyrez.enumerations.Gods method*), 36
 isOnline() (*pyrez.enumerations.Status method*), 43
 isSupport() (*pyrez.enumerations.Champions method*), 34
 isWarrior() (*pyrez.enumerations.Gods method*), 36
 Item (*class in pyrez.models*), 48
 Item (*class in pyrez.models.Paladins*), 49
 Item (*class in pyrez.models.Smite*), 49
 ItemDescription (*class in pyrez.models*), 48
 Izanami (*pyrez.enumerations.Gods attribute*), 37
J
 Janus (*pyrez.enumerations.Gods attribute*), 37
 Jenos (*pyrez.enumerations.Champions attribute*), 34
 Jing_Wei (*pyrez.enumerations.Gods attribute*), 37
 Jormungandr (*pyrez.enumerations.Gods attribute*), 37

Joust_1v1_Ranked_GamePad
(*pyrez.enumerations.QueueSmite* attribute), 42
Joust_1v1_Ranked_Keyboard
(*pyrez.enumerations.QueueSmite* attribute), 43
Joust_3v3_Ranked_GamePad
(*pyrez.enumerations.QueueSmite* attribute), 43
Joust_3v3_Ranked_Keyboard
(*pyrez.enumerations.QueueSmite* attribute), 43
Joust_3v3_Training
(*pyrez.enumerations.QueueSmite* attribute), 43
Joust_Practice_Easy
(*pyrez.enumerations.QueueSmite* attribute), 43
Joust_Practice_Medium
(*pyrez.enumerations.QueueSmite* attribute), 43
Joust_Queue_3v3 (*pyrez.enumerations.QueueSmite*
attribute), 43
Joust_vs_AI_Easy (*pyrez.enumerations.QueueSmite*
attribute), 43
Joust_vs_AI_Medium
(*pyrez.enumerations.QueueSmite* attribute), 43
JSON (*pyrez.enumerations.Format* attribute), 35
Jungle_Practice (*pyrez.enumerations.QueueSmite*
attribute), 43
Jungle_Practice_Presele_
(*pyrez.enumerations.QueueSmite* attribute), 43

K

Kali (*pyrez.enumerations.Gods* attribute), 37
Khan (*pyrez.enumerations.Champions* attribute), 34
Khepri (*pyrez.enumerations.Gods* attribute), 37
Kinessa (*pyrez.enumerations.Champions* attribute), 34
King_Arthur (*pyrez.enumerations.Gods* attribute), 37
Koga (*pyrez.enumerations.Champions* attribute), 34
Kukulkan (*pyrez.enumerations.Gods* attribute), 37
Kumbhakarna (*pyrez.enumerations.Gods* attribute), 37
Kuzenbo (*pyrez.enumerations.Gods* attribute), 37

L

Language (class in *pyrez.enumerations*), 38
LATIN_AMERICA_NORTH (*pyrez.enumerations.Region*
attribute), 39
Leaderboard (class in *pyrez.models.RealmRoyale*), 49
LeaderboardDetails (class in
pyrez.models.RealmRoyale), 49
LeagueLeaderboard (class in *pyrez.models*), 48
LeagueSeason (class in *pyrez.models*), 48
Lex (*pyrez.enumerations.Champions* attribute), 34
Lian (*pyrez.enumerations.Champions* attribute), 34
LinkedAccount (class in *pyrez.models.HiRez*), 49
Live_Battlegrounds_Duo
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Battlegrounds_Quad
(*pyrez.enumerations.QueuePaladins* attribute),

