Урок 7: Регистрация в боте

В данном уроке вы узнаете как реализовать регистрацию и авторизацию в боте на платформе **Metabot**.

Мы создадим бот, который будет собирать данные пользователя и создавать для него персону. С помощью функционала персон он будет определять регистрировался пользователь в данном боте или нет. Уже зарегистрированные пользователи будут проходить через процесс авторизации.

> Подробнее изучить работу бота вы можете с помощью нашего примера — бота в Telegram: **@RegistrationMetabot**

В этом уроке вы познакомитесь с таким функционалом платформы как **Персоны**.

> Перед началом следует ознакомиться с предыдущими уроками **Hello Humans: ваше руководство по быстрому старту** и **Metabot 101: Вывод фото в боте по REST API**

Инструкция по разработке: Создание скриптов

1. Чтобы не отвлекаться на это в будущем, сразу создадим стартовый скрипт с приветствием, маршрут связанный с ним и канал Телеграм, при помощи которого будем тестировать бота, как мы делали это в предыдущих уроках.

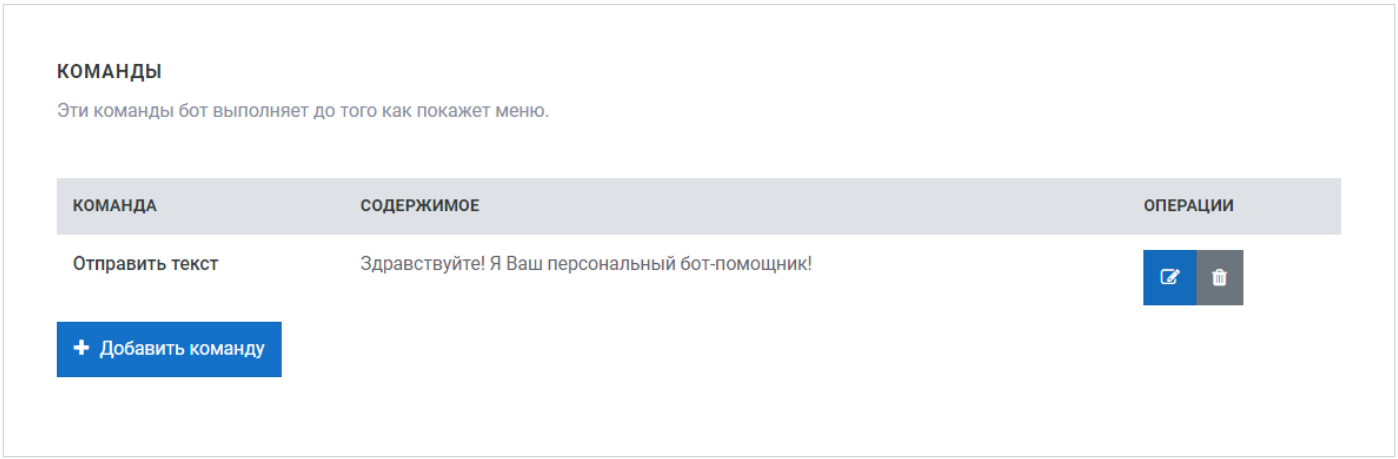
Нам потребуются следующие скрипты:

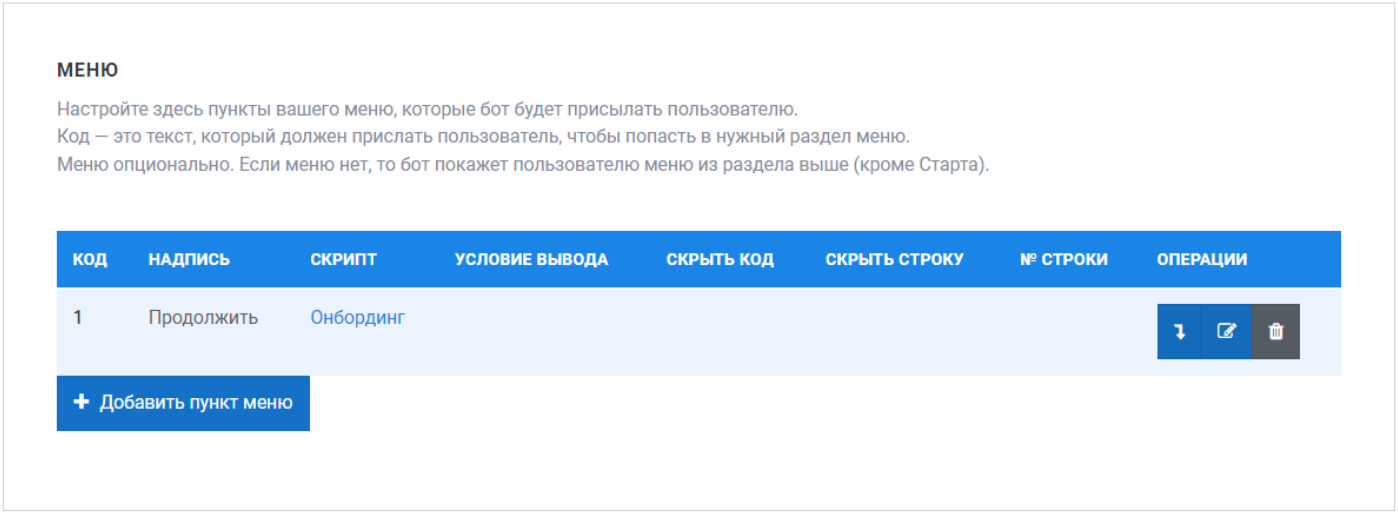
- Стартовый скрипт;
- Меню;
- Fallback;
- Скрипт с выбором регистрации\авторизации;
- Скрипты регистрации;
- Скрипты авторизации.

