**2024/01/16 00:00|Лекция. Хендлеры обработки сообщений**

## Хендлеры обработки сообщений

### Как происходит обработка сообщений в чат ботах, что это такое и зачем нужно?

Обработчики сообщений в чат ботах называются хендлерами. Если зайти в диспетчер, увидим, что бывает «**message\_handlers**» - это **декоратор**, который позволяет работать с сообщениями. Эта функция асинхронна. (Рис. 1)

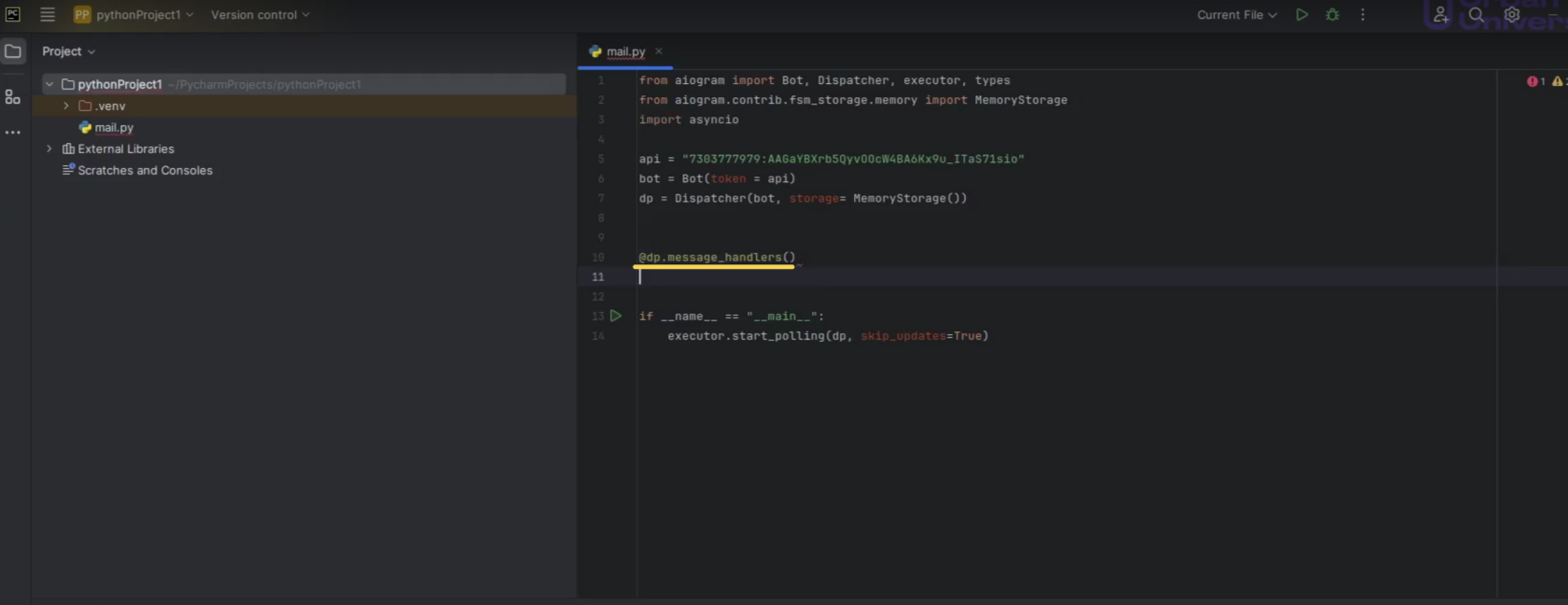


Рис. 1

Сделаем функцию, которая будет обрабатывать все сообщения «**all\_message**». Она будет в себя получать сообщение. Как только мы получим сообщение, мы будем об этом узнавать через, к примеру, «**print**». (Рис. 2)

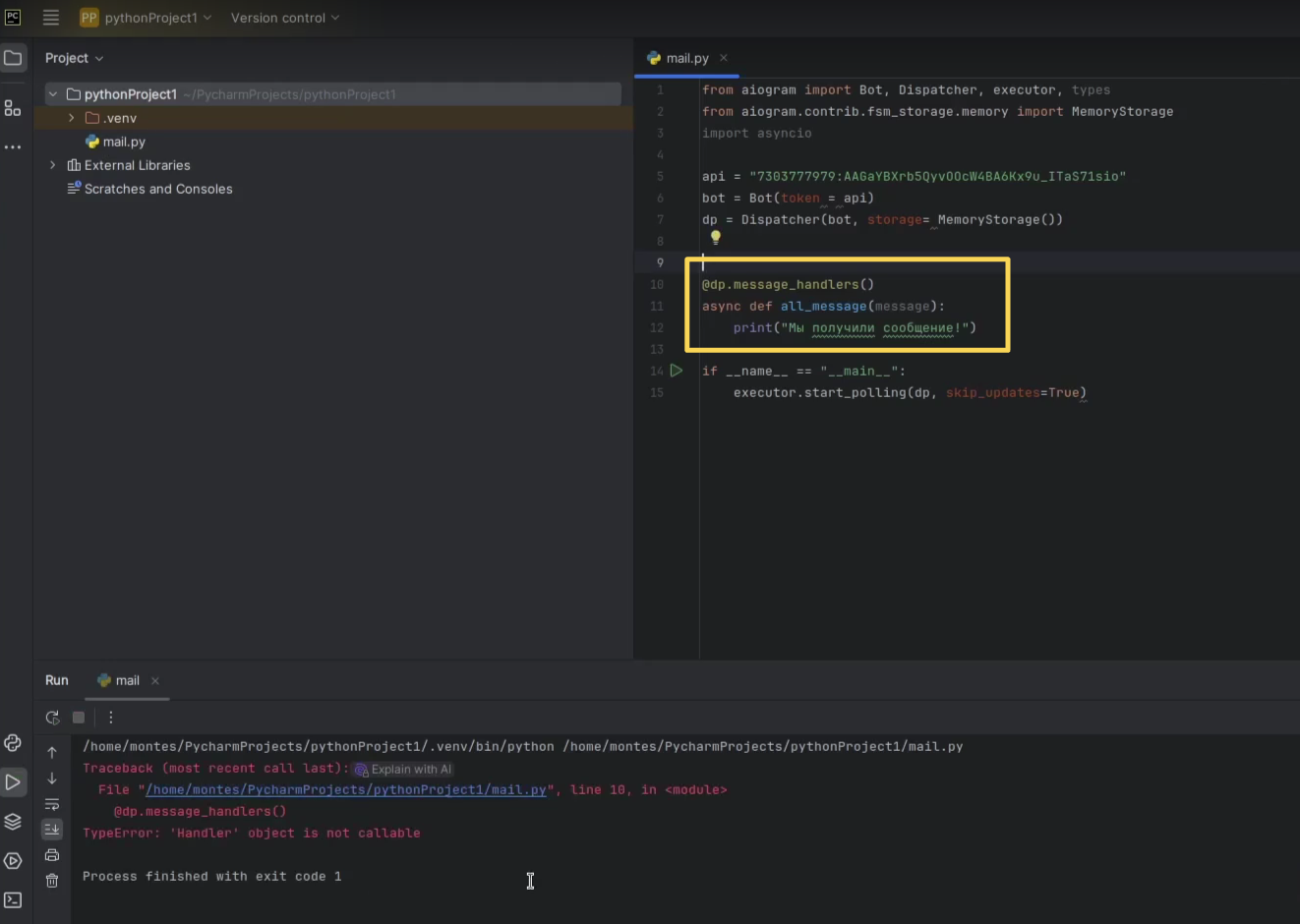


Рис. 2

Видим, что «**Handler is not callable**». Это опечатка. Правильно «**message\_handler**» вместо «**message\_handlers**». Часто здесь бывают ошибки. Исправим. (Рис. 3)

