» Why use Ampalibe

Why use Ampalibe

Webhooks & process Management

No need to manage webhooks and data

messages are received directly in a main function

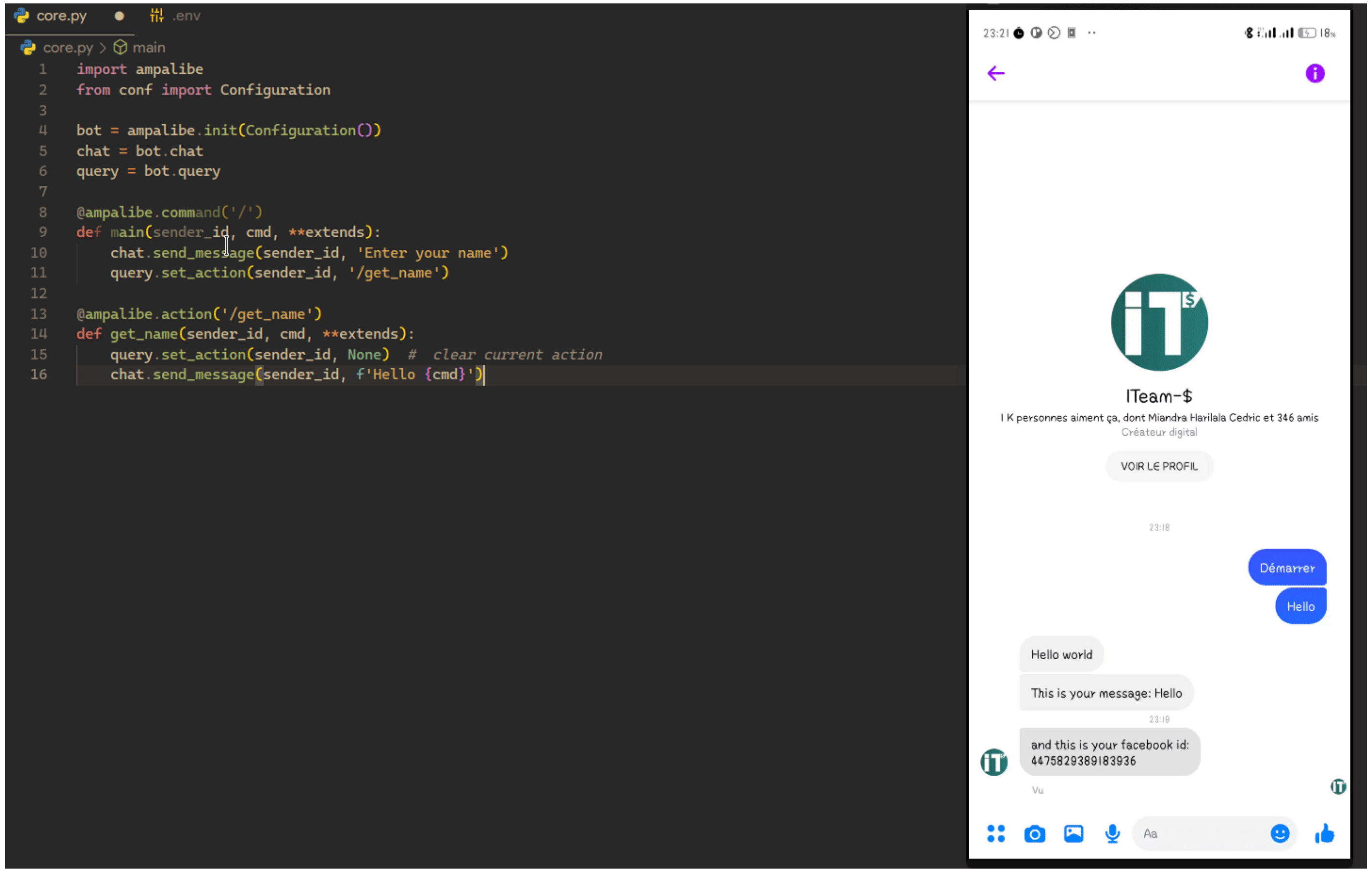
```
import ampalibe
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
   chat.send_text(sender_id, 'Hello world')
   chat.send_text(sender_id, f'This is your message: {cmd}')
   chat.send_text(sender_id, f'and this is your facebook id: {sender_id}')
```