Перейдем к их созданию.

Стартовый скрипт и скрипт с выбором регистрации\авторизации

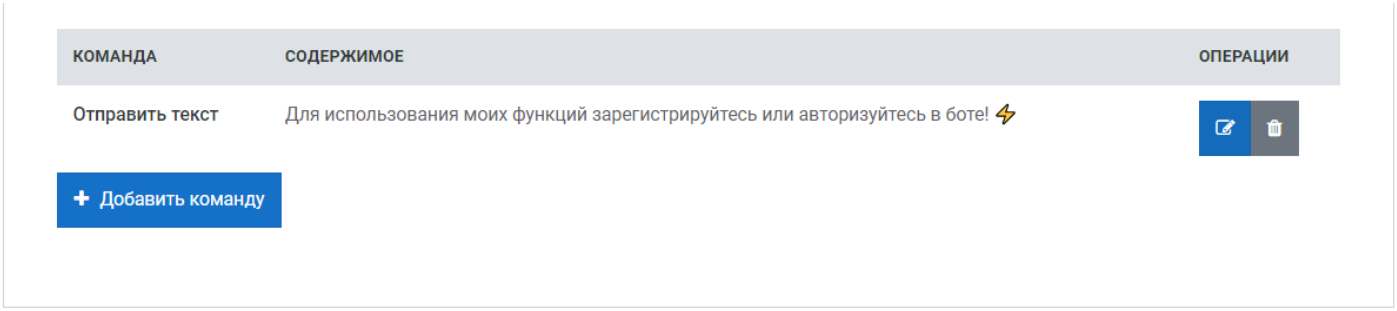
Стартовый скрипт будет активироваться всего раз после начала лидом диалога с ботом. В нем должно быть всего две команды: отправка сообщения с приветствием и переход на следующий скрипт. Переход можно сделать так же при помощи кнопки.

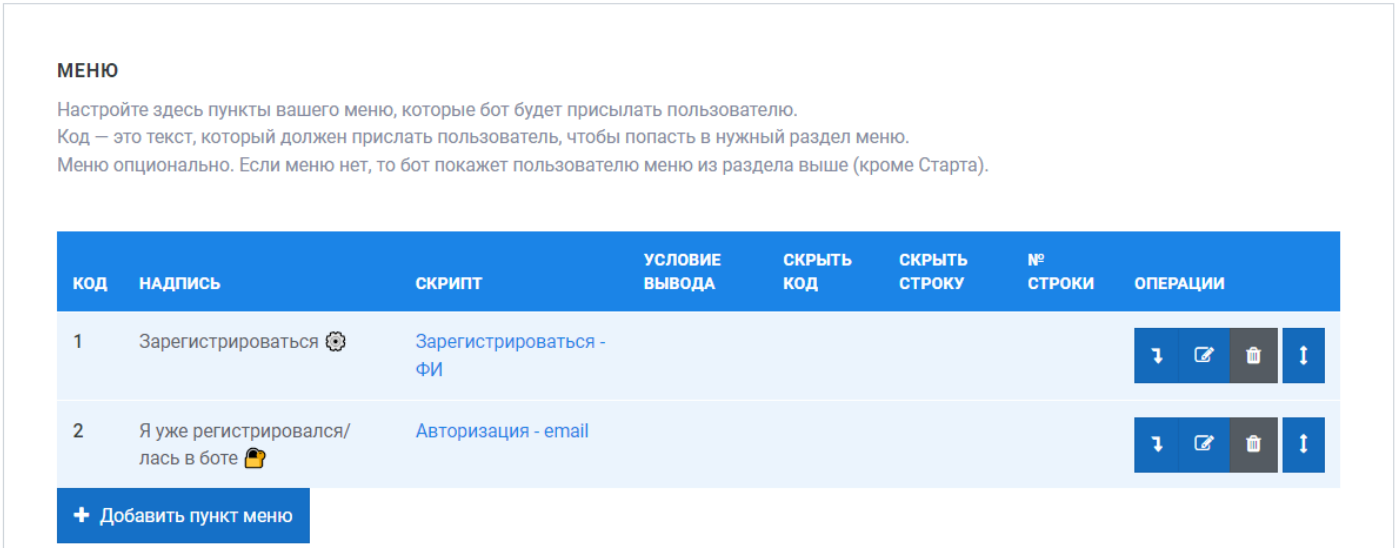




Скрипт с выбором регистрации\авторизации нужен для того чтобы пользователь не проходил лишний раз процесс регистрации. Если он уверен что уже регистрировался в боте с этого или другого лида, то он может сразу перейти к процессу авторизации.