40
Live_Battlegrounds_Solo
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Competitive_GamePad
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Competitive_Keyboard
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Duo (*pyrez.enumerations.QueueRealmRoyale* at-
tribute), 41
Live_Duo_Low_Level
(*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Duo_Mid_Level
(*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Event_Ascension_Peak
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Event_End_Times
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Event_Rise_Of_Furia
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Onslaught (*pyrez.enumerations.QueuePaladins*
attribute), 40
Live_Practice_Onslaught
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Practice_Siege
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Practice_Team_Deathmatch
(*pyrez.enumerations.QueuePaladins* attribute),
40
Live_Siege (*pyrez.enumerations.QueuePaladins* at-
tribute), 40
Live_Solo (*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Solo_Low_Level
(*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Solo_Mid_Level
(*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Squad (*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Squad_Low_Level
(*pyrez.enumerations.QueueRealmRoyale*
attribute), 41
Live_Squad_Mid_Level

(*pyrez.enumerations.QueueRealmRoyale attribute*), 41
 Live_Team_DeathMatch (*pyrez.enumerations.QueuePaladins attribute*), 41
 Live_Test_Maps (*pyrez.enumerations.QueuePaladins attribute*), 41
 Live_Tutorial (*pyrez.enumerations.QueueRealmRoyale attribute*), 41
 Live_Wars (*pyrez.enumerations.QueueRealmRoyale attribute*), 41
 LiveMatch (*class in pyrez.models*), 48
 Loadout (*class in pyrez.models.Paladins*), 49
 LoadoutItem (*class in pyrez.models.Paladins*), 49
 Loki (*pyrez.enumerations.Gods attribute*), 37
 Loki_Dungeon (*pyrez.enumerations.QueueSmite attribute*), 43

M

Maeve (*pyrez.enumerations.Champions attribute*), 34
 Mage (*pyrez.enumerations.Classes.Classes attribute*), 35
 makeRequest () (*pyrez.api.PaladinsAPI method*), 14
 makeRequest () (*pyrez.api.RealmRoyaleAPI method*), 20
 makeRequest () (*pyrez.api.SmiteAPI method*), 29
 Makoa (*pyrez.enumerations.Champions attribute*), 34
 Mal_Damba (*pyrez.enumerations.Champions attribute*), 34
 Master (*pyrez.enumerations.Tier attribute*), 44
 Match (*class in pyrez.models*), 48
 Match (*class in pyrez.models.RealmRoyale*), 49
 MatchBase (*class in pyrez.models*), 48
 MatchException, 45
 MatchHistory (*class in pyrez.models*), 48
 MatchHistory (*class in pyrez.models.RealmRoyale*), 49
 MatchId (*class in pyrez.models*), 48
 Medusa (*pyrez.enumerations.Gods attribute*), 37
 MenuItem (*class in pyrez.models*), 48
 Mercury (*pyrez.enumerations.Gods attribute*), 37
 MergedPlayer (*class in pyrez.models*), 48
 Merlin (*pyrez.enumerations.Gods attribute*), 37
 Moji (*pyrez.enumerations.Champions attribute*), 34
 MOTD (*class in pyrez.models*), 48
 MOTD (*pyrez.enumerations.QueueSmite attribute*), 43
 Multi_Queue (*pyrez.enumerations.QueuePaladins attribute*), 41

N

Ne_Zha (*pyrez.enumerations.Gods attribute*), 37
 Neith (*pyrez.enumerations.Gods attribute*), 37
 Nemesis (*pyrez.enumerations.Gods attribute*), 37
 Nike (*pyrez.enumerations.Gods attribute*), 38
 NoResult, 45

NORTH_AMERICA (*pyrez.enumerations.Region attribute*), 39
 Not_Found (*pyrez.enumerations.Status attribute*), 43
 NotFound, 45
 NotSupported, 45
 Novice_Queue (*pyrez.enumerations.QueueSmite attribute*), 43
 Nox (*pyrez.enumerations.Gods attribute*), 38
 Nu_Wa (*pyrez.enumerations.Gods attribute*), 38

O

Odin (*pyrez.enumerations.Gods attribute*), 38
 Offline (*pyrez.enumerations.Status attribute*), 43
 Online (*pyrez.enumerations.Status attribute*), 43
 onSessionCreated (*pyrez.api.PaladinsAPI attribute*), 6
 onSessionCreated (*pyrez.api.RealmRoyaleAPI attribute*), 15
 onSessionCreated (*pyrez.api.SmiteAPI attribute*), 22
 Osiris (*pyrez.enumerations.Gods attribute*), 38

