

---

# **gramhopper Documentation**

*Release 1.0.7*

**Or Bin, Meir Halachmi**

**Feb 28, 2022**



---

# Content

---

<b>1</b>	<b>Triggers</b>	<b>1</b>
1.1	Event Streak Trigger . . . . .	1
1.2	Filter Triggers . . . . .	2
1.3	Text Triggers . . . . .	3
<b>2</b>	<b>Responses</b>	<b>5</b>
2.1	Match Responses . . . . .	5
2.2	Preset Responses . . . . .	6
	<b>Index</b>	<b>9</b>



## 1.1 Event Streak Trigger

### 1.1.1 event\_streak

```
class EventStreakTrigger (streak_timeout_sec: float, event_count: int,  
                          counting_event_trigger: gramhopper.triggers.basic_triggers.BaseTrigger,  
                          resetting_event_trigger: Optional[gramhopper.triggers.basic_triggers.BaseTrigger] = None)
```

Event-streak trigger. This is used to trigger a rule in a case of a streak of events, for example when a certain phrase is written a few times in a row.

This trigger gets a defined trigger as a “counting” event and optionally another one as a “resetting” event (to stop counting). It also gets the count of events to be considered a streak, and the timeout of the streak (the allowed time period between the first event and the last event in the streak).

```
__init__ (streak_timeout_sec: float, event_count: int, counting_event_trigger: gramhopper.triggers.basic_triggers.BaseTrigger,  
          resetting_event_trigger: Optional[gramhopper.triggers.basic_triggers.BaseTrigger] = None)
```

Constructs the trigger.

#### Parameters

- **streak\_timeout\_sec** – The allowed time period between the first event and the last event in the streak
- **event\_count** – The count of events to be considered a streak
- **counting\_event\_trigger** – A trigger to identify an event that counts towards the streak
- **resetting\_event\_trigger** – A trigger to identify an event that resets the streak

## 1.2 Filter Triggers

### class FilterTriggers

Text-based triggers.

**chat** = <class 'gramhopper.triggers.filter\_triggers.\_ChatFilterBasedTrigger'>

A user filter trigger. See more in [\\_ChatFilterBasedTrigger](#).

**language** = <class 'gramhopper.triggers.filter\_triggers.\_LanguageFilterBasedTrigger'>

A user filter trigger. See more in [\\_LanguageFilterBasedTrigger](#).

**message\_type** = <class 'gramhopper.triggers.filter\_triggers.\_MessageTypeFilterBasedTrigger'>

A user filter trigger. See more in [\\_MessageTypeFilterBasedTrigger](#).

**user** = <class 'gramhopper.triggers.filter\_triggers.\_UserFilterBasedTrigger'>

A user filter trigger. See more in [\\_UserFilterBasedTrigger](#).

### 1.2.1 filter.user

**class \_UserFilterBasedTrigger** (*nickname: str = None, user\_id: int = None, username: str = None*)

A user filter trigger. This is used to trigger a rule when an incoming message comes from a specific user.

**\_\_init\_\_** (*nickname: str = None, user\_id: int = None, username: str = None*)

Constructs the trigger. *nickname* can be used if such a nickname is defined in users.json file. Otherwise, one of *user\_id* and *username* should be specified.

#### Parameters

- **message\_filter** – The filter to test if the message passes through
- **nickname** – The nickname of the user to pass messages from.
- **user\_id** – The Telegram user ID of the user to pass messages from.
- **username** – The Telegram username of the user to pass messages from.

### 1.2.2 filter.chat

**class \_ChatFilterBasedTrigger** (*chat\_id: int = None, username: str = None*)

A chat filter trigger. This is used to trigger a rule when an incoming message is in a specific chat.

**\_\_init\_\_** (*chat\_id: int = None, username: str = None*)

Constructs the trigger. One and only one of *chat\_id* and *username* should be specified.

#### Parameters

- **chat\_id** – The Telegram chat ID of the chat to pass messages from.
- **username** – The Telegram username whose chat to pass messages from.

### 1.2.3 filter.language

**class \_LanguageFilterBasedTrigger** (*lang: Union[str, List[str]]*)

A language filter trigger. This is used to trigger a rule when an incoming message is in a specific language.

**\_\_init\_\_** (*lang: Union[str, List[str]]*)

Constructs the trigger.

**Parameters lang** – The language code/s in which to pass messages.

### 1.2.4 filter.message\_type

**class \_MessageTypeFilterBasedTrigger** (*message\_type*)

A message type filter trigger. This is used to trigger a rule when an incoming message is of a specific type.