2. Для данного скрипта нам нужно добавить команду **Отправить текст** с объяснением, что пользователю делать дальше, и две кнопки меню: одна с переходом к регистрации, другая с переходом к авторизации.

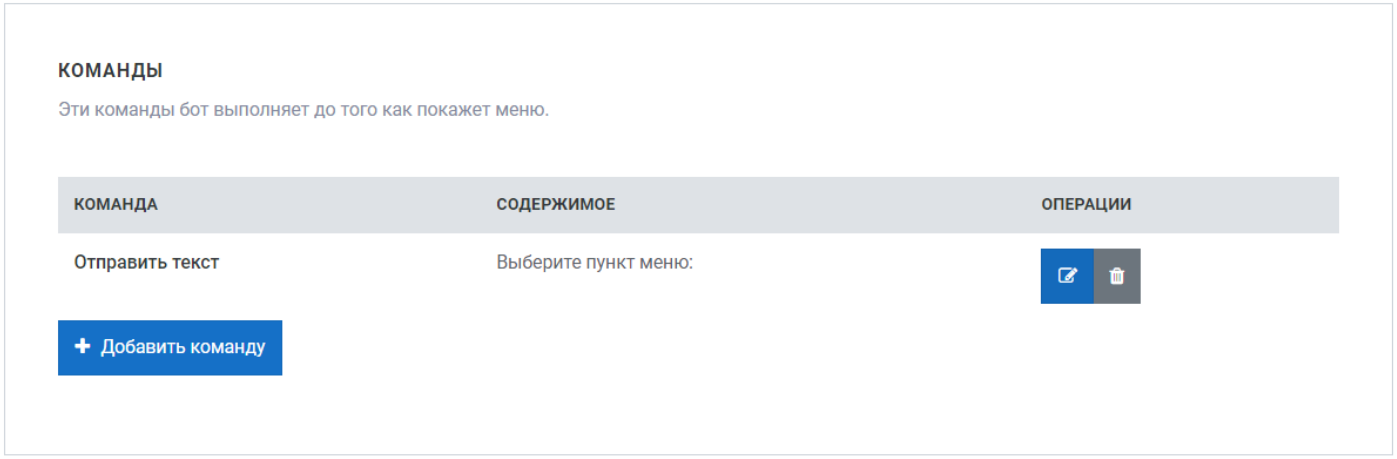


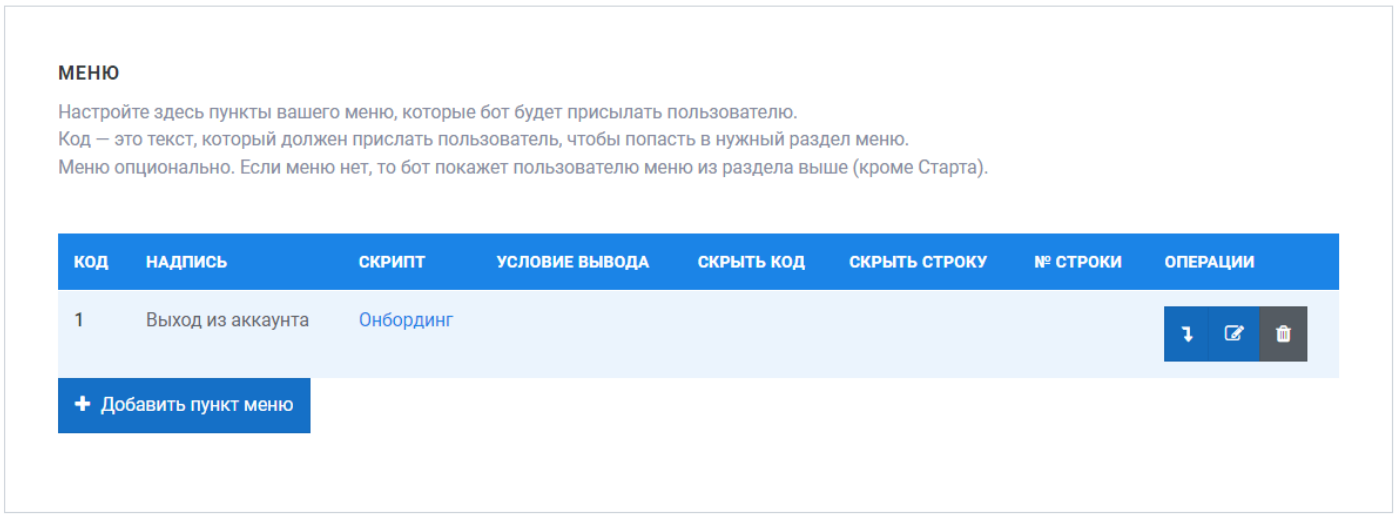


Скрипт меню и Fallback

Скрипт меню — это скрипт в который будет попадать пользователь после авторизации. После этого скрипта можно сделать для бота любой функционал, исходя из ваших потребностей.

3. Мы рекомендуем создать в данном скрипте кнопку с выходом из аккаунта в **Скрипт с выбором**, чтобы пользователь мог зарегистрироваться в боте по другой почте.