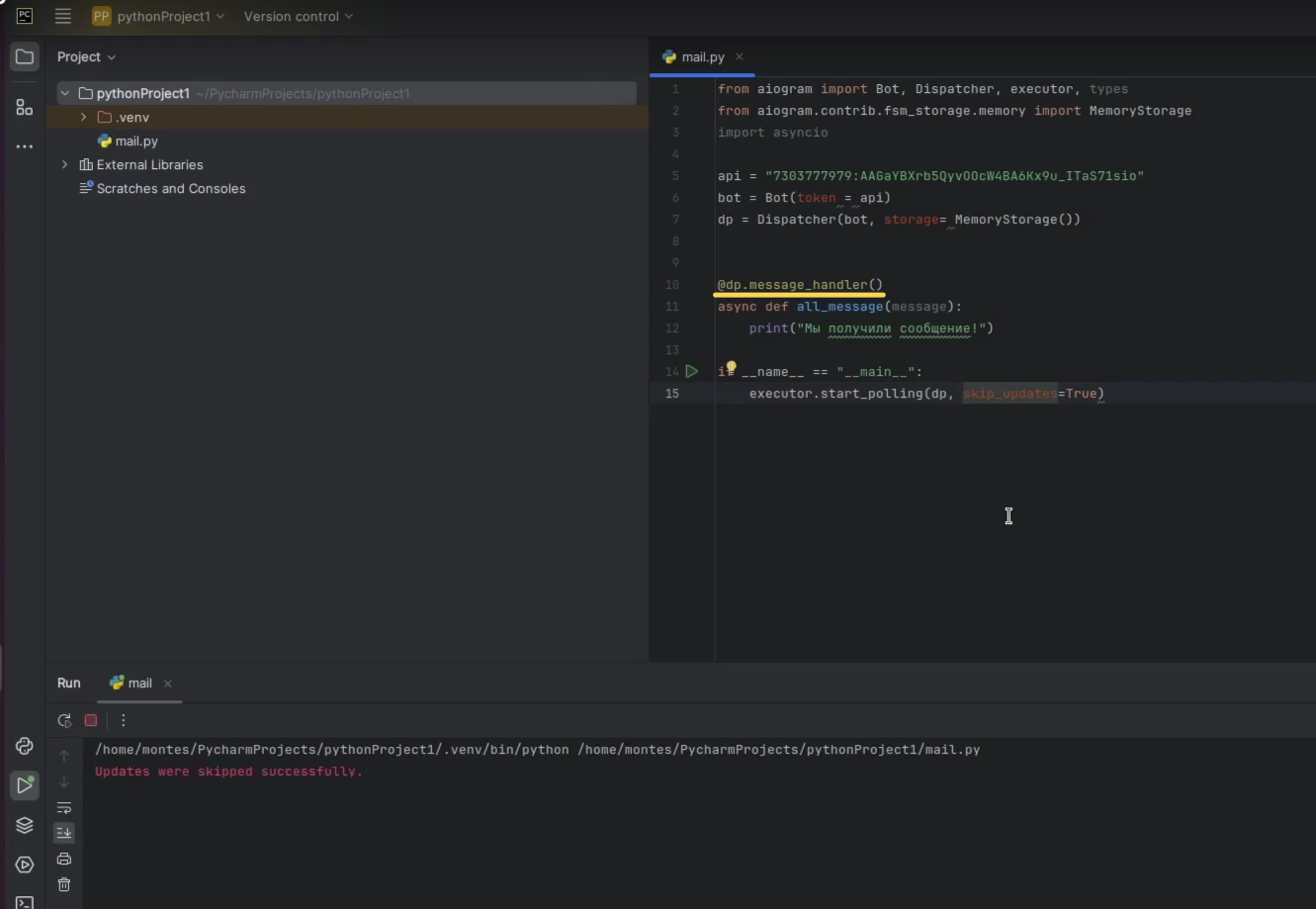


Рис. 3

Запускаем, заходим в бот и напишем «привет». «Мы получили сообщение». (Рис. 4)

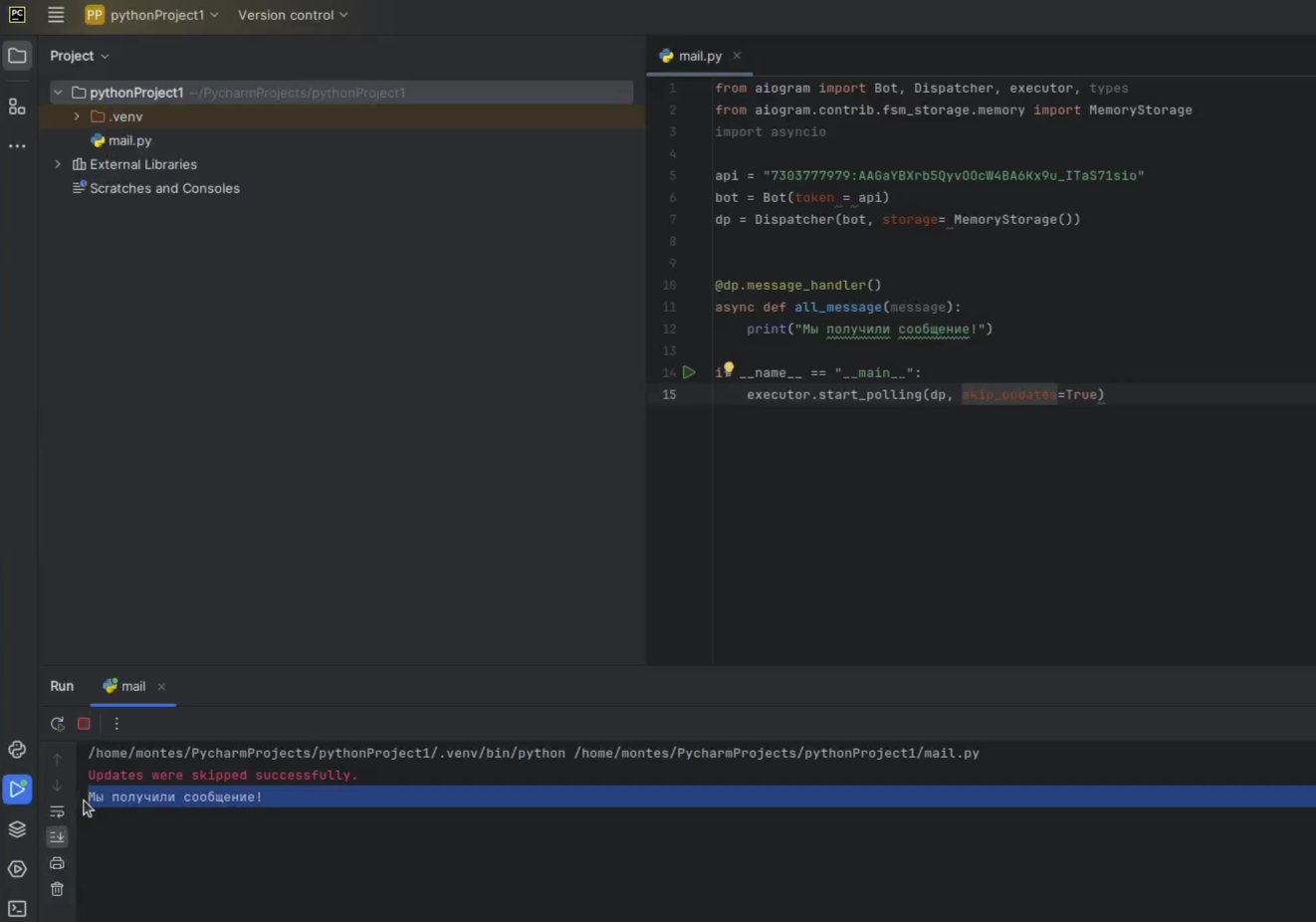


Рис. 4

Сделаем еще один «handler» и в него напишем текст «**Urban**». (Рис. 5)