P

Page (*class in pyrez.models.StatusPage*), 50
 PALADINS (*pyrez.enumerations.Endpoint attribute*), 35
 PALADINS_STRIKE (*pyrez.enumerations.Endpoint attribute*), 35
 PaladinsAPI (*class in pyrez.api*), 5
 PaladinsOnly, 45
 PatchInfo (*class in pyrez.models*), 48
 Payload (*pyrez.enumerations.QueuePaladins attribute*), 41
 Pele (*pyrez.enumerations.Gods attribute*), 38
 Perf_Capture_Map (*pyrez.enumerations.QueuePaladins attribute*), 41
 Ping (*class in pyrez.models*), 48
 ping () (*pyrez.api.PaladinsAPI method*), 14
 ping () (*pyrez.api.RealmRoyaleAPI method*), 21
 ping () (*pyrez.api.SmiteAPI method*), 29
 Pip (*pyrez.enumerations.Champions attribute*), 34
 Platinum_I (*pyrez.enumerations.Tier attribute*), 44
 Platinum_II (*pyrez.enumerations.Tier attribute*), 44
 Platinum_III (*pyrez.enumerations.Tier attribute*), 44
 Platinum_IV (*pyrez.enumerations.Tier attribute*), 44
 Platinum_V (*pyrez.enumerations.Tier attribute*), 44
 Player (*class in pyrez.models*), 48
 Player (*class in pyrez.models.Paladins*), 49
 Player (*class in pyrez.models.RealmRoyale*), 49
 Player (*class in pyrez.models.Smite*), 49
 PlayerAchievements (*class in pyrez.models*), 48
 PlayerBase (*class in pyrez.models*), 48
 PlayerId (*class in pyrez.models*), 48
 PlayerNotFound, 45
 PlayerPS (*class in pyrez.models*), 48

PlayerStatus (class in *pyrez.models*), 48
 Polish (*pyrez.enumerations.Language* attribute), 39
 PortalAccount (class in *pyrez.models.HiRez*), 49
 PortalId (class in *pyrez.enumerations*), 39
 Portuguese (*pyrez.enumerations.Language* attribute), 39
 Poseidon (*pyrez.enumerations.Gods* attribute), 38
 Post (class in *pyrez.models.Paladins*), 49
 Practice (*pyrez.enumerations.QueuePaladins* attribute), 41
 Practice (*pyrez.enumerations.QueueSmite* attribute), 43
 PS4 (*pyrez.enumerations.PortalId* attribute), 39
 PvE_Hands_That_Bind (*pyrez.enumerations.QueuePaladins* attribute), 41
 PvE_HnS (*pyrez.enumerations.QueuePaladins* attribute), 41
 PvE_Survival (*pyrez.enumerations.QueuePaladins* attribute), 41
pyrez.enumerations.Classes (module), 35
pyrez.exceptions (module), 45
pyrez.models (module), 47
pyrez.models.HiRez (module), 48
pyrez.models.Paladins (module), 49
pyrez.models.RealmRoyale (module), 49
pyrez.models.Smite (module), 49
pyrez.models.StatusPage (module), 49
 PyrezException, 45

Q

QueuePaladins (class in *pyrez.enumerations*), 39
 QueueRealmRoyale (class in *pyrez.enumerations*), 41
 QueueSmite (class in *pyrez.enumerations*), 41
 QueueStats (class in *pyrez.models*), 48