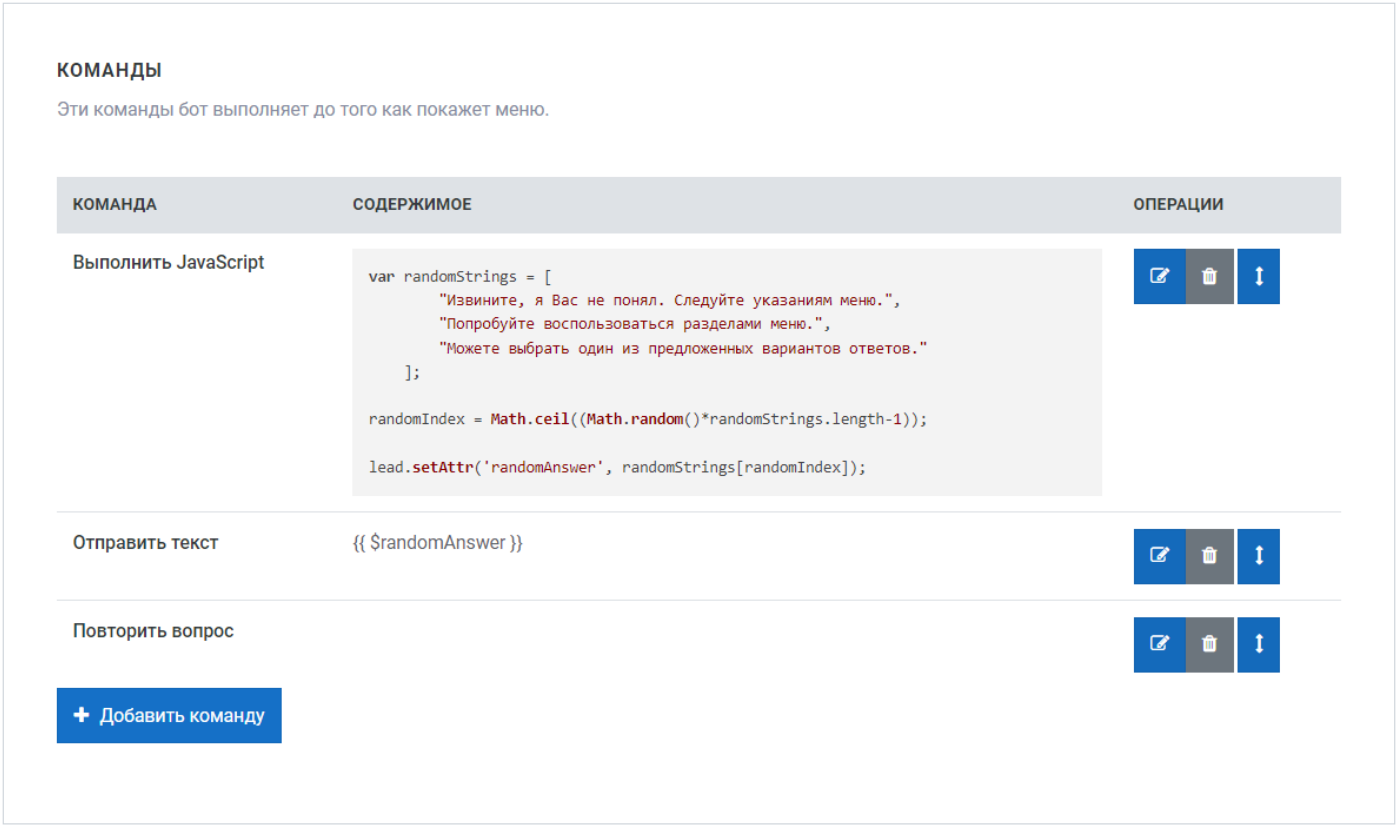


Скрипт Fallback — это скрипт, который будет выполняться в том случае, если пользователь ввел вопрос/намерение или фразу не заложенные в базе знаний вашего бота**.**

4. В свойствах данного скрипта следует установить тип Fallback:



В данном уроке мы попробуем интересный способ реализации данного скрипта.



При помощи JS скрипта можно сделать бота более живым и дать ему несколько вариантов ответа на нестандартную реакцию пользователя:

```
var randomStrings = [
        "Извините, я Вас не понял. Следуйте указаниям меню.",
        "Попробуйте воспользоваться разделами меню.",
        "Можете выбрать один из предложенных вариантов ответов."
    ];

randomIndex = Math.ceil((Math.random()*randomStrings.length-1));

lead.setAttr('randomAnswer', randomStrings[randomIndex]);
```

5. В скрипте случайным образом выбирается индекс для массива ответов, тем самым выбирается случайный ответ. Далее полученный ответ выводится лиду в команде **Отправить текст**.

6. Затем нужно добавить команду **Повторить вопрос**, чтобы бот мог вернуться к последнему сообщению отправленному пользователю перед переходом в данный скрипт.

Основные скрипты для работы бота созданы. Перейдем к созданию регистрации и авторизации.

Скрипты регистрации

В процессе регистрации бот будет собирать данные у пользователя, затем проверять почту пользователя на занятость и создавать персону с собранными данными.