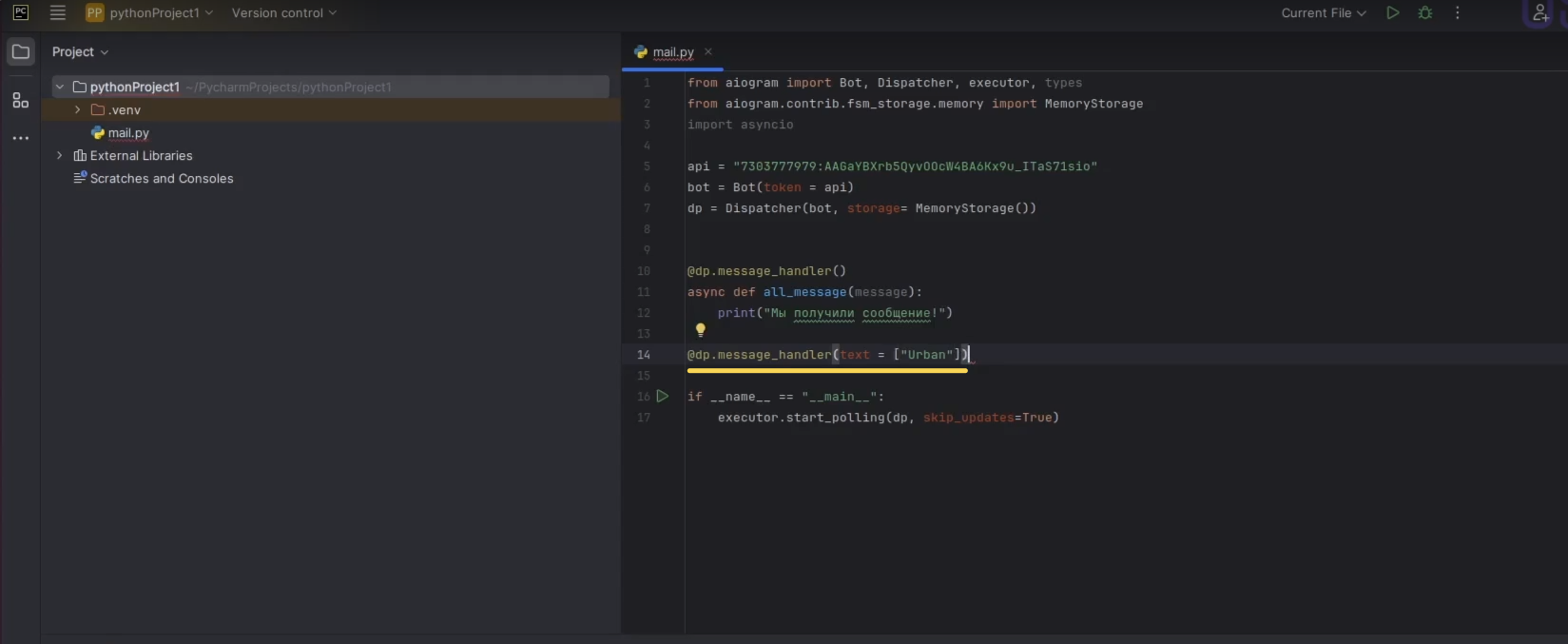


Рис. 5

Напишем «**async**», функцию назовем «**urban\_massage**». В качестве передаваемого параметра будет сообщение «urban\_massage». (Рис. 6)

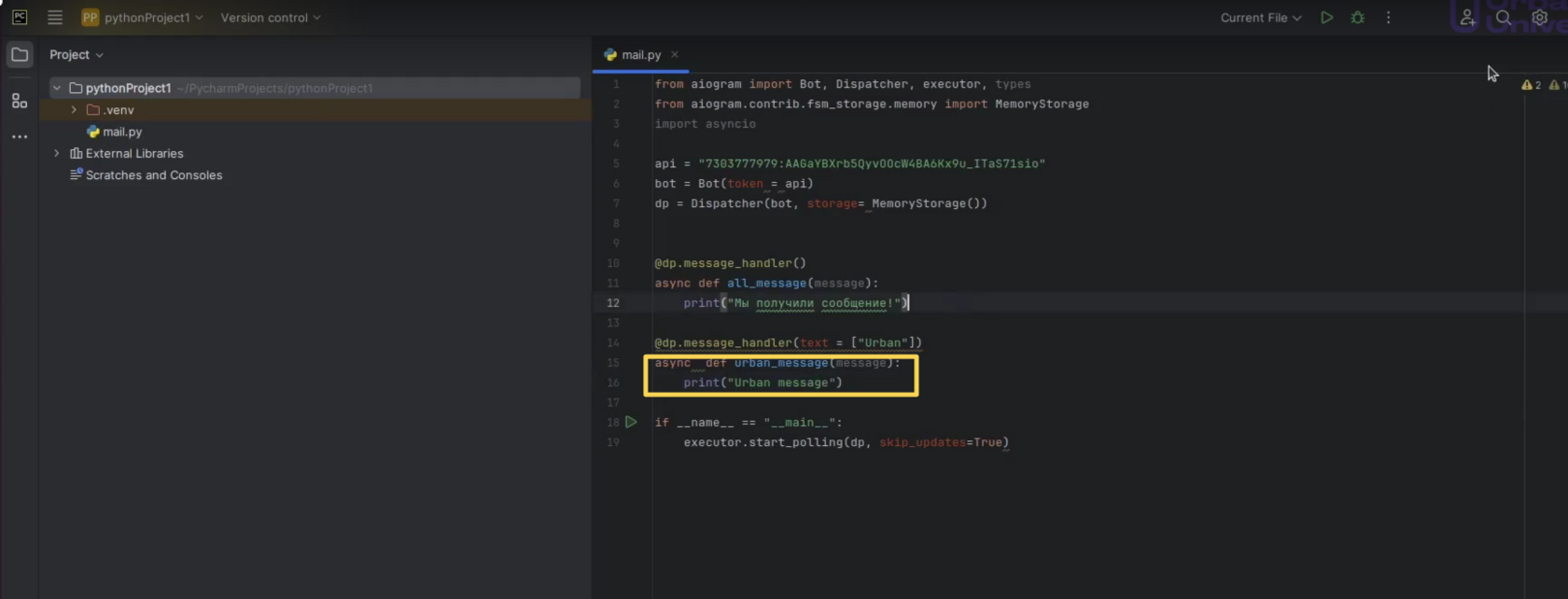


Рис. 6

Запустим. Напишем что-нибудь и увидим, что получили сообщение. (Рис. 7, 8)

