nhl

Matt Hostetter

Feb 09, 2024

API REFERENCE

The *nhl* library is a Python 3 package for NHL game and player stats.

CHAPTER

ONE

NHL

A Python 3 API for NHL game and player stats.

class nhl.**Conference**(*id: int*, *name: str*, *name_short: str*, *abbreviation: str*)

NHL conference object.

abbreviation : str

Conference abbreviation

id : int

The NHL statsapi universal conference ID

name : str

Conference name

name_short : str

Conference name shortened

class nhl.**Division**(*id: int*, *name: str*, *name_short: str*, *abbreviation: str*)

NHL division object.

abbreviation : str

Division abbreviated name

id : int

The NHL statsapi universal division ID

name : str

Division name

name_short : str

Division short name

class nhl.**Event**(*game_id: int*, *id: int*, *type: str*, *subtype: str*, *time:* GameTime, *location:* Location, *value: float*, *score: tuple[int, int]*, *by_player_id: int*, *on_player_id: int*, *by_team_id: int*, *on_team_id: int*)

NHL event object.

class nhl.**Franchise**(*id: int*, *name: str*)

NHL franchise object.

id : int

The NHL statsapi universal franchise ID

name : str

Franchise name

class nhl.**GameTime**(*period: int*, *period_sec: int*)

NHL game time object.

This object represents a unique time of the game. There are convenience properties to convert the game time into convenient formats.

property min_sec : tuple[int, int]

Elapsed minutes and seconds of the game.

period : int

Game period. 1-3 for regulation. 4+ for overtime.

property period_min_sec : tuple[int, int]

Elapsed minutes and seconds of the period.

period_sec : int

Elapsed seconds of the period.

property period_str : str

Period number as string (i.e. "2nd")

property sec : int

Elapsed seconds of the game.

class nhl.**Location**(*x: int | None = None*, *y: int | None = None*)

NHL location object.

distance(other)

Measure distance between current location and another on-ice location.

Parameters

other : *Location*
location to measure distance from

Returns
distance (ft)

Return type
float

x : int | None = **None**

Rink position in ft along x-axis (length) [-100, 100]

y : int | None = **None**

Rink position in ft along y-axis (breadth) [-42, 42]

class nhl.**Official**(*id: int*, *name: str*)

NHL official object.

property first_name : str

Official's first name

id : int

The NHL statsapi universal official ID

property last_name : str

Official's last name

name : str

Official's full name

class nhl.**Player**(*id: int*, *name: str*, *number: int*, *position: str*, *height: int*, *weight: int*, *shoots_catches: str*, *birth_date: date*, *birth_city: str*, *birth_country: str*)

NHL player object.

property age : int

Current age in years

birth_city : str

Player's birth city

birth_country : str

Player's birth country

birth_date : date

Player's birth date

property first_name : str

Player's first name

height : int

Player's height in total inches

property height_ft_in : tuple[int, int]

Height in feet and inches (*height* // 12, *height* % 12)

id : int

The NHL statsapi universal player ID

property last_name : str

Player's last name

name : str

Player's full name

number : int

Player's primary number

position : str

Player's primary position ("LW", "C", "RW", "D", "G")

shoots_catches : str

Indication of whether the player shoots (skater)/catches (goalie) "L" or "R"

weight : int

Player's weight in lbs

class nhl.**Shift**(*game_id: int*, *player_id: int*, *shift_id: int*, *on:* GameTime, *off:* GameTime)

NHL shift object.

game_id : int

NHL statsapi unique game ID

length : int

Shift length in seconds

off : *GameTime*

Shift end game time

on : *GameTime*

Shift start game time

player_id : int

NHL statsapi unique player ID

shift_id : int

Shift number for specified game

class nhl.**Team**(*id: int*, *location: str*, *name: str*, *abbreviation: str*, *first_year: int*, *division:* Division, *conference:* Conference, *franchise:* Franchise)

NHL team object.

abbreviation : str

Team's name abbreviated

conference : *Conference*

The NHL conference the team is in

division : *Division*

The NHL division the team is in

first_year : int

First year of play

franchise : *Franchise*

The NHL franchise the team belongs to

property full_name : str

Team's full name

id : int

The NHL statsapi universal team ID

location : str

Team's location

name : str

Team's name

class nhl.**Venue**(*id: int*, *name: str*)

NHL venue object.

id : int

The NHL statsapi universal venue ID

name : str

Venue name

CHAPTER

TWO

NHL.STATSAPI

A module containing functions for querying the NHL statsapi REST API https://statsapi.web.nhl.com/api/v1.

nhl.statsapi.**conference**(id: *int*) → *Conference*

Fetches a single conference by its ID.

Parameters

id: int
The NHL statsapi conference ID.

Returns
A *Conference* object.

nhl.statsapi.**conferences**() → list[*Conference*]

Fetches all conferences.

Returns
A list of *Conference* objects.

nhl.statsapi.**division**(id: *int*) → *Division*

Fetches a single division by its ID.

Parameters

id: int
The NHL statsapi division ID.

Returns
A *Division* object.

nhl.statsapi.**divisions**() → list[*Division*]

Fetches all divisions.

Returns
A list of *Division* objects.

nhl.statsapi.**franchise**(id: *int*) → *Franchise*

Fetches a single franchise by its ID.

Parameters

id: int
The NHL statsapi franchise ID.

Returns
A *Franchise* object.

nhl.statsapi.**franchises**() → list[*Franchise*]

Fetches all franchises.

Returns
A list of *Franchise* objects.

nhl.statsapi.**player**(id: *int*) → *Player*

Fetches a single player by its ID.

Parameters

id: int
The NHL statsapi player ID.

Returns
A *Player* object.

nhl.statsapi.**players**(*ids: Sequence[int]*) → list[*Player*]

Fetches multiple players by their IDs.

Returns
A list of *Player* objects.

nhl.statsapi.**shifts**(game_id: *int*, player_id: *int*) → list[*Shift*]

Fetches all shifts of a player of a given game.

Parameters

game_id: int
The NHL statsapi game ID.

player_id: int
The NHL statsapi player ID.

Returns
A list of *Shift* objects.

Raises
ValueError – If the player is not found in the game.

nhl.statsapi.**team**(id: *int*) → *Team*

Fetches a single team by its ID.

Parameters

id: int
The NHL statsapi team ID.

Returns
A *Team* object.

nhl.statsapi.**teams**() → list[*Team*]

Fetches all teams.

Returns
A list of *Team* objects.

nhl.statsapi.**venue**(id: *int*) → *Venue*

Fetches a single venue by its ID.

Parameters

id: int
The NHL statsapi venue ID.

Returns
A *Venue* object.

nhl.statsapi.**venues**() → list[*Venue*]

Fetches all venues.

Returns
A list of *Venue* objects.

CHAPTER

THREE

VERSIONING

The *nhl* library uses semantic versioning. Releases are versioned `major.minor.patch`.

Major versions introduce API-changing features. Minor versions add features that are backwards-compatible with other releases. Patch versions make backwards-compatible bug fixes.

3.1 Alpha releases

Versions before `0.1.0` are alpha releases. Alpha releases are versioned `0.0.alpha`. There is no API compatibility guarantee for them. They can be thought of as `0.0.alpha-major`.

3.2 Beta releases

Versions before `1.0.0` are beta releases. Beta releases are versioned `0.beta.x` and are API-compatible. They can be thought of as `0.beta-major.beta-minor`.

PYTHON MODULE INDEX

n

INDEX

O

P

S

T

V

W

X

Y