Нам потребуются следующие скрипты:

- Скрипты сбора данных;
- Скрипт проверки;
- Скрипт проверки почты;
- Скрипт "Почта занята";
- Создание персоны.

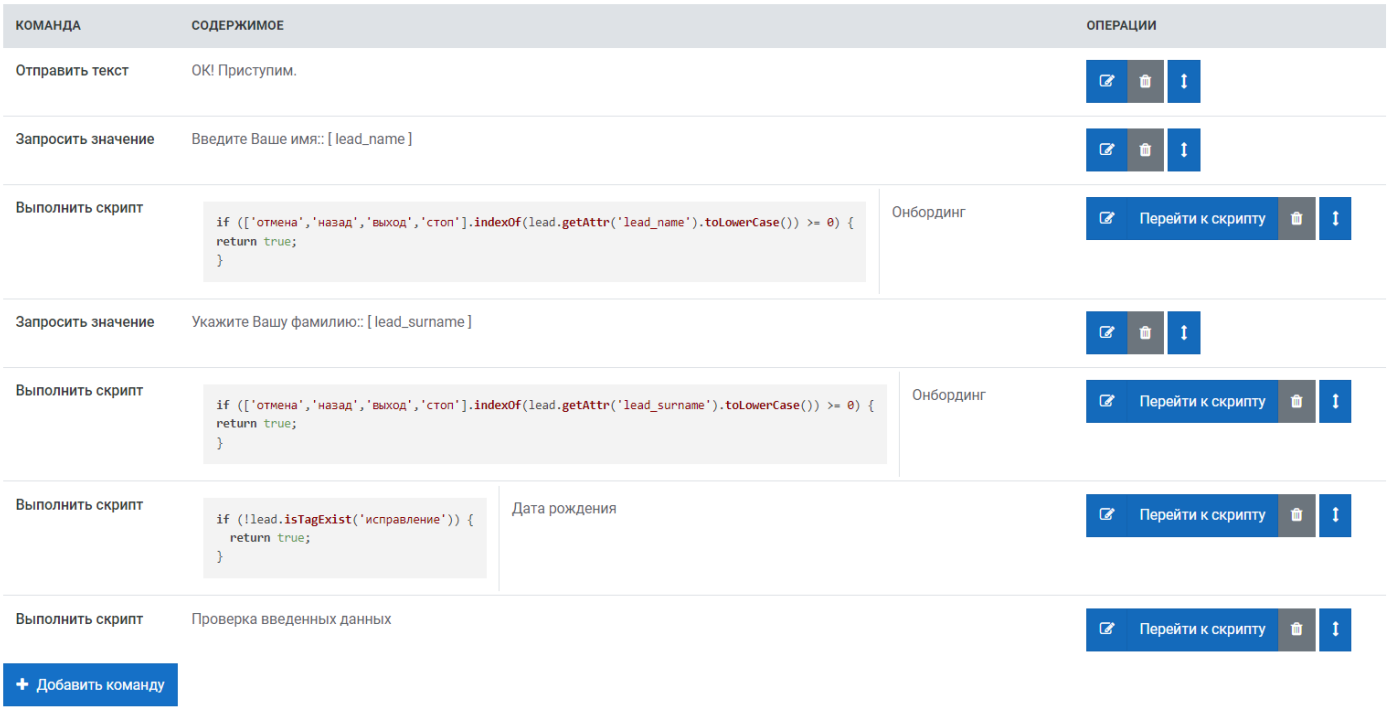
Перейдем к их созданию.

Скрипты сбора данных

Данные скрипты предназначены для того, чтобы узнать информацию о пользователе. Это может быть имя, дата рождения, пол и т.д.

1. В таком скрипте нужно создать команду **Запросить значение**, команду перехода на следующий скрипт и команду перехода на скрипт проверки.

Для примера разберем скрипт сбора информации об имени:



1. Первым делом бот запрашивает имя пользователя.
2. Затем происходит проверка: если пользователь передумал и ввел одно из ключевых слов, указанных в условии, то регистрация прерывается и бот переходит к **Скрипту**

выбора. Данную проверку можно добавлять после каждой команды **Запросить значение**.

3. Далее бот запрашивает фамилию пользователя и снова выполняет проверку.
4. Затем бот проверяет существует ли тег **исправление**. Если его нет, то происходит переход к следующему скрипту сбора данных.
5. Если же тег был найден, то бот переходит к скрипту проверки.

Проверка на прерывание регистрации:

```
if (['отмена','назад','выход','стоп'].indexOf(lead.getAttr('lead_name').toLowerCase()) >= 0) {
return true;
}
```

Данные скрипты необязательны и их может быть как много, так и вообще не быть.