R

Ra (*pyrez.enumerations.Gods* attribute), 38
 Raijin (*pyrez.enumerations.Gods* attribute), 38
 Rama (*pyrez.enumerations.Gods* attribute), 38
 Ranked (class in *pyrez.models*), 48
 Ratatoskr (*pyrez.enumerations.Gods* attribute), 38
 Ravana (*pyrez.enumerations.Gods* attribute), 38
 REALM_ROYALE (*pyrez.enumerations.Endpoint* attribute), 35
 RealmRoyaleAPI (class in *pyrez.api*), 15
 RealmRoyaleOnly, 45
 Region (class in *pyrez.enumerations*), 39
 RequestError, 45
 requestsLeft() (*pyrez.models.DataUsed* method), 47
 responseFormat (*pyrez.api.PaladinsAPI* attribute), 6
 responseFormat (*pyrez.api.RealmRoyaleAPI* attribute), 16

responseFormat (*pyrez.api.SmiteAPI* attribute), 22
 RSS (*pyrez.enumerations.Format* attribute), 35
 Ruckus (*pyrez.enumerations.Champions* attribute), 34
 Russian (*pyrez.enumerations.Language* attribute), 39

S

ScheduledMaintenances (class in *pyrez.models.StatusPage*), 50
 Scylla (*pyrez.enumerations.Gods* attribute), 38
 searchPlayers() (*pyrez.api.PaladinsAPI* method), 14
 searchPlayers() (*pyrez.api.RealmRoyaleAPI* method), 21
 searchPlayers() (*pyrez.api.SmiteAPI* method), 30
 searchTeams() (*pyrez.api.SmiteAPI* method), 30
 Seris (*pyrez.enumerations.Champions* attribute), 34
 Serqet (*pyrez.enumerations.Gods* attribute), 38
 ServerStatus (class in *pyrez.models*), 48
 Session (class in *pyrez.models*), 48
 sessionId (*pyrez.api.PaladinsAPI* attribute), 6
 sessionId (*pyrez.api.RealmRoyaleAPI* attribute), 16
 sessionId (*pyrez.api.SmiteAPI* attribute), 22
 SessionLimit, 45
 sessionsLeft() (*pyrez.models.DataUsed* method), 47
 Set (*pyrez.enumerations.Gods* attribute), 38
 Sha_Lin (*pyrez.enumerations.Champions* attribute), 34
 Shooting_Range (*pyrez.enumerations.QueuePaladins* attribute), 41
 Siege_4v4 (*pyrez.enumerations.QueueSmite* attribute), 43
 Silver_I (*pyrez.enumerations.Tier* attribute), 44
 Silver_II (*pyrez.enumerations.Tier* attribute), 44
 Silver_III (*pyrez.enumerations.Tier* attribute), 44
 Silver_IV (*pyrez.enumerations.Tier* attribute), 44
 Silver_V (*pyrez.enumerations.Tier* attribute), 44
 Skadi (*pyrez.enumerations.Gods* attribute), 38
 Skin (class in *pyrez.models*), 48
 Skye (*pyrez.enumerations.Champions* attribute), 34
 SMITE (*pyrez.enumerations.Endpoint* attribute), 35
 SmiteAPI (class in *pyrez.api*), 21
 SmiteOnly, 45
 Sobek (*pyrez.enumerations.Gods* attribute), 38
 Sol (*pyrez.enumerations.Gods* attribute), 38
 SOUTHEAST_ASIA (*pyrez.enumerations.Region* attribute), 39
 Spanish (*pyrez.enumerations.Language* attribute), 39
 Spanish_Latin_America (*pyrez.enumerations.Language* attribute), 39
 Status (class in *pyrez.enumerations*), 43
 Status (class in *pyrez.models.StatusPage*), 50
 STATUS_PAGE (*pyrez.enumerations.Endpoint* attribute), 35