**\_\_init\_\_** (*message\_type*)

Constructs the trigger.

**Parameters message\_type** – The message type of which to filter to pass messages, for example: 'photo', 'status\_update.left\_chat\_member' or 'document'. See more in `telegram.ext.filters.Filters`.

## 1.3 Text Triggers

**class TextTriggers**

Text-based triggers.

**has\_exact\_word** = <class 'gramhopper.triggers.text\_triggers.\_HasExactWordTrigger'>

A word trigger. See more in `_HasExactWordTrigger`.

**has\_substring** = <class 'gramhopper.triggers.text\_triggers.\_HasSubstringTrigger'>

A substring trigger. See more in `_HasSubstringTrigger`.

**regexp** = <class 'gramhopper.triggers.text\_triggers.\_RegExpTrigger'>

A regexp-based trigger. See more in `_RegExpTrigger`.

### 1.3.1 text.has\_exact\_word

**class \_HasExactWordTrigger** (*word: Union[str, List[str]]*)

Word trigger - the same as substring trigger, but for exact words search. This is used to trigger a rule when a certain word (or one of a list of words) exists in an incoming message.

**\_\_init\_\_** (*word: Union[str, List[str]]*)

Constructs the trigger.

**Parameters word** – The word/s to search in the message

### 1.3.2 text.has\_substring

**class \_HasSubstringTrigger** (*substring: Union[str, List[str]], exact: bool = False*)

Substring trigger. This is used to trigger a rule when a certain substring (or one of a list of substrings) exists in an incoming message.

**\_\_init\_\_** (*substring: Union[str, List[str]], exact: bool = False*)

Constructs the trigger.

**Parameters**

- **substring** – The substring/s to search in the message
- **exact** – Whether the exact substring should appear (as a whole word) in the message

### 1.3.3 text.regex

**class** `_RegExpTrigger` (*pattern: str*)

Regular-expression-based trigger. This is used to trigger a rule when an incoming message matches the given pattern.

`__init__` (*pattern: str*)

Constructs the trigger.

**Parameters** `pattern` – The pattern to test the message’s match with



## 2.1 Match Responses

There are some matched-based responses, which should normally come after a regexp trigger

### **class MatchResponses**

Regexp-based responses. These responses use the regexp match result from the trigger, as well as the given template, to build the response text.

**message** = <class 'gramhopper.responses.match\_responses.\_MatchMessageResponse'>

A regexp-based **message** response. See more in [\\_MatchMessageResponse](#).

**reply** = <class 'gramhopper.responses.match\_responses.\_MatchReplyResponse'>

A regexp-based **reply** response. See more in [\\_MatchReplyResponse](#).

### 2.1.1 match.message

**class \_MatchMessageResponse** (*template: str, parse\_mode: Optional[str] = None*)

A regexp-based response in which the response method is a normal message

**\_\_init\_\_** (*template: str, parse\_mode: Optional[str] = None*)

Constructs the response.

#### **Parameters**

- **template** – The template to use when building the response text
- **parse\_mode** – Optional parse mode for the message. Read more in [telegram.ParseMode](#).

### 2.1.2 match.reply

**class \_MatchReplyResponse** (*template: str, parse\_mode: Optional[str] = None*)

A regexp-based response in which the response method is a reply to the triggering message

```
__init__(template: str, parse_mode: Optional[str] = None)  
Constructs the response.
```

#### Parameters

- **template** – The template to use when building the response text
- **parse\_mode** – Optional parse mode for the message. Read more in `telegram.ParseMode`.

## 2.2 Preset Responses

### class PresetResponses

Preset responses. These responses use a preset response/s to respond with. If a list of responses is given, one of them will be chosen randomly for each response.

```
document = <class 'gramhopper.responses.preset_responses._PresetDocumentResponse'>  
A preset document response. See more in _PresetDocumentResponse.
```

```
message = <class 'gramhopper.responses.preset_responses._PresetMessageResponse'>  
A preset message response. See more in _PresetMessageResponse.
```

```
reply = <class 'gramhopper.responses.preset_responses._PresetReplyResponse'>  
A preset reply response. See more in _PresetReplyResponse.
```

### 2.2.1 preset.document

```
class _PresetDocumentResponse (preset_response: Union[str, telegram.files.document.Document],  
                                parse_mode: Optional[str] = None)
```

A preset response in which the response method is a document

```
__init__(preset_response: Union[str, telegram.files.document.Document], parse_mode: Op-  
tional[str] = None)  
Constructs the response.
```

#### Parameters

- **preset\_response** – The preset document URL or document object
- **parse\_mode** – Optional parse mode for the message. Read more in `telegram.ParseMode`.