Action Management

Manages the actions expected by the users

define the function of the next treatment

```
import ampalibe
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat
query = bot.query

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    chat.send_text(sender_id, 'Enter your name')
    query.set_action(sender_id, '/get_name')

@ampalibe.action('/get_name')
def get_name(sender_id, cmd, **extends):
    query.set_action(sender_id, None)  #  clear current action
    chat.send_text(sender_id, f'Hello {cmd}')
```



```python
import ampalibe
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat
query = bot.query

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    chat.send_message(sender_id, 'Enter your name')
    query.set_action(sender_id, '/get_name')

@ampalibe.action('/get_name')
def get_name(sender_id, cmd, **extends):
    query.set_action(sender_id, None)  #  clear current action
    chat.send_message(sender_id, f'Hello {cmd}')
```

23:18

Démarrer

Hello

Hello world

This is your message: Hello

23:19

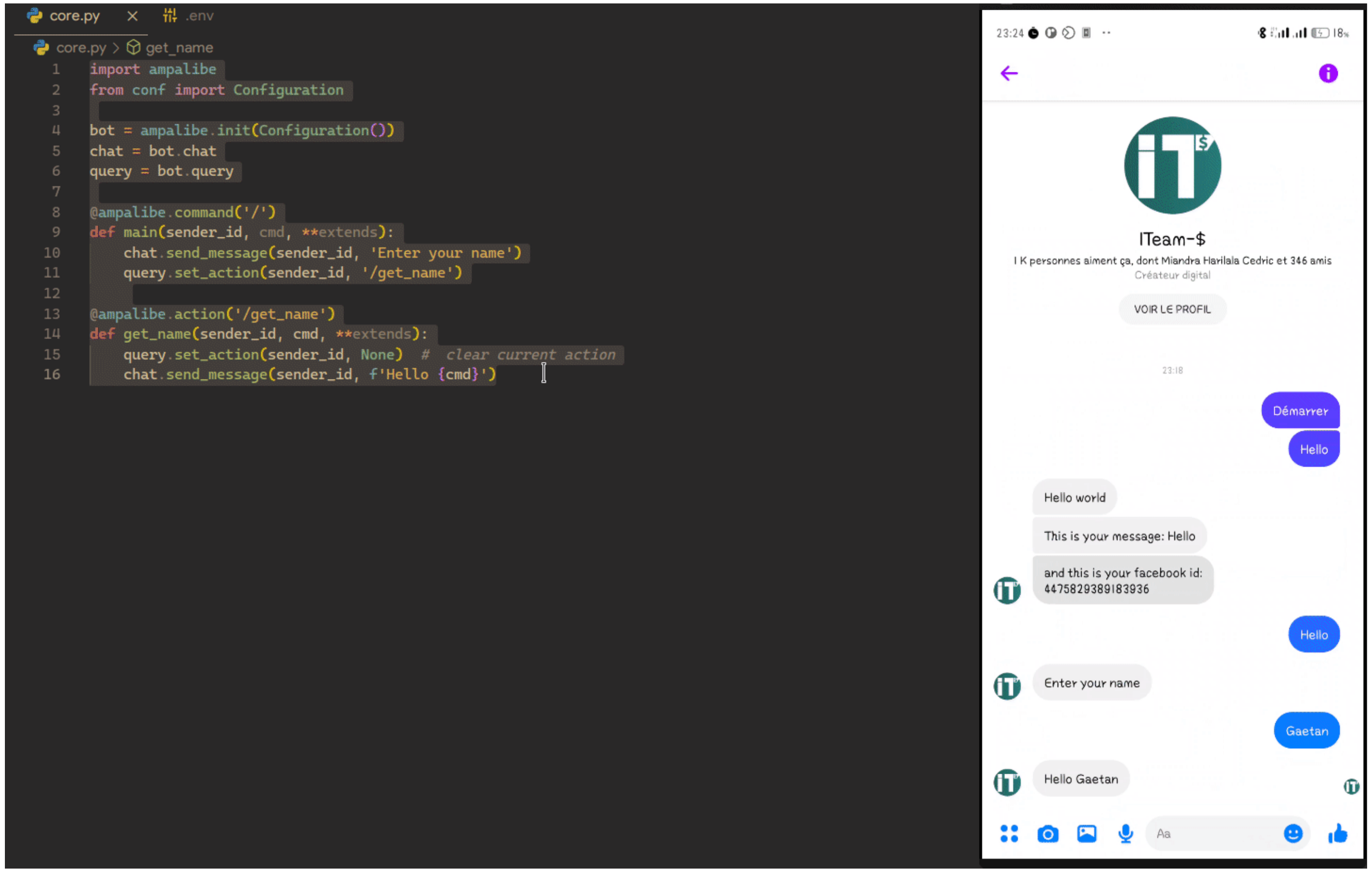
and this is your facebook id: 4475829389183936