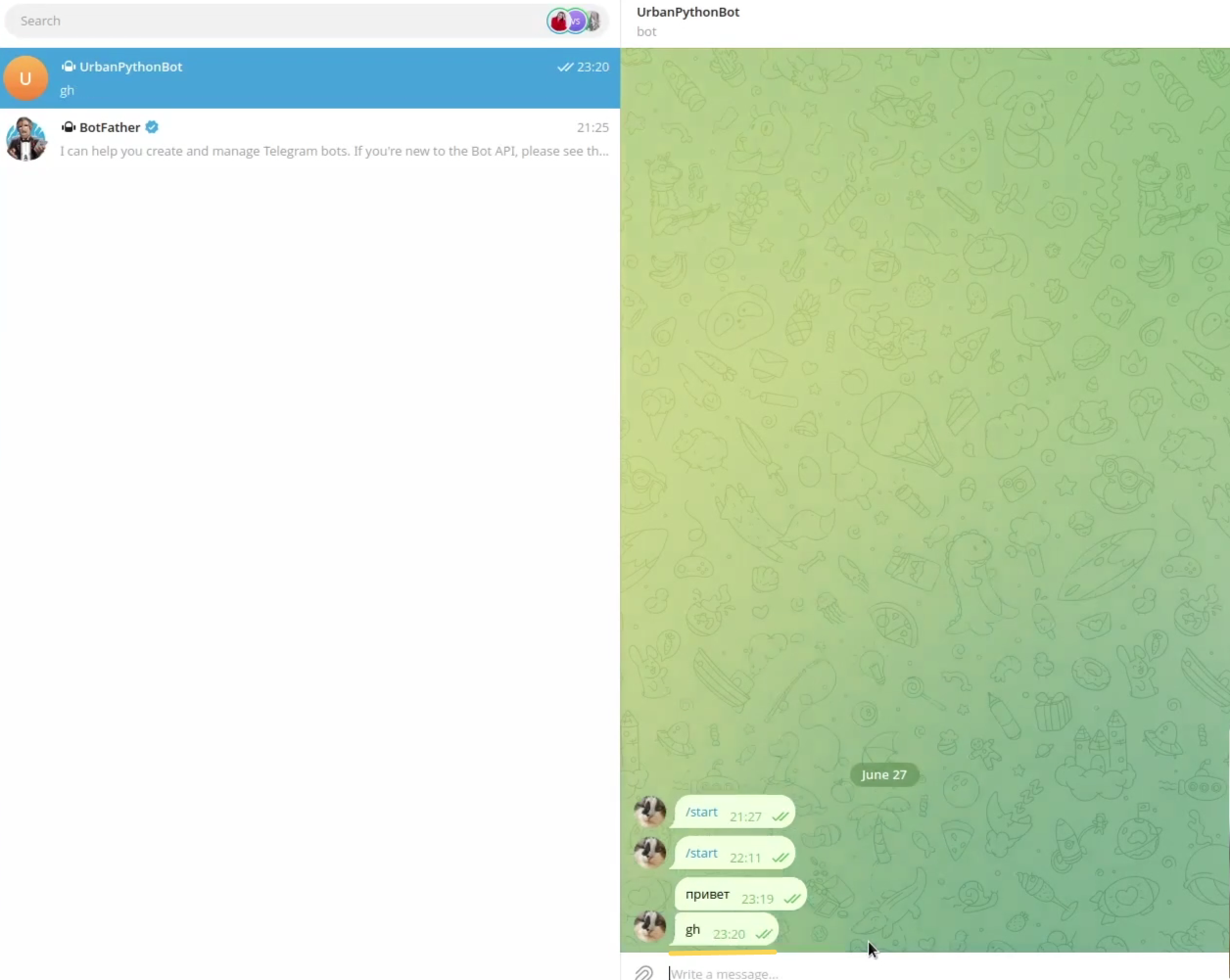


Рис. 7

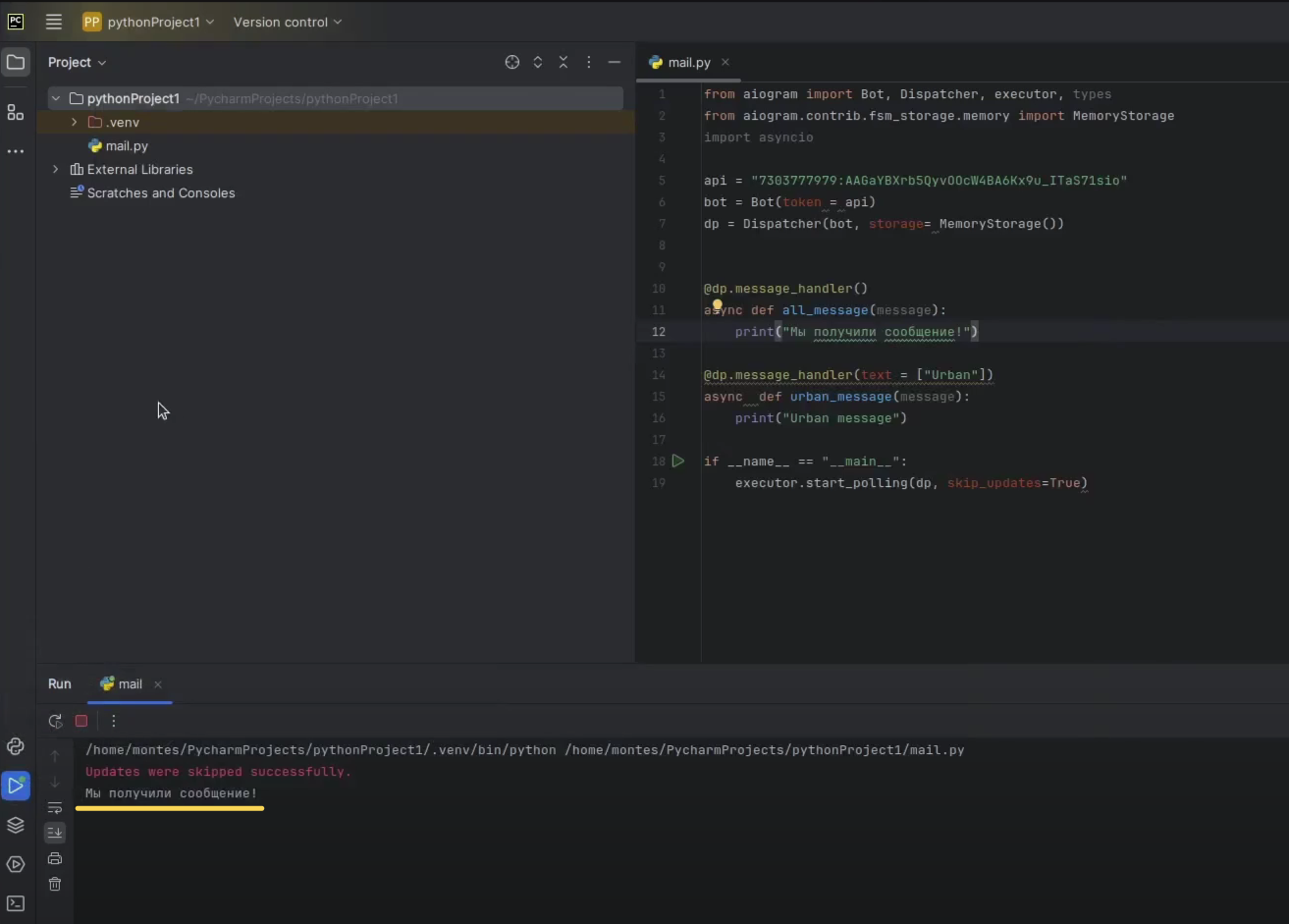


Рис. 8

Запустим «urban» и видим, что ничего не случилось. (Рис. 9, 10)