StatusPage (class in *pyrez.models*), 48
 StatusPage (class in *pyrez.models.StatusPage*), 50
 statusPage (*pyrez.api.PaladinsAPI* attribute), 6
 statusPage (*pyrez.api.RealmRoyaleAPI* attribute), 16
 statusPage (*pyrez.api.SmiteAPI* attribute), 22
 StatusPageAPI (class in *pyrez.api*), 30
 Steam (*pyrez.enumerations.PortalId* attribute), 39
 storeSession (*pyrez.api.PaladinsAPI* attribute), 6
 storeSession (*pyrez.api.RealmRoyaleAPI* attribute), 16
 storeSession (*pyrez.api.SmiteAPI* attribute), 22
 Strix (*pyrez.enumerations.Champions* attribute), 34
 Sun_Wukong (*pyrez.enumerations.Gods* attribute), 38
 Susano (*pyrez.enumerations.Gods* attribute), 38
 Switch (*pyrez.enumerations.PortalId* attribute), 39
 Sylvanus (*pyrez.enumerations.Gods* attribute), 38

T

Talent (class in *pyrez.models.RealmRoyale*), 49
 Talus (*pyrez.enumerations.Champions* attribute), 35
 Tencent_Alpha_Test_Queue_Coop
 (*pyrez.enumerations.QueuePaladins* attribute), 41
 Terminus (*pyrez.enumerations.Champions* attribute), 35
 Terra (*pyrez.enumerations.Gods* attribute), 38
 TestSession (class in *pyrez.models*), 48
 testSession() (*pyrez.api.PaladinsAPI* method), 14
 testSession() (*pyrez.api.RealmRoyaleAPI* method), 21
 testSession() (*pyrez.api.SmiteAPI* method), 30
 Thanatos (*pyrez.enumerations.Gods* attribute), 38
 The_Morrigan (*pyrez.enumerations.Gods* attribute), 38
 Thor (*pyrez.enumerations.Gods* attribute), 38
 Thoth (*pyrez.enumerations.Gods* attribute), 38
 Tier (class in *pyrez.enumerations*), 43
 TopMatch (class in *pyrez.models.Smite*), 49
 Torvald (*pyrez.enumerations.Champions* attribute), 35
 Transaction (class in *pyrez.models.HiRez*), 49
 Turkish (*pyrez.enumerations.Language* attribute), 39
 Tutorial (*pyrez.enumerations.QueuePaladins* attribute), 41
 Tyr (*pyrez.enumerations.Gods* attribute), 38
 Tyra (*pyrez.enumerations.Champions* attribute), 35

U

Ullr (*pyrez.enumerations.Gods* attribute), 38
 UnexpectedException, 45
 Unknown (*pyrez.enumerations.PortalId* attribute), 39
 Unranked (*pyrez.enumerations.Tier* attribute), 44
 UserInfo (class in *pyrez.models.HiRez*), 49

V

Vamana (*pyrez.enumerations.Gods* attribute), 38
 Viktor (*pyrez.enumerations.Champions* attribute), 35
 Vivian (*pyrez.enumerations.Champions* attribute), 35
 Vulcan (*pyrez.enumerations.Gods* attribute), 38

W

Warrior (*pyrez.enumerations.Classes.Classes* attribute), 35
 Willo (*pyrez.enumerations.Champions* attribute), 35
 WIPPE_High_Rollers
 (*pyrez.enumerations.QueuePaladins* attribute), 41
 WIPPE_Leap_Frogs
 (*pyrez.enumerations.QueuePaladins* attribute), 41
 WIPPE_Los_Pollos_Fernandos
 (*pyrez.enumerations.QueuePaladins* attribute), 41
 WrongCredentials, 45

X

Xbalanque (*pyrez.enumerations.Gods* attribute), 38
 Xbox (*pyrez.enumerations.PortalId* attribute), 39
 Xing_Tian (*pyrez.enumerations.Gods* attribute), 38
 XML (*pyrez.enumerations.Format* attribute), 35

Y

Ying (*pyrez.enumerations.Champions* attribute), 35
 Ymir (*pyrez.enumerations.Gods* attribute), 38

Z

Zeus (*pyrez.enumerations.Gods* attribute), 38
 Zhin (*pyrez.enumerations.Champions* attribute), 35
 Zhong_Kui (*pyrez.enumerations.Gods* attribute), 38
 zzRETIRED (*pyrez.enumerations.QueuePaladins* attribute), 41