

---

# **bundesliga-tippspiel-reminder**

*Release 0.2.1*

**Hermann Krumrey**

**Sep 13, 2019**



# CONTENTS

<b>1</b>	<b>bundesliga_tippspiel_reminder package</b>	<b>3</b>
1.1	Subpackages . . . . .	3
1.2	Submodules . . . . .	4
1.3	bundesliga_tippspiel_reminder.BundesligaTippspielReminderBot module . . . . .	4
1.4	bundesliga_tippspiel_reminder.BundesligaTippspielReminderParser module . . . . .	5
1.5	bundesliga_tippspiel_reminder.api module . . . . .	5
1.6	Module contents . . . . .	6
<b>2</b>	<b>bundesliga_tippspiel_reminder</b>	<b>7</b>
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



Contents:



## BUNDESLIGA\_TIPPSPIEL\_REMINDER PACKAGE

### 1.1 Subpackages

#### 1.1.1 bundesliga\_tippspiel\_reminder.db package

##### Submodules

##### bundesliga\_tippspiel\_reminder.db.ApiKey module

**class** `bundesliga_tippspiel_reminder.db.ApiKey.ApiKey` (\*\*kwargs)

Bases: `sqlalchemy.ext.declarative.api.Base`

Stores the API key for a user

`__init__` (\*\*kwargs)

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in kwargs.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

**id**

The ID of the API key

**key**

The actual API key

**kudubot\_user**

The kudubot address

**kudubot\_user\_id**

The kudubot address ID

**tippspiel\_user**

The bundesliga-tippspiel user

##### bundesliga\_tippspiel\_reminder.db.Reminder module

**class** `bundesliga_tippspiel_reminder.db.Reminder.Reminder` (\*\*kwargs)

Bases: `sqlalchemy.ext.declarative.api.Base`

Database table that defines a reminder.

**\_\_init\_\_** (*\*\*kwargs*)

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in *kwargs*.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

**get\_due\_matches** (*matches: List[Dict[str, Any]], bets: List[Dict[str, Any]]*) → List[Dict[str, Any]]

Retrieves any due matches for the reminder settings :param matches: All matches :param bets: All bets of the user :return: A list of due matches

**id**

The ID of the reminder

**kudubot\_user**

The kudubot address

**kudubot\_user\_id**

The kudubot address ID

**last\_reminder**

The time when the last reminder was sent. Format in the form %Y-%m-%d:%H-%M-%S

**property last\_reminder\_datetime**

**Returns** The 'last\_reminder' parameter as a datetime object

**reminder\_time**

The time before the next unbet match when the reminder message will be sent. Unit: seconds

**property reminder\_time\_delta**

**Returns** The 'reminder\_time' parameter as a datetime timedelta

## Module contents

### 1.2 Submodules

### 1.3 bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot module

**class** bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot.**BundesligaTippspielRemi**

Bases: kudubot.Bot.Bot

The bundesliga tippspiel reminder bot



**bg\_iteration** (*\_*: int, *db\_session*: sqlalchemy.orm.session.Session)  
 Implements behaviours of the Background thread that periodically checks if any reminders are due :return: None

**property bg\_pause**  
 The pause between background iterations :return: The pause in seconds

**is\_authorized** (*address*: kudubot.db.Address.Address, *\_*: Dict[str, Any], *db\_session*: sqlalchemy.orm.session.Session) → bool  
 Checks if a user is authorized :param address: The user to check :param \_: possible command arguments :param db\_session: The database session to use :return: True if authorized, False otherwise

**classmethod name** () → str  
**Returns** The name of the bot

**classmethod parsers** () → List[kudubot.parsing.CommandParser.CommandParser]  
**Returns** The parsers for the bot

**classmethod unauthorized\_message** () → str  
**Returns** A custom message sent to a user if they tried to access a feature that requires authorization without being authorized

**classmethod version** () → str  
**Returns** The current version of the bot

## 1.4 bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderParser module

**class** bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderParser.**BundesligaTippspielReminderParser**  
 Bases: kudubot.parsing.CommandParser.CommandParser  
 Parser for the bot that defines the available commands

**classmethod commands** () → List[kudubot.parsing.Command.Command]  
**Returns** The available commands

**classmethod name** () → str  
**Returns** The name of the bot

## 1.5 bundesliga\_tippspiel\_reminder.api module

bundesliga\_tippspiel\_reminder.api.**api\_is\_authorized** (*api\_key*: str) → bool  
 Checks whether or not an API key is authorized :param api\_key: The API key to check :return: True if authorized, False otherwise

bundesliga\_tippspiel\_reminder.api.**api\_request** (*endpoint*: str, *method*: str, *params*: Dict[Any, Any], *api\_key*: Optional[str] = None) → Dict[Any, Any]  
 Sends a request to the API :param endpoint: The endpoint to send the request to :param method: The HTTP method to use :param params: The parameters to use :param api\_key: The API key to use :return: The response