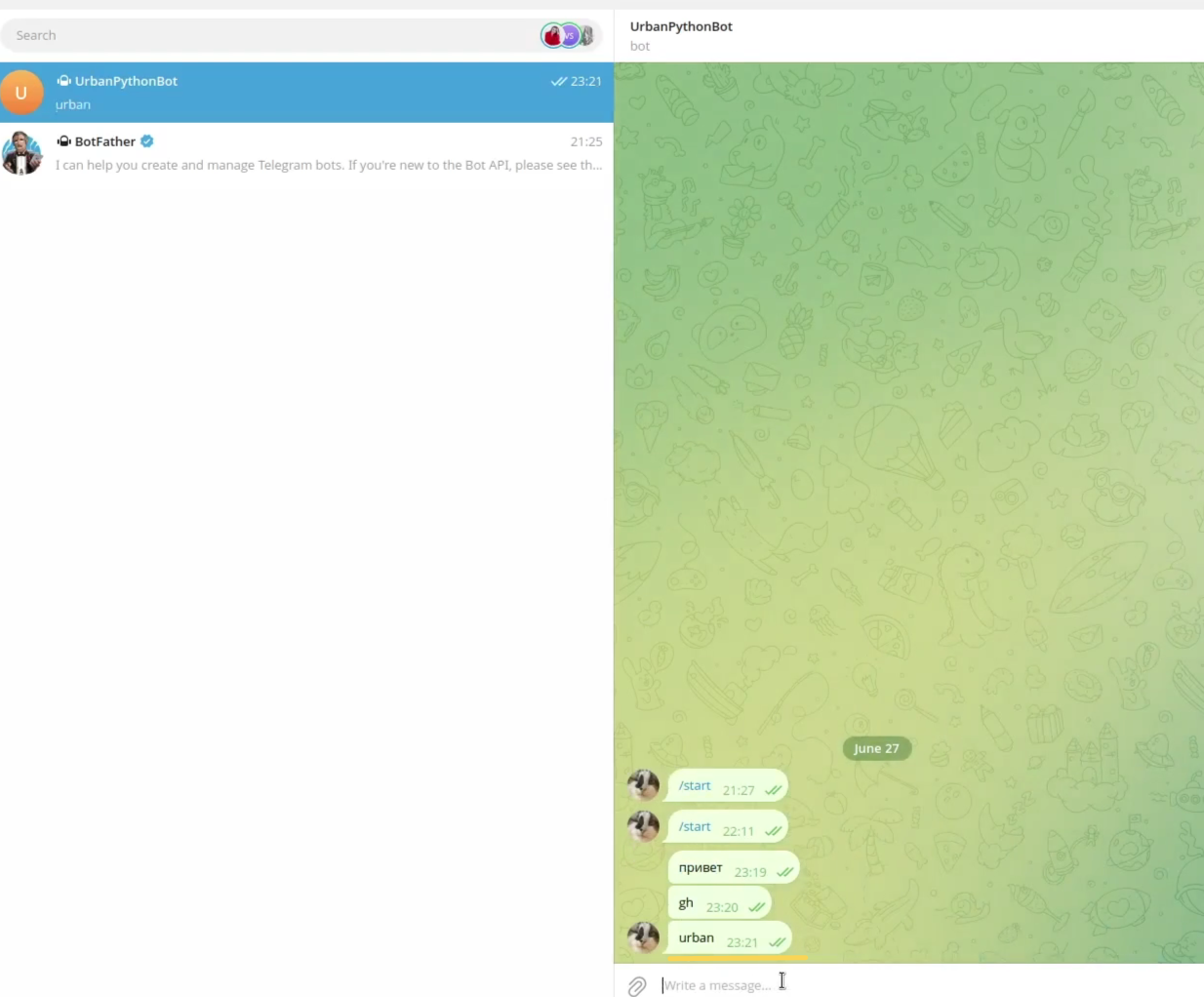


Рис. 9

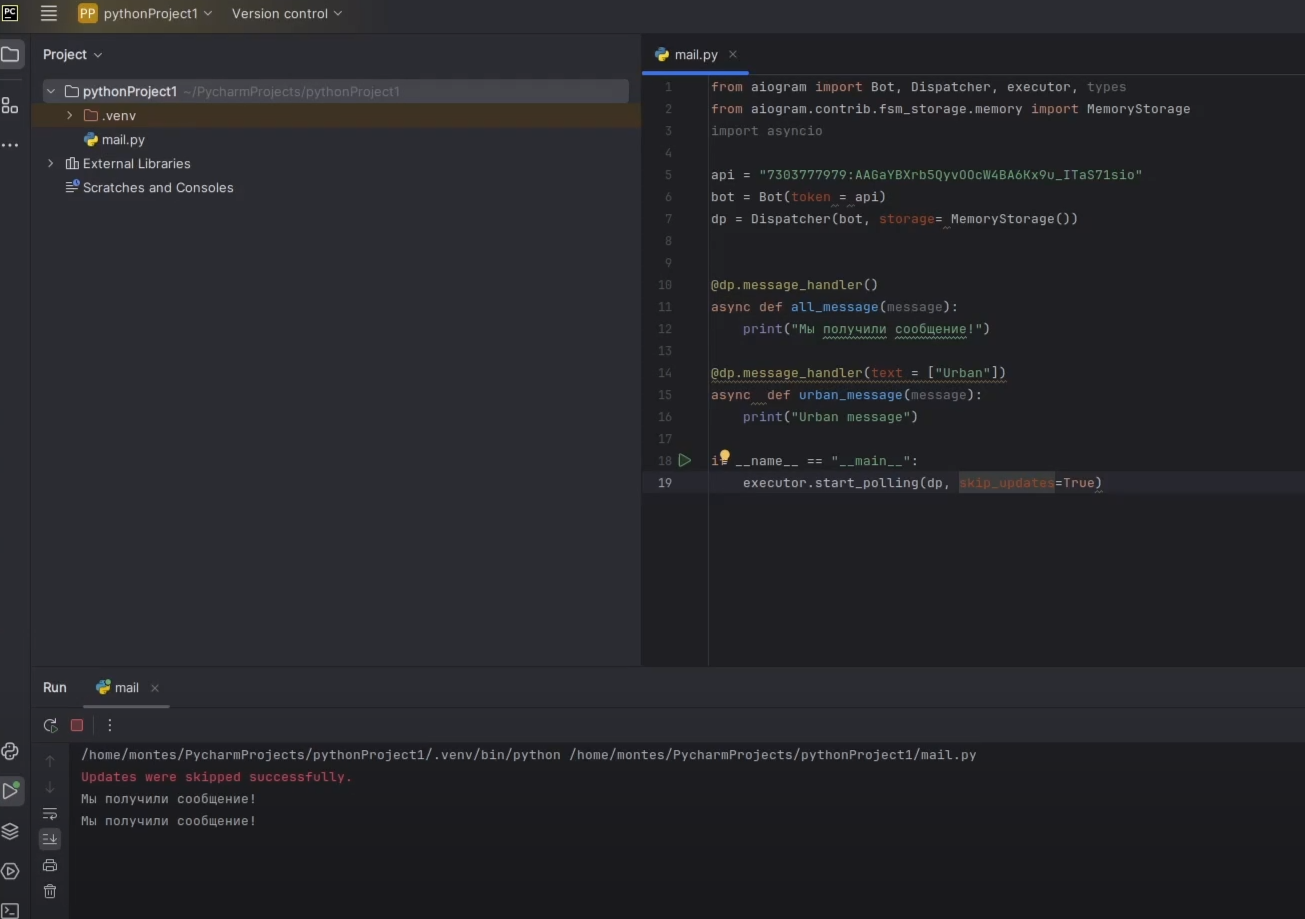


Рис. 10

Напишем «Urban» с большой буквы, но и не в этом дело. (Рис. 11)

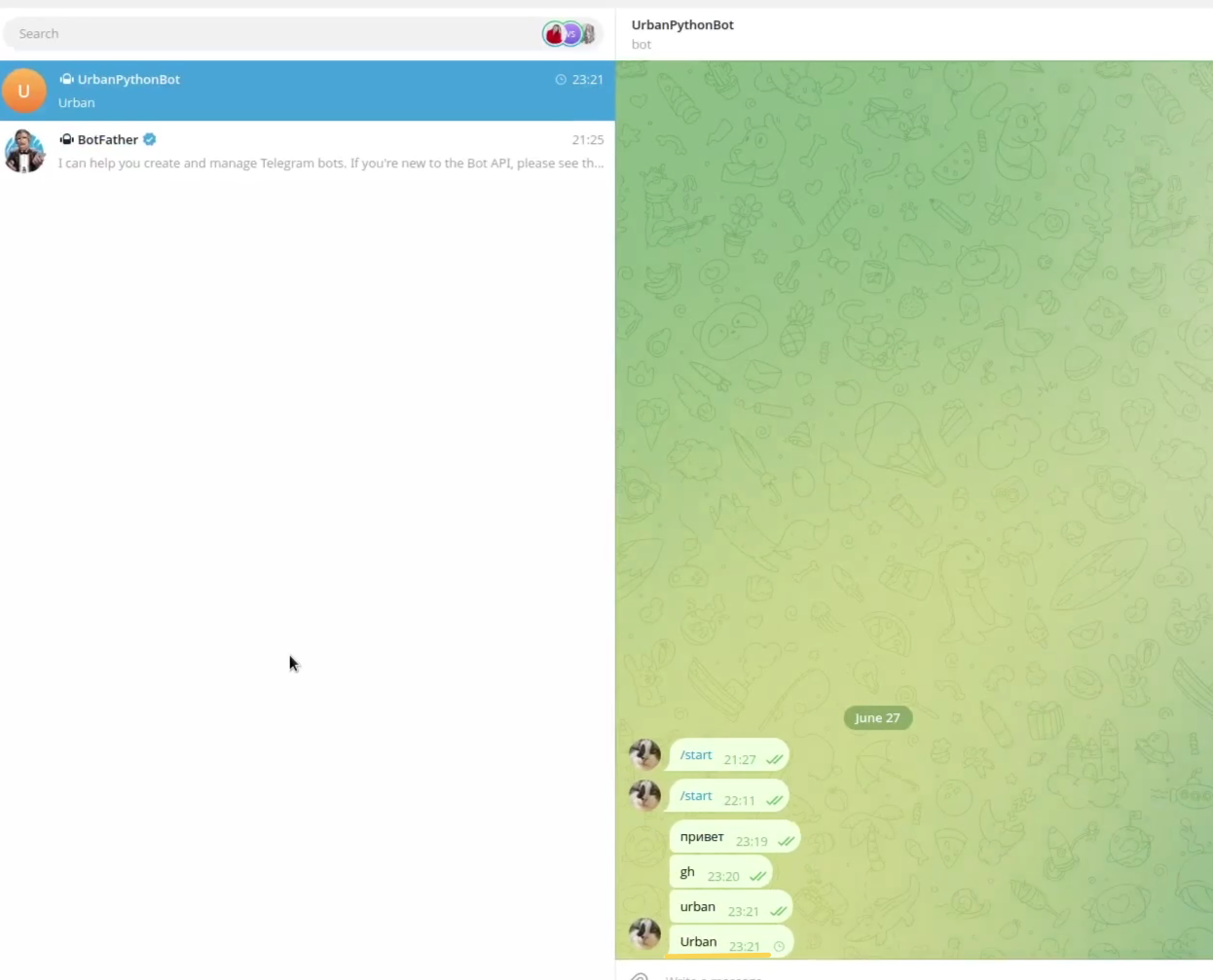


Рис. 11

Сейчас «message\_handler» отрабатывает сообщение по тексту и есть некий вариант его обработки. Добавим пару сообщений чтобы было удобней. Запустим, напишем сообщение и снова ничего не происходит. (Рис. 12)