Vu

Temporary data Management

Manage temporary data easily with set, get, and delete methods

```python
import ampalibe
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat
query = bot.query

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    chat.send_text(sender_id, 'Enter your mail')
    query.set_action(sender_id, '/get_mail')

@ampalibe.action('/get_mail')
def get_mail(sender_id, cmd, **extends):
    # save the mail in temporary data
    query.set_temp(sender_id, 'mail', cmd)

    chat.send_text(sender_id, f'Enter your password')
    query.set_action(sender_id, '/get_password')


@ampalibe.action('/get_password')
def get_password(sender_id, cmd, **extends):
    query.set_action(sender_id, None)  # clear current action
    mail = query.get_temp(sender_id, 'mail')  # get mail in temporary data

    chat.send_text(sender_id, f'your mail and your password are {mail} {cmd}')
    query.del_temp(sender_id, 'mail')  # delete temporary data
```



```python
import ampalibe
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat
query = bot.query

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    chat.send_message(sender_id, 'Enter your name')
    query.set_action(sender_id, '/get_name')

@ampalibe.action('/get_name')
def get_name(sender_id, cmd, **extends):
    query.set_action(sender_id, None)  #  clear current action
    chat.send_message(sender_id, f'Hello {cmd}')
```

Démarrer
Hello
Hello world
This is your message: Hello
and this is your facebook id: 4475829389183936
Hello
Enter your name
Gaetan
Hello Gaetan

Payload Management

Manage Payload easily

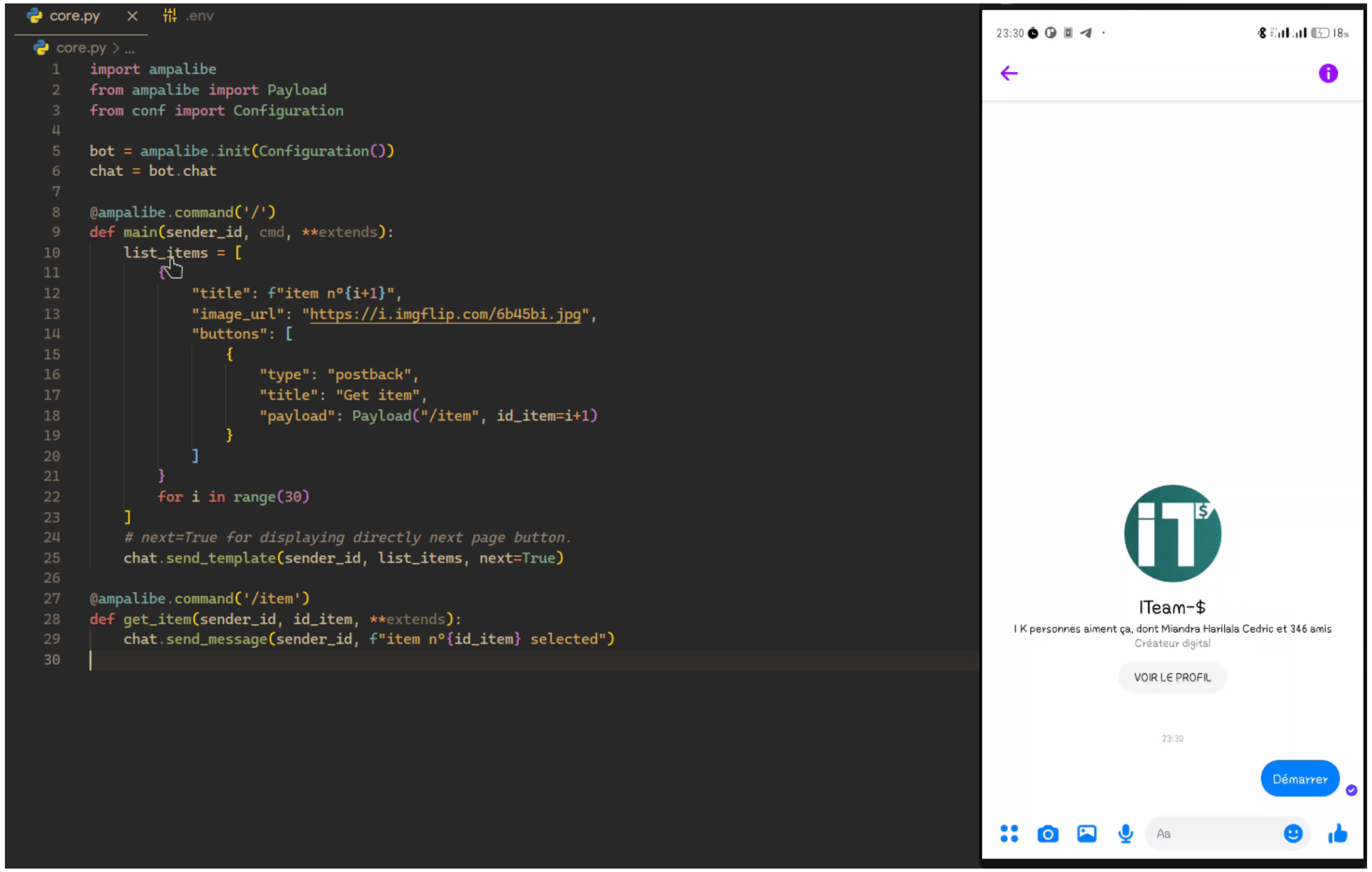
send data with Payload object and get it in destination function's parameter

```python
import ampalibe
from ampalibe import Payload
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat


@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    quick_rep = [
        {
            "content_type": "text",
            "title": 'Angela',
            "payload": Payload('/membre', name='Angela', ref='2016-sac')
        },
        {
            "content_type": "text",
            "title": 'Rivo',
            "payload": Payload('/membre', name='Rivo')
        }
    ]
    chat.send_quick_reply(sender_id, quick_rep, 'Who?')


@ampalibe.command('/membre')
def get_membre(sender_id, cmd, name, **extends):
    chat.send_text(sender_id, "Hello " + name)

    # if the arg is not defined in the list of parameters,
    # it is put in the extends variable
    if extends.get('ref'):
        chat.send_text(sender_id, 'your ref is ' + extends.get('ref'))
```