### 2.2.2 preset.message

```
class _PresetMessageResponse (preset_response: Union[str, List[str]], parse_mode: Optional[str]  
                                = None)
```

A preset response in which the response method is a normal message

```
__init__(preset_response: Union[str, List[str]], parse_mode: Optional[str] = None)  
Constructs the response.
```

#### Parameters

- **preset\_response** – The preset response or list of responses
- **parse\_mode** – Optional parse mode for the message. Read more in `telegram.ParseMode`.

### 2.2.3 preset.reply

**class** `_PresetReplyResponse` (*preset\_response: Union[str, List[str]], parse\_mode: Optional[str] = None*)

A preset response in which the response method is a reply to the triggering message

`__init__` (*preset\_response: Union[str, List[str]], parse\_mode: Optional[str] = None*)

Constructs the response.

#### Parameters

- **preset\_response** – The preset response or list of responses
- **parse\_mode** – Optional parse mode for the message. Read more in `telegram.ParseMode`.



## Symbols

\_ChatFilterBasedTrigger (class in `gramhopper.triggers.filter_triggers`), 2  
 \_HasExactWordTrigger (class in `gramhopper.triggers.text_triggers`), 3  
 \_HasSubstringTrigger (class in `gramhopper.triggers.text_triggers`), 3  
 \_LanguageFilterBasedTrigger (class in `gramhopper.triggers.filter_triggers`), 2  
 \_MatchMessageResponse (class in `gramhopper.responses.match_responses`), 5  
 \_MatchReplyResponse (class in `gramhopper.responses.match_responses`), 5  
 \_MessageTypeFilterBasedTrigger (class in `gramhopper.triggers.filter_triggers`), 3  
 \_PresetDocumentResponse (class in `gramhopper.responses.preset_responses`), 6  
 \_PresetMessageResponse (class in `gramhopper.responses.preset_responses`), 6  
 \_PresetReplyResponse (class in `gramhopper.responses.preset_responses`), 7  
 \_RegExpTrigger (class in `gramhopper.triggers.text_triggers`), 4  
 \_UserFilterBasedTrigger (class in `gramhopper.triggers.filter_triggers`), 2  
 \_\_init\_\_() (EventStreakTrigger method), 1  
 \_\_init\_\_() (\_ChatFilterBasedTrigger method), 2  
 \_\_init\_\_() (\_HasExactWordTrigger method), 3  
 \_\_init\_\_() (\_HasSubstringTrigger method), 3  
 \_\_init\_\_() (\_LanguageFilterBasedTrigger method), 2  
 \_\_init\_\_() (\_MatchMessageResponse method), 5  
 \_\_init\_\_() (\_MatchReplyResponse method), 5  
 \_\_init\_\_() (\_MessageTypeFilterBasedTrigger method), 3  
 \_\_init\_\_() (\_PresetDocumentResponse method), 6  
 \_\_init\_\_() (\_PresetMessageResponse method), 6  
 \_\_init\_\_() (\_PresetReplyResponse method), 7  
 \_\_init\_\_() (\_RegExpTrigger method), 4  
 \_\_init\_\_() (\_UserFilterBasedTrigger method), 2

## C

chat (FilterTriggers attribute), 2

## D

document (PresetResponses attribute), 6

## E

EventStreakTrigger (class in `gramhopper.triggers.event_streak_trigger`), 1

## F

FilterTriggers (class in `gramhopper.triggers.filter_triggers`), 2

## H

has\_exact\_word (TextTriggers attribute), 3

has\_substring (TextTriggers attribute), 3

## L

language (FilterTriggers attribute), 2

## M

MatchResponses (class in `gramhopper.responses.match_responses`), 5  
 message (MatchResponses attribute), 5  
 message (PresetResponses attribute), 6  
 message\_type (FilterTriggers attribute), 2

## P

PresetResponses (class in `gramhopper.responses.preset_responses`), 6

## R

regexp (TextTriggers attribute), 3

reply (MatchResponses attribute), 5

reply (PresetResponses attribute), 6

## T

TextTriggers (class in gramhopper.triggers.text\_triggers),  
3

## U

user (FilterTriggers attribute), 2