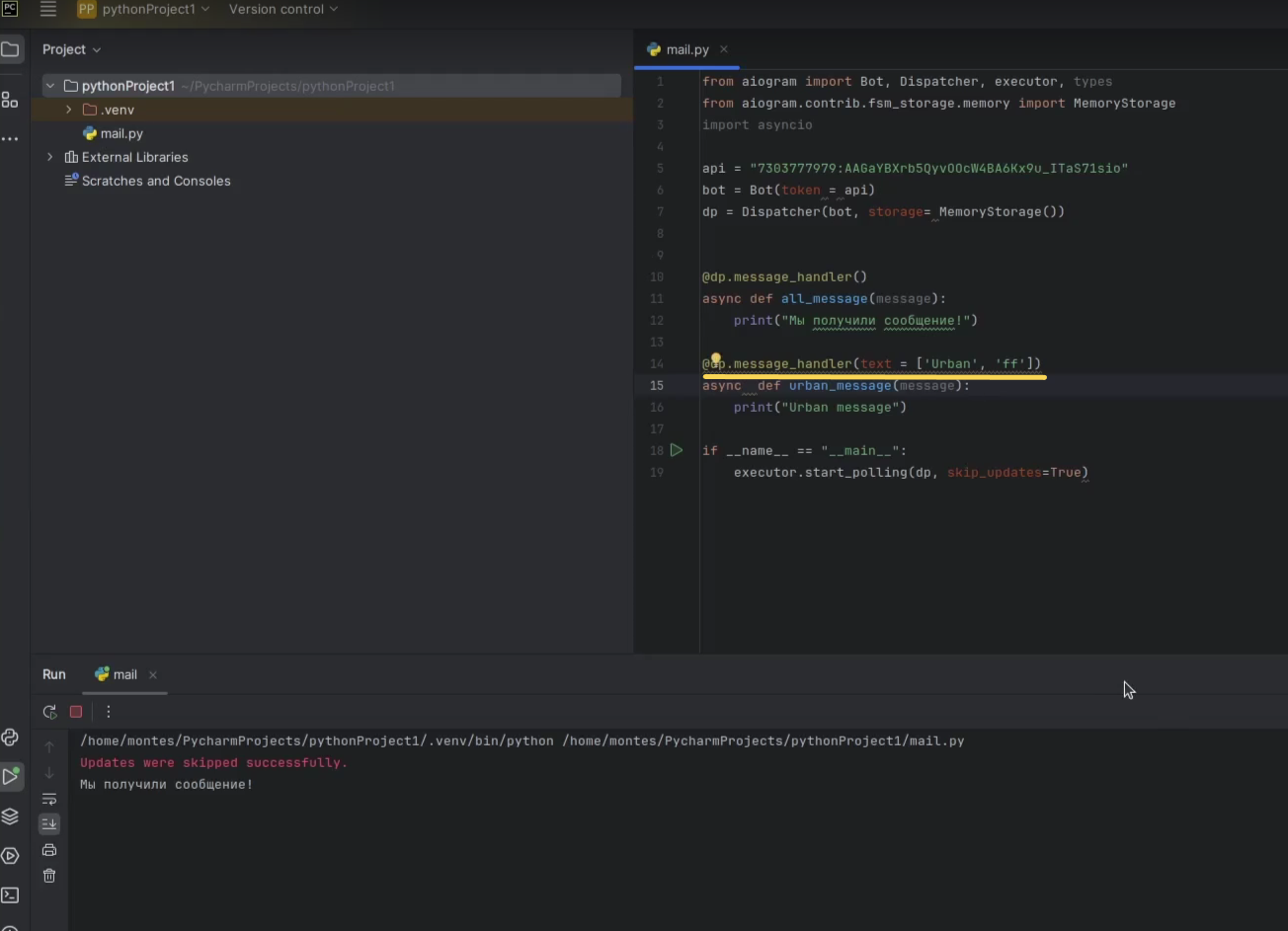


Рис. 12

**Проблема** в том, что отработал сначала более **общий «handler»**. Если мы не хотим, чтобы он перехватывал сообщение, его нужно «отбирать». Хендлеры зависят друг от друга, т.е. только 1 хендлер может отработать 1 сообщение. (Рис. 13)

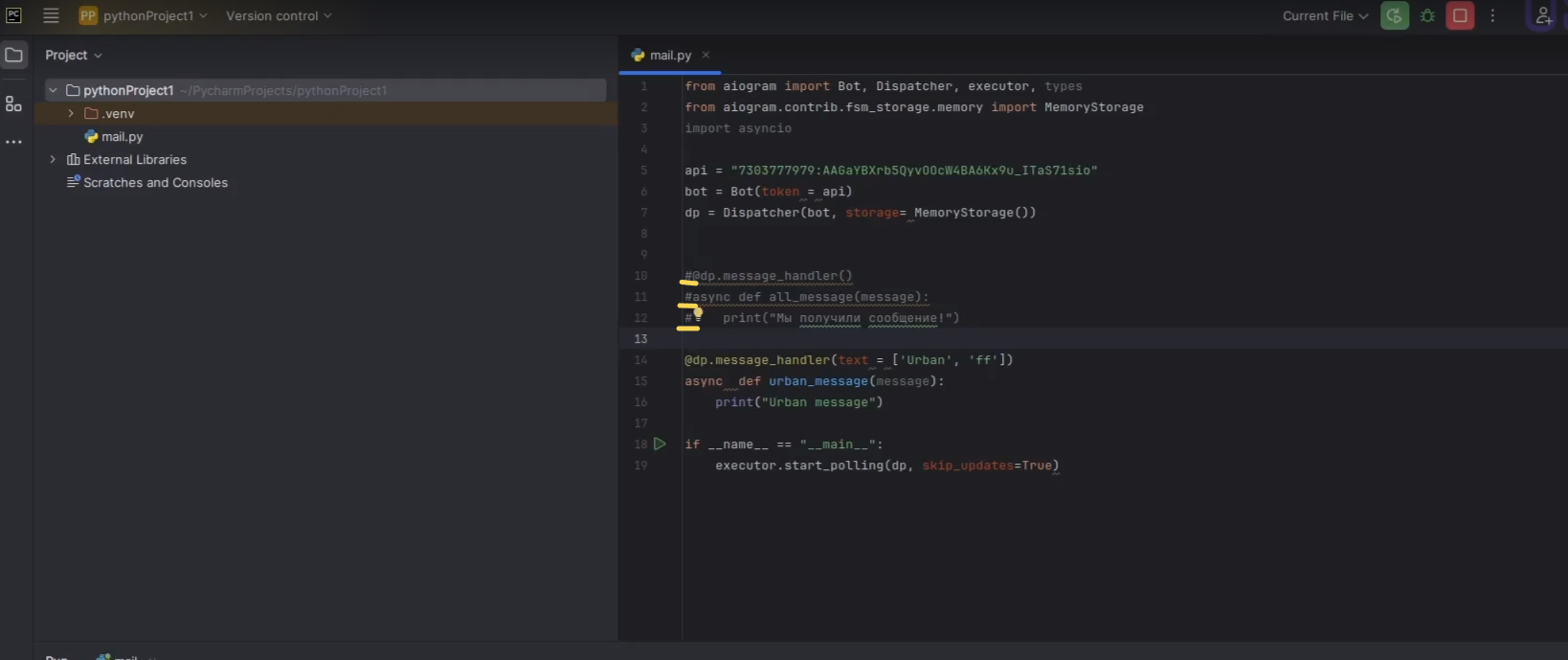


Рис. 13

Напишем в бот «ff» и видим «Urban message». (Рис. 14)

