

Пояснительная записка к проекту MCStudio

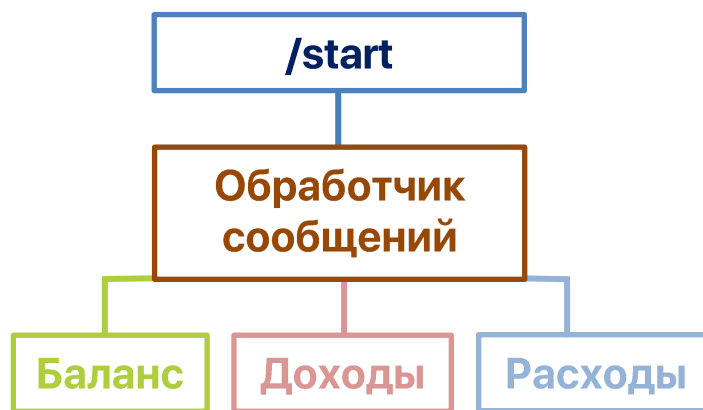
Проект **MCStudio** – это Telegram-бот, написанный на Python с помощью PyTelegramBotAPI. Исходный код программы опубликован на Github по адресу <https://github.com/deltaproduction/mcstudiobot>. Также бот использует сервер Pythonanywhere.

1. Создание бота

Создание бота происходило с помощью встроенного бота Telegram **@BotFather**. Выбор названия, описания, картинки происходил на этом этапе.

2. Разработка

У бота есть обработчик сообщений. Когда пользователь получает сообщение, оно проходит через него и в дальнейшем он распределяет действия.



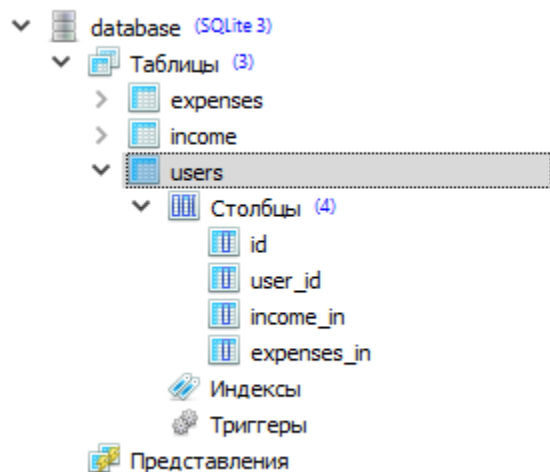
При запуске бота (при отправке **/start**) пользователь видит следующее сообщение:

Это бот **MCStudio**
С его помощью вы можете:

- 💰 Вести учёт своих денежных средств
- ✓ Проверять баланс
- ✚ Добавлять свои доходы
- ✖ Добавлять свои расходы
- 📱 И все это вы можете делать где угодно!

Пользователю предоставляется возможность зарегистрироваться. Данные о пользователе хранятся в базе данных в таблице **users**.



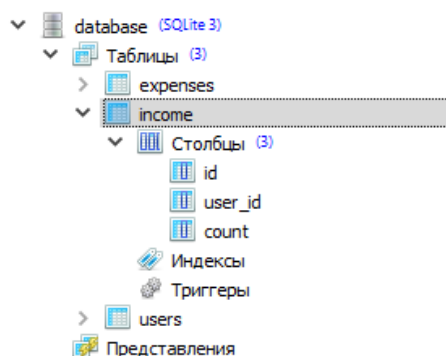


	id	user_id	income_in	expenses_in
1	20	1266347268	0	0

Таблица содержит все данные о пользователях:

- **id** – идентификатор пользователя в системе,
- **user_id** – идентификатор пользователя в Telegram, необходимый для того, чтобы различать пользователей,
- **income_in** – флаг, необходимый для того, чтобы принимать значения доходов пользователя,
- **expenses_in** – флаг, необходимый для того, чтобы принимать значения расходов пользователя.