## 1.6 Module contents

```
bundesliga_tippspiel_reminder.sentry_dsn = 'https://bb7b3f62b88d48309ed1539af4a53b6a@sentry
```

The sentry DSN for this project

```
bundesliga_tippspiel_reminder.version = '0.2.1'
```

The current version of otaku\_info\_bot

**BUNDESLIGA\_TIPPSPIEL\_REMINDER**



## INDICES AND TABLES

- genindex
- modindex
- search



## PYTHON MODULE INDEX

### b

bundesliga\_tippspiel\_reminder, 6

bundesliga\_tippspiel\_reminder.api, 5

bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot,  
4

bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderParser,  
5

bundesliga\_tippspiel\_reminder.db, 4

bundesliga\_tippspiel\_reminder.db.ApiKey,  
3

bundesliga\_tippspiel\_reminder.db.Reminder,  
3





## Symbols

**\_\_init\_\_()** (*bundesliga\_tippspiel\_reminder.db.ApiKey.ApiKey* **method**), 3  
**\_\_init\_\_()** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **method**), 3

**A**  
**api\_is\_authorized()** (*in module bundesliga\_tippspiel\_reminder.api*), 5  
**api\_request()** (*in module bundesliga\_tippspiel\_reminder.api*), 5  
**ApiKey** (*class in bundesliga\_tippspiel\_reminder.db.ApiKey*), 3

**B**  
**bg\_iteration()** (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot* **method**), 4  
**bg\_pause()** (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot* **property**), 5  
**bundesliga\_tippspiel\_reminder** (*module*), 6  
**bundesliga\_tippspiel\_reminder.api** (*module*), 5  
**bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot** (*module*), 4  
**bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderParser** (*module*), 5  
**bundesliga\_tippspiel\_reminder.db** (*module*), 4  
**bundesliga\_tippspiel\_reminder.db.ApiKey** (*module*), 3  
**bundesliga\_tippspiel\_reminder.db.Reminder** (*module*), 3  
**BundesligaTippspielReminderBot** (*class in bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot*), 4  
**BundesligaTippspielReminderParser** (*class in bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderParser*), 5

**C**  
**commands()** (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot* **class method**), 5

**G**  
**get\_due\_matches()** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **method**), 4

**I**  
**id** (*bundesliga\_tippspiel\_reminder.db.ApiKey.ApiKey* **attribute**), 3  
**id** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **attribute**), 4  
**is\_authorized()** (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot* **method**), 4

**K**  
**key** (*bundesliga\_tippspiel\_reminder.db.ApiKey.ApiKey* **attribute**), 3  
**kudubot\_user** (*bundesliga\_tippspiel\_reminder.db.ApiKey.ApiKey* **attribute**), 3  
**kudubot\_user** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **attribute**), 4  
**kudubot\_user\_id** (*bundesliga\_tippspiel\_reminder.db.ApiKey.ApiKey* **attribute**), 3  
**kudubot\_user\_id** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **attribute**), 4

**L**  
**last\_reminder** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **attribute**), 4  
**last\_reminder\_datetime()** (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder* **property**), 4

## N

`name()` (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot.BundesligaTippspielReminderBot class method*), 5

`name()` (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderParser.BundesligaTippspielReminderParser class method*), 5

## P

`parsers()` (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot.BundesligaTippspielReminderBot class method*), 5

## R

`Reminder` (*class in bundesliga\_tippspiel\_reminder.db.Reminder*), 3

`reminder_time` (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder attribute*), 4

`reminder_time_delta()` (*bundesliga\_tippspiel\_reminder.db.Reminder.Reminder property*), 4

## S

`sentry_dsn` (*in module bundesliga\_tippspiel\_reminder*), 6

## T

`tippspiel_user` (*bundesliga\_tippspiel\_reminder.db.ApiKey.ApiKey attribute*), 3

## U

`unauthorized_message()` (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot.BundesligaTippspielReminderBot class method*), 5

## V

`version` (*in module bundesliga\_tippspiel\_reminder*), 6

`version()` (*bundesliga\_tippspiel\_reminder.BundesligaTippspielReminderBot.BundesligaTippspielReminderBot class method*), 5