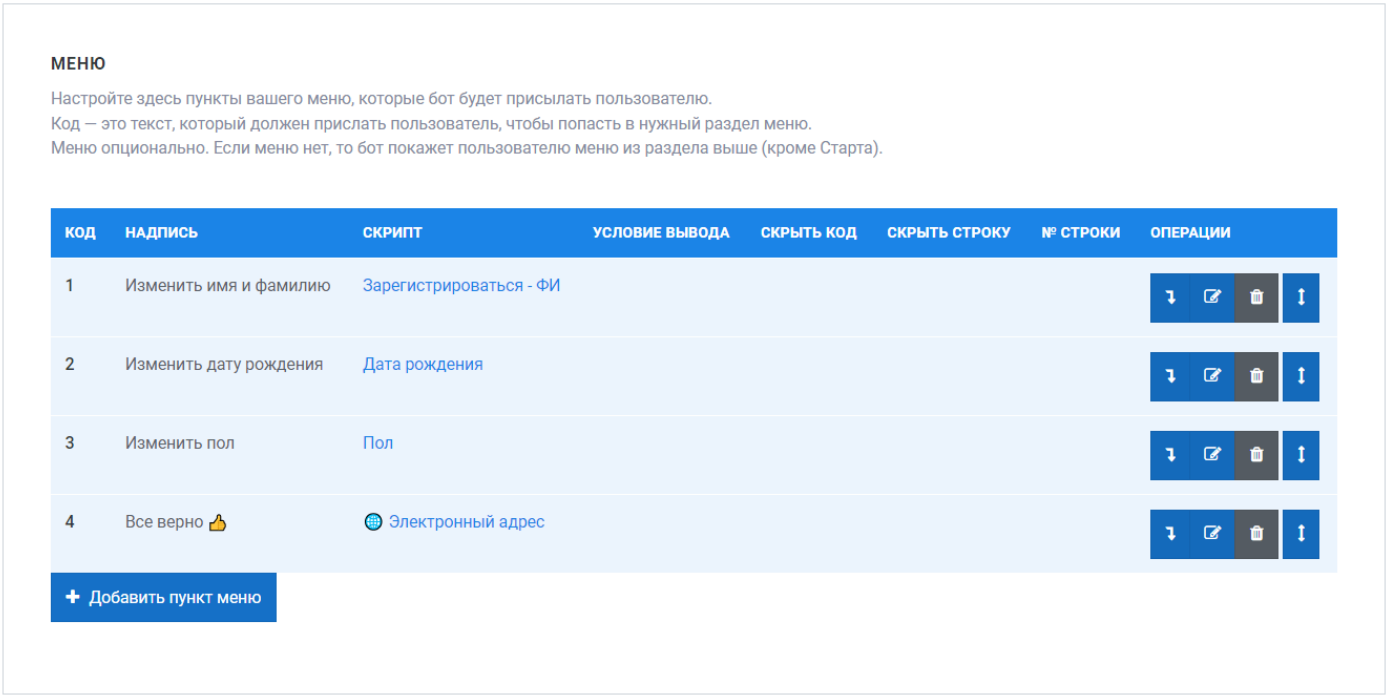
Скрипт проверки

Данный скрипт нужен для того, чтобы удостовериться, что пользователь ввел все данные верно. Если пользователь найдет ошибку, то он сможет исправить ее, в ином случае регистрация продолжится.

2. В скрипте проверки нужно создать две команды: команду **Добавить тег исправление** и **Отправить текст** со всеми ранее собранными данными.



3. Для всех данных нужно добавить в меню кнопку с переходом на скрипт запроса этих данных. Так же в меню должна быть кнопка перехода на скрипт ввода почты.



Скрипт проверки почты

В данном скрипте бот будет узнавать почту пользователя и проверять ее на занятость.

4. Первым делом нужно добавить команду **Удалить теги** и удалить тег **исправление**.

5. Затем запрашиваем почту у пользователя со следующим регулярным выражением: (отмена|[^@\s]+@[^@\s]+\.[^@\s]+).

6. Далее выполняем проверку на прерывание регистрации.

7. После этого создаем JS скрипт с проверкой на существование персоны с такой почтой и, в зависимости от ответа, выполняем скрипт завершения регистрации или **Почта занята**.

