
Chatfirst Python Client Documentation

Release a0

Ivan Tertychnyy

November 02, 2016

| | | |
|----------|---------------------------|-----------|
| 1 | Installation | 3 |
| 2 | Quickstart | 5 |
| 3 | Get the code | 7 |
| 4 | Contents: | 9 |
| 4.1 | Build adapter | 9 |
| 4.2 | Build channel | 10 |
| 5 | Indices and tables | 13 |

Chatfirst Python Client provides simple interface to [Chatfirst API](#)

Installation

```
pip install chatfirst
```

Quickstart

```
#!/usr/bin/env python

from chatfirst.client import Chatfirst

client = Chatfirst(your_chatfirst_user_token)
my_bots = client.bots_list()
```

Get the code

The [source](#) is available on Github.

Contents:

4.1 Build adapter

class `chatfirst.client.Chatfirst` (*token*)

The Chatfirst class is a wrapper of [Chatfirst API](#) This class provides easy access to [bots](#), [push](#), [talk](#), [broadcast methods](#)

Parameters `token` (*str*) – user’s token from Chatfirst dashboard

broadcast (*bot, channel_type, text*)

Use this method to broadcast text message to all users of bot.

Parameters

- **bot** ([Bot](#)) – bot that will push user
- **channel_type** (*str*) – one of [telegram, facebook, slack]
- **text** (*str*) – text message

push (*bot, channel_type, ar, user_id*)

Use this method to push message to user of bot. The message should be packed into `ActionResponse` object. This allows to push text messages, buttons, images. This also allows to force current state of user.

Parameters

- **bot** ([Bot](#)) – bot that will push user
- **channel_type** (*str*) – one of [telegram, facebook, slack]
- **ar** ([ActionResponse](#)) – message packed in response object
- **user_id** (*str*) – user id in used channel

class `chatfirst.models.ActionResponse` (*data={}*)

Main class defining `ActionResponse` entity

Parameters `data` (*dict*) – serialized object

count = `None`

ignore this parameter

entities = `None`

`data["Entities"]`, list of `LinkedEntity` objects

forced = `None`

`data["ForcedState"]`, forced transition to state (useful trick for scoped scenarios and error handling)

keyboard = None
data["ForcedKeyboard"], keys to be shown to user

messages = None
data["Messages"], list of messages to return as text messages

to_json()
Serialize object to json dict

Returns dict

class chatfirst.models.**LinkedEntity**(data={})
The way to show images in bot. Every object will be rendered as gallery or set of image messages.

Parameters **data** (*dict*) – serialized object

desc = None
data["Description"], description of image

name = None
data["Name"], name of image

options = None
data["EntityOptions"], keys to be shown to user

to_json()
Serialize object to json dict

Returns dict

url = None
data["ImageUrl"], url of image

4.2 Build channel

class chatfirst.client.**Chatfirst**(token)
The Chatfirst class is a wrapper of [Chatfirst API](#) This class provides easy access to [bots](#), [push](#), [talk](#), [broadcast](#) [methods](#)

Parameters **token** (*str*) – user's token from Chatfirst dashboard

bots_create (*bot*)
Save new bot

Parameters **bot** (*Bot*) – bot object to save

bots_delete (*bot*)
Delete existing bot

Parameters **bot** (*Bot*) – bot to delete

bots_get (*bot*)
Fetch and fill Bot object

Parameters **bot** (*Bot*) – empty bot object with name to search

Return type *Bot*

Returns filled bot object

bots_list ()
List all user's bots

Return type list of Bot

Returns user's bots

bots_update (*bot*)

Update Bot

Parameters **bot** (Bot) – bot object to update

talk (*bot, message*)

Talk to bot and get response based You can use this method to integrate the platform with your own channels

Parameters

- **bot** (Bot) – bot to talk to
- **message** (Message) – message to send to bot

Return type *ActionResponse*

Returns response object

class chatfirst.models.**Bot** (*data={}*)

Main class defining Bot entity

Parameters **data** (*dict*) – serialized object

fancy_name = None

data["fancy_name"], long name of bot

language = None

data["language"], 0 - Russian, 1 - English

name = None

data["name"], short name of bot

scenario = None

data["scenario"], scenario of bot

to_json ()

Serialize object to json dict

Returns dict

class chatfirst.models.**Message** (*data={}*)

Use this class if you want to integrate your own channel and use talk method. Chatfirst platform accepts incoming talk requests with special json object:

Parameters **data** (*dict*) – serialized object

first_name = None

data["FirstName"], first name of talking user

id_ = None

data["InterlocutorId"], internal id of talking user

last_name = None

data["LastName"], last name of talking user

text = None

data["Text"], text message that user sent

to_json ()

Serialize object to json dict

Returns dict

username = None

data["Username"], username of talking user

Indices and tables

- `genindex`
- `modindex`
- `search`

A

ActionResponse (class in chatfirst.models), 9

B

Bot (class in chatfirst.models), 11

bots_create() (chatfirst.client.Chatfirst method), 10

bots_delete() (chatfirst.client.Chatfirst method), 10

bots_get() (chatfirst.client.Chatfirst method), 10

bots_list() (chatfirst.client.Chatfirst method), 10

bots_update() (chatfirst.client.Chatfirst method), 11

broadcast() (chatfirst.client.Chatfirst method), 9

C

Chatfirst (class in chatfirst.client), 9, 10

count (chatfirst.models.ActionResponse attribute), 9

D

desc (chatfirst.models.LinkedEntity attribute), 10

E

entities (chatfirst.models.ActionResponse attribute), 9

F

fancy_name (chatfirst.models.Bot attribute), 11

first_name (chatfirst.models.Message attribute), 11

forced (chatfirst.models.ActionResponse attribute), 9

I

id_ (chatfirst.models.Message attribute), 11

K

keyboard (chatfirst.models.ActionResponse attribute), 9

L

language (chatfirst.models.Bot attribute), 11

last_name (chatfirst.models.Message attribute), 11

LinkedEntity (class in chatfirst.models), 10

M

Message (class in chatfirst.models), 11

messages (chatfirst.models.ActionResponse attribute), 10

N

name (chatfirst.models.Bot attribute), 11

name (chatfirst.models.LinkedEntity attribute), 10

O

options (chatfirst.models.LinkedEntity attribute), 10

P

push() (chatfirst.client.Chatfirst method), 9

S

scenario (chatfirst.models.Bot attribute), 11

T

talk() (chatfirst.client.Chatfirst method), 11

text (chatfirst.models.Message attribute), 11

to_json() (chatfirst.models.ActionResponse method), 10

to_json() (chatfirst.models.Bot method), 11

to_json() (chatfirst.models.LinkedEntity method), 10

to_json() (chatfirst.models.Message method), 11

U

url (chatfirst.models.LinkedEntity attribute), 10

username (chatfirst.models.Message attribute), 12