В таблицах **income** и **expenses** хранятся данные обо всех доходах и расходах пользователей.



	id	user_id	count
1	258	1266347268	1000
2	259	1266347268	500
3	260	1266347268	100
4	261	1266347268	100
5	262	1266347268	50
6	263	1266347268	100
7	264	1266347268	50
8	265	1266347268	100
9	266	1266347268	50
10	267	1266347268	100
11	268	1266347268	10
12	269	1266347268	10
13	270	1266347268	10
14	271	1266347268	10
15	272	1266347268	10

Поля у таблиц **income** и **expenses** одинаковы:

- **id** – идентификатор данных,
- **user_id** – идентификатор пользователя, которому принадлежит информация о доходах или расходах,
- **count** – количество денег, обозначающих информацию о доходах или расходах.

Работа с базой данных осуществляется с помощью **db.py**. Этот файл содержит в себе методы для работы с базой данных. Эти методы используются в **main.py**.

Одним из таких методов является метод **create_user**, вызываемый для регистрации пользователя.

После нажатия на кнопку «Зарегистрироваться» вызывается метод **create_user** класса **Database**. Он принимает всего один аргумент – идентификатор пользователя в Telegram. Столбец, в который записывается идентификатор пользователя, называется **user_id**. Далее, после успешной регистрации пользователя и добавления его в базу данных пользователю предлагается выбрать одно из трех действий. При нажатии на кнопку клавиатуры в чат от лица пользователя отправляется сообщение с содержимым кнопки. Обработчик сообщений обрабатывает такой запрос, и в соответствии с его содержимым предоставляет пользователю следующие возможности:

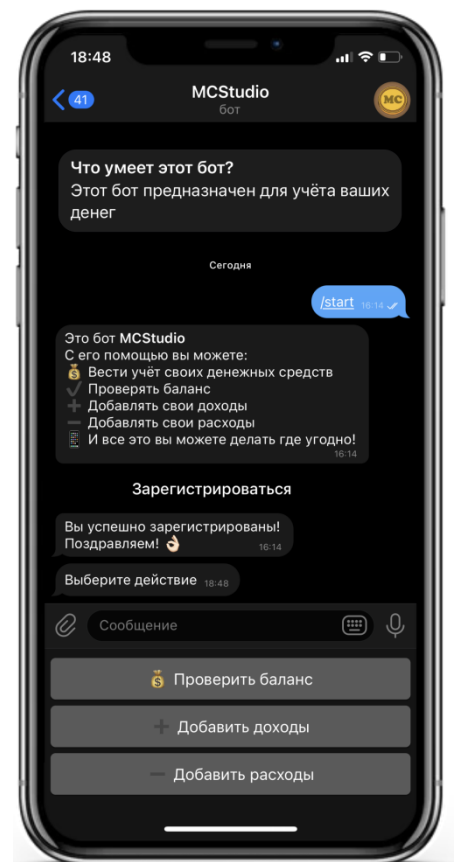
- **Проверка баланса**

Эта опция позволяет просмотреть баланс пользователя. Баланс высчитывается исходя из суммарного количества доходов и расходов пользователя. Сумма расходов вычитывается из суммы доходов, оставляя так называемый **профит** – чистую выручку.

- **Добавление доходов и расходов**

В вышеупомянутом классе **Database** существуют две важные группы методов: отвечающие за доходы и за расходы.

Нажав на «**Добавить доходы**» или «**Добавить расходы**», пользователь получает возможность предоставить сервису информацию о доходах или расходах. Значения записываются в базу



данных с помощью методов **add_user_income** и **add_user_expenses**. После нажатия кнопку появляется инструкция:

Для того, чтобы добавить доходы, необходимо перечислить их отдельными сообщениями.
Как только закончите, напишите **end**.

С помощью команды **end** программа понимает, когда прекратить поток получения информации о доходах. Программа регистронезависимая, что позволяет пользователю вводить команду в любом регистре: **End**, **END** и т.д.

3. Сервер

MCStudio расположен на сервере **Pythonanywhere**. Был создан специальный аккаунт, на котором была запущена командная строка. Все файлы проекта хранятся именно на этом сервере. С помощью предустановленного Python 3.9 был запущен бесплатный сервер на 3 месяца.