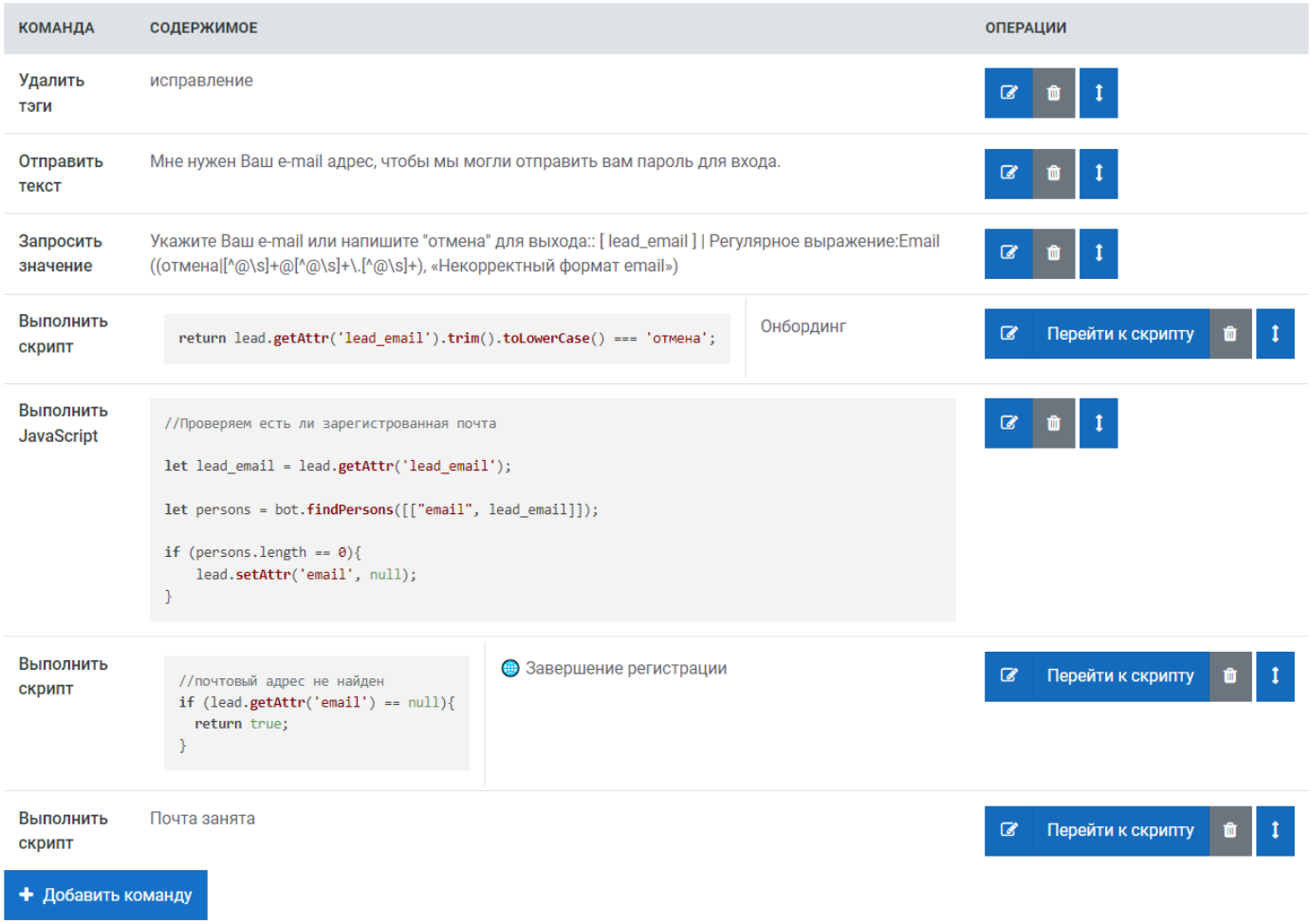
```
//Проверяем есть ли зарегистрованная почта

let lead_email = lead.getAttr('lead_email');

let persons = bot.findPersons([["email", lead_email]]);

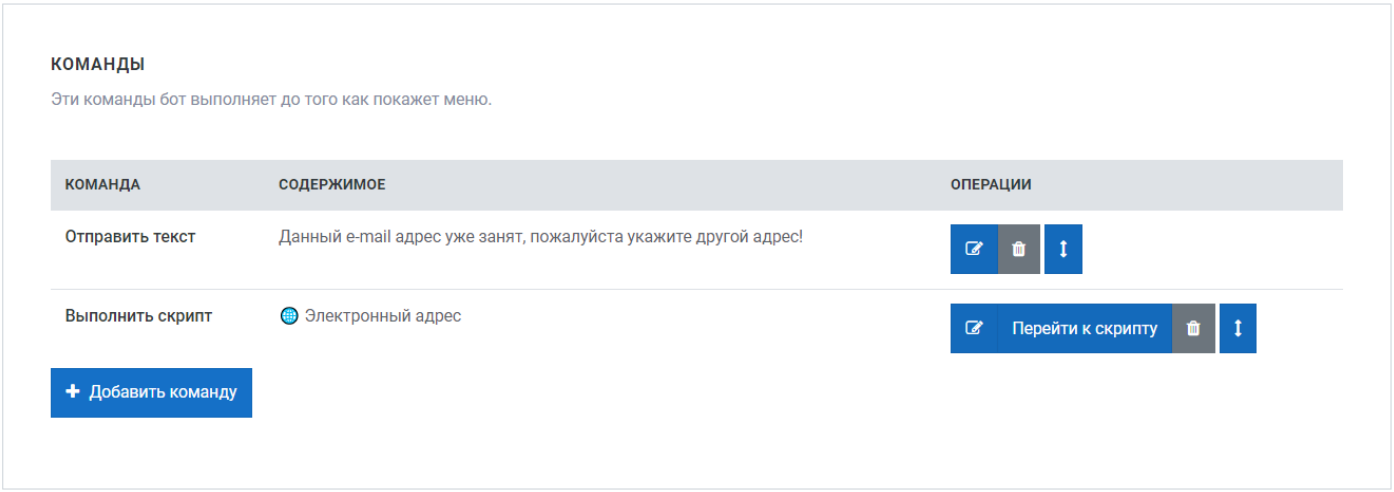
if (persons.length == 0){
    lead.setAttr('email', null);
}
```

Этот код ищет персону с заданным email в боте, и если не находит, то возвращает **null**.



Скрипт "Почта занята"

8. Этот скрипт должен содержать всего две команды: отправка текста с оповещением и переход к скрипту запроса почты.