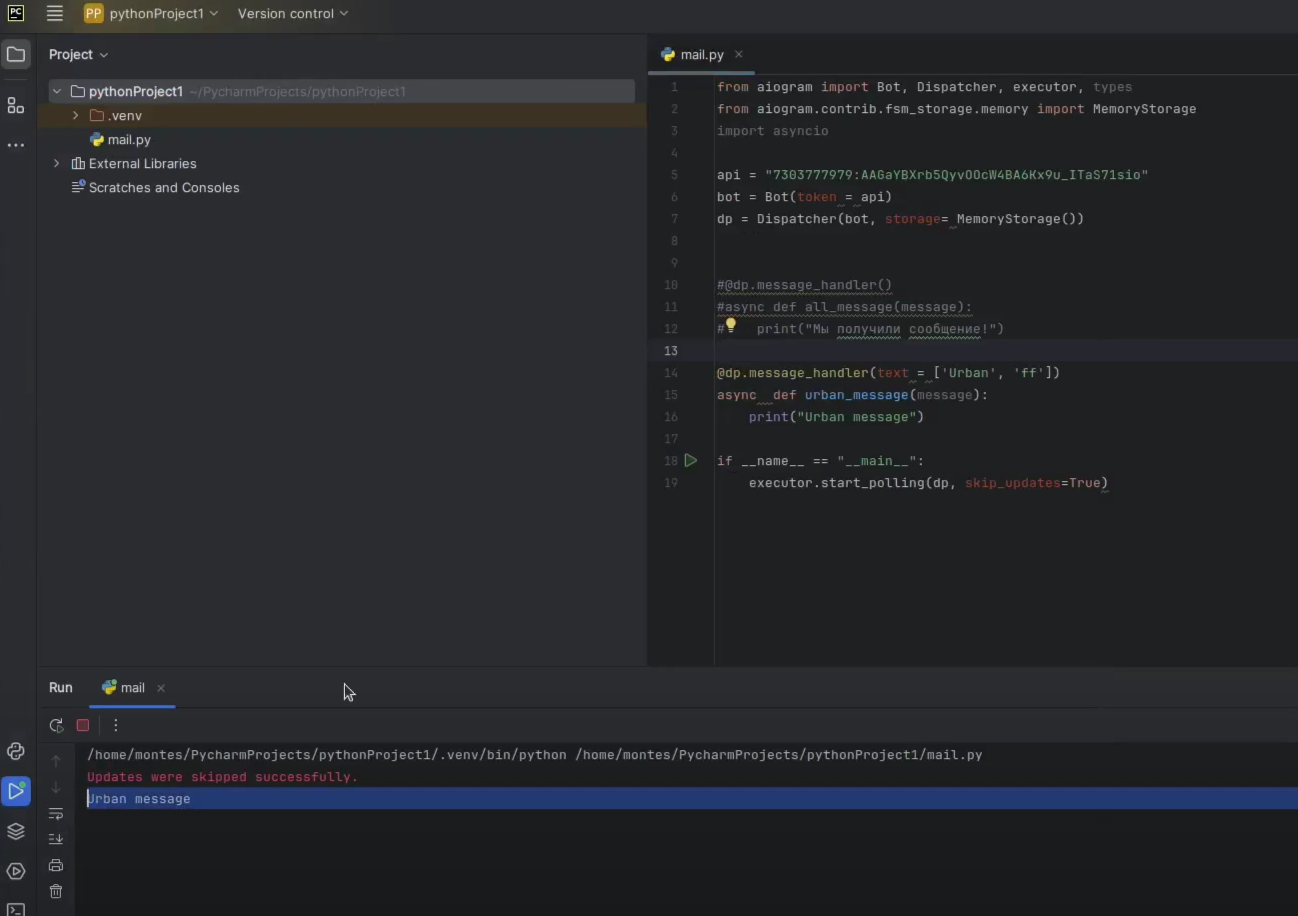


Рис. 14

Это произошло, потому что первый хендлер перехватывает все сообщения, а второй только сообщения специфического вида.

Можно сделать систему, которая позволит нам отрабатывать некие команды. Сделаем еще один диспетчер «message\_handler». Здесь будет параметр «**commands**», укажем команду «**start**». Сделаем функцию «**async def**», будем печатать «Start message». (Рис. 15)

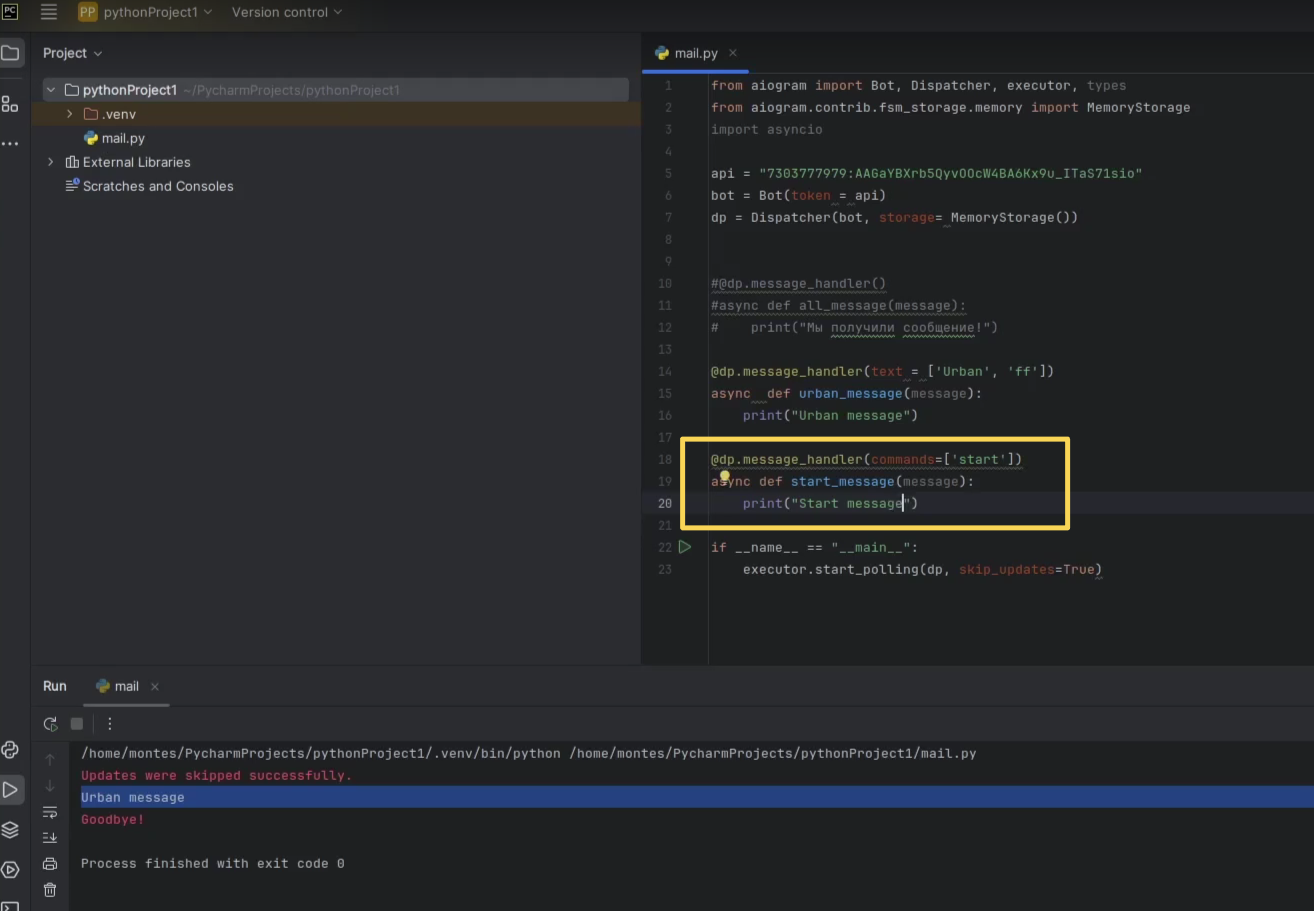


Рис. 15

Снова запустим. Если пишем «ff», то у нас «Urban message», если «/start», т.е. команду «start», мы получаем «Start message». (Рис. 16)