Advanced Messenger API

No need to manage the length of the items to send: A next page button will be displayed directly

```python
import ampalibe
from ampalibe import Payload
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    list_items = [
        {
            "title": f"item n°{i+1}",
            "image_url": "https://i.imgflip.com/6b45bi.jpg",
            "buttons": [
                {
                    "type": "postback",
                    "title": "Get item",
                    "payload": Payload("/item", id_item=i+1)
                }
            ]
        }
        for i in range(30)
    ]
    # next=True for displaying directly next page button.
    chat.send_template(sender_id, list_items, next=True)

@ampalibe.command('/item')
def get_item(sender_id, id_item, **extends):
    chat.send_text(sender_id, f"item n°{id_item} selected")
```



```python
import ampalibe
from ampalibe import Payload
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat

@ampalibe.command('/')
def main(sender_id, cmd, **extends):
    list_items = [
        {
            "title": f"item n°{i+1}",
            "image_url": "https://i.imgflip.com/6b45bi.jpg",
            "buttons": [
                {
                    "type": "postback",
                    "title": "Get item",
                    "payload": Payload("/item", id_item=i+1)
                }
            ]
        }
        for i in range(30)
    ]
    # next=True for displaying directly next page button.
    chat.send_template(sender_id, list_items, next=True)

@ampalibe.command('/item')
def get_item(sender_id, id_item, **extends):
    chat.send_message(sender_id, f"item n°{id_item} selected")
```

Langage Management

Language management is directly managed by Ampalibe

langs.json

```
{
    "hello_world": {
        "en": "Hello World",
        "fr": "Bonjour le monde"
    },

    "ampalibe": {
        "en": "Jackfruit",
        "fr": "Jacquier",
        "mg": "Ampalibe"
    }
}
```

core.py

```python
import ampalibe
from ampalibe import translate
from conf import Configuration

bot = ampalibe.init(Configuration())
chat = bot.chat
query = bot.query

@ampalibe.command('/')
def main(sender_id, lang, cmd, **extends):
    chat.send_text(
        sender_id,
        translate('hello_world', lang)
    )
    query.set_lang(sender_id, 'en')
    query.set_action(sender_id, '/what_my_lang')


@ampalibe.action('/what_my_lang')
def other_func(sender_id, lang, cmd, **extends):
    query.set_action(sender_id, None)

    chat.send_text(sender_id, 'Your lang is ' + lang + ' now')
    chat.send_text(
        sender_id,
        translate('hello_world', lang)
    )
```