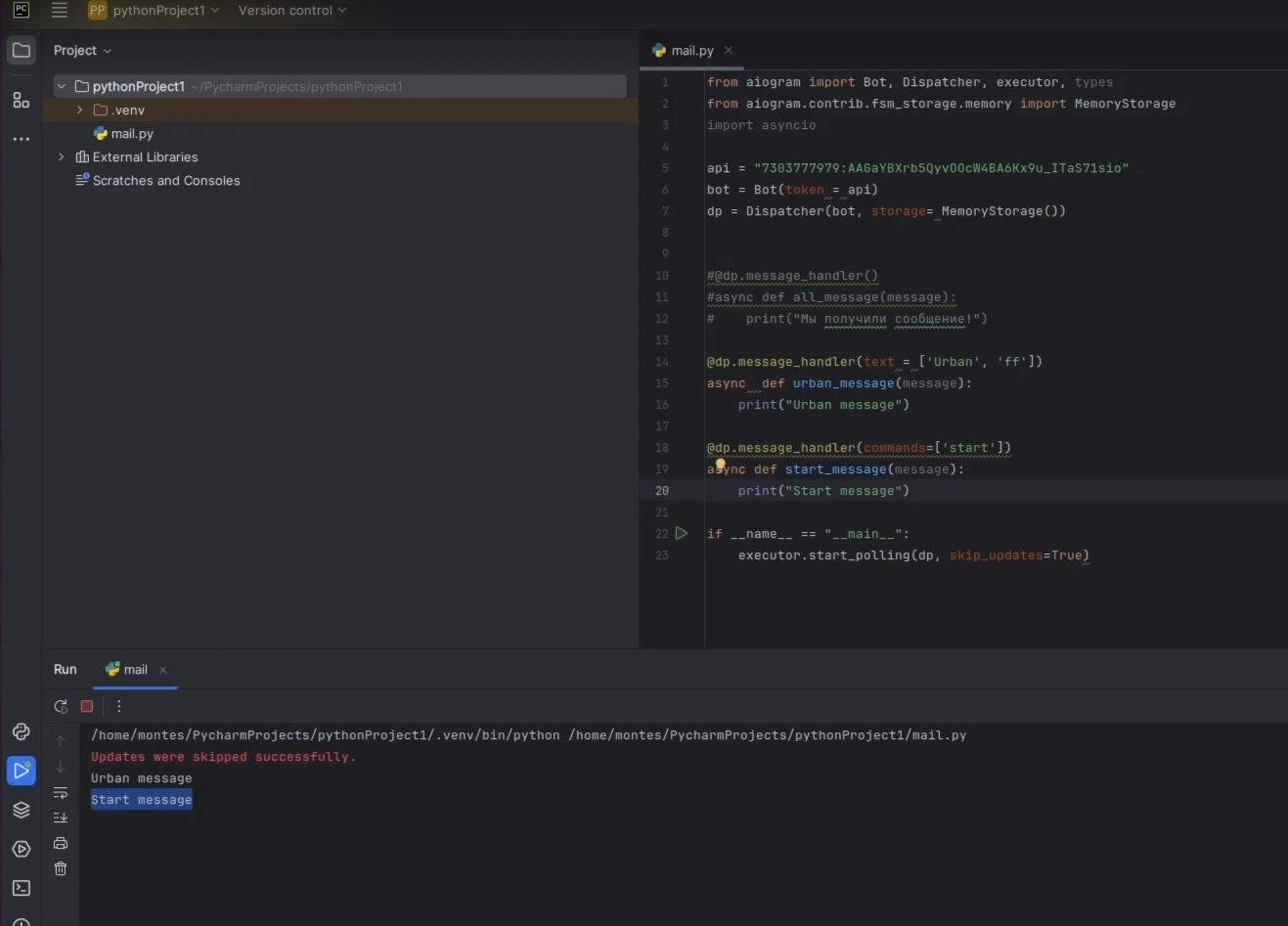


Рис. 16

Расположение хендлеров влияет. Пишем «ff», отработает «Urban message». (Рис. 17)

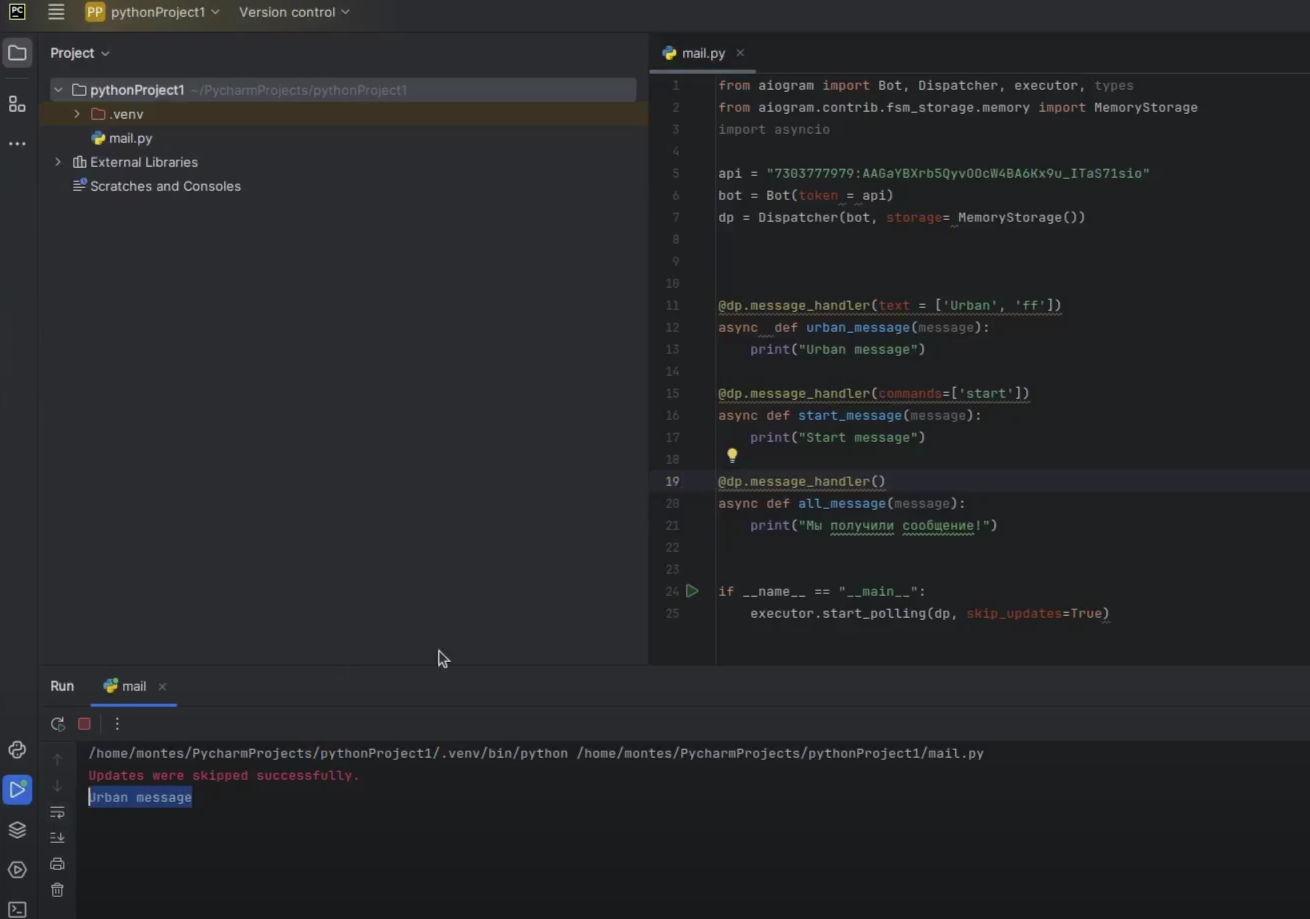


Рис. 17

### По какому принципу работает «Urban message»?

Мы находимся в Phyton, и здесь функции отрабатывают сверху вниз. Если мы делаем какой-то **специфичный хендлер**, мы его помещаем **в начало**, а все**общие** **хендлеры** (реагировать на команду, сообщение) ставим**в конце**. Если общий хендлер поставить в начало, то он будет брать все сообщения и специфичные хендлеры будут пропускаться. В обратном случае получаем ситуацию, когда все необходимые хендлеры отрабатывают, если ничего не попало под первые, то отрабатывает последний хендлер.

**«message\_handler» имеет 2 параметра – «text», и «commands».** «**text**» - это то сообщение, на которое стоит реагировать, «**commands**» - сообщение после «/» на которое стоит реагировать. Если оставить поле пустым, то оно будет реагировать на все вокруг себя.