Скрипт создания персоны

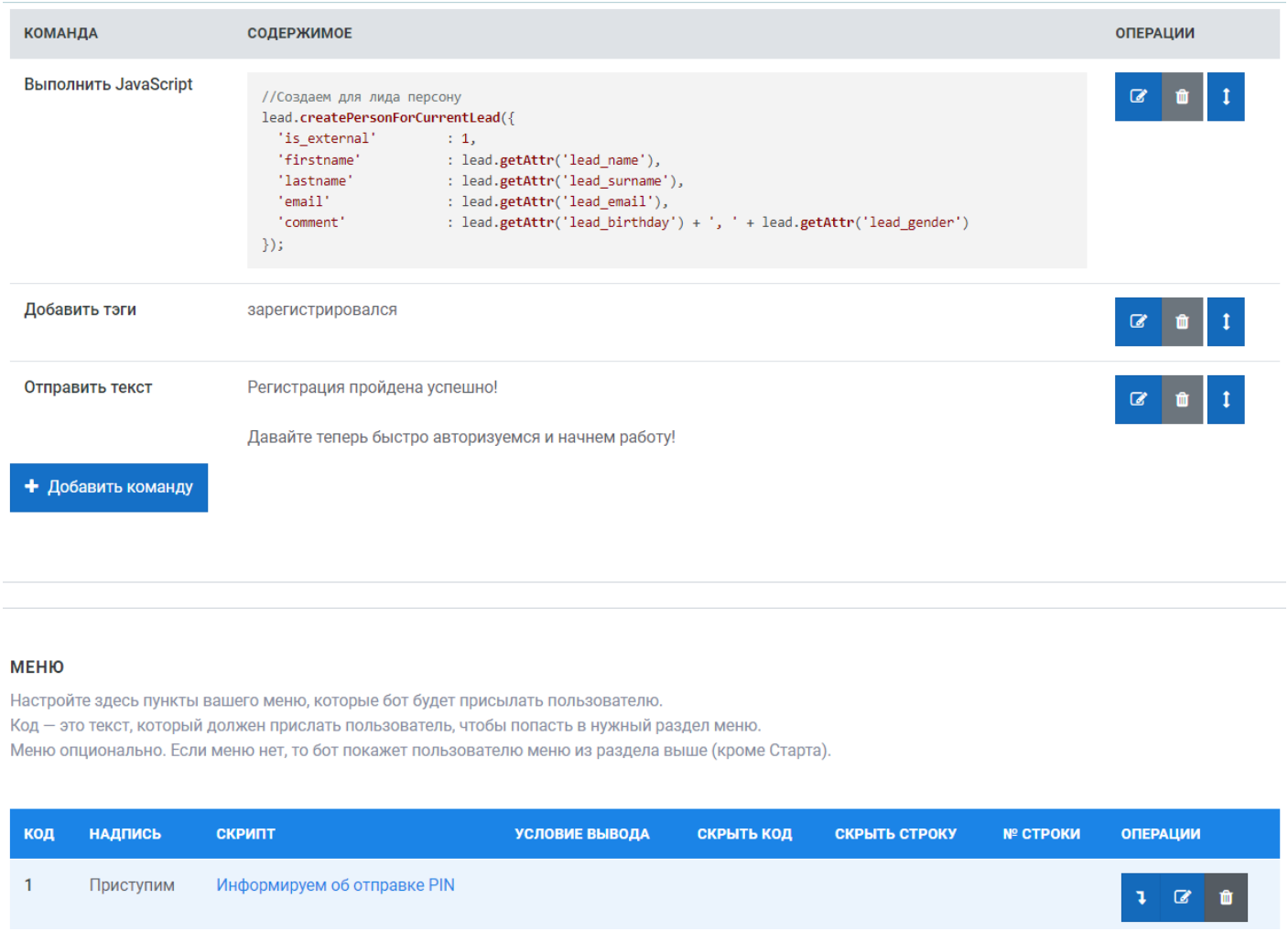
В данном скрипте происходит создание персоны с заданной почтой и переход к авторизации.

9. Для создания команды с созданием персоны используйте данный код:

```
//Создаем для лида персону
lead.createPersonForCurrentLead({
  'is_external'         : 1,
  'firstname'           : lead.getAttr('lead_name'),
  'lastname'            : lead.getAttr('lead_surname'),
  'email'               : lead.getAttr('lead_email'),
  'comment'             : lead.getAttr('lead_birthday') + ', ' + lead.getAttr('lead_gender')
});
```

В этот скрипт так же можно добавить тег **зарегистрировался**.

10. Затем отправляем текст с оповещением пользователя о завершении регистрации и добавляем в меню кнопку с переходом к авторизации.



Скрипты авторизации

В процессе авторизации бот будет узнавать почту у пользователя, затем проверять ее на занятость и отправлять PIN на нее для авторизации.

Нам потребуются следующие скрипты:

- Скрипт запроса почты;
- Скрипт проверки почты;
- Скрипт отправки PIN;
- Скрипт ввода PIN;
- Скрипт проверки PIN.

Перейдем к их созданию.

Скрипт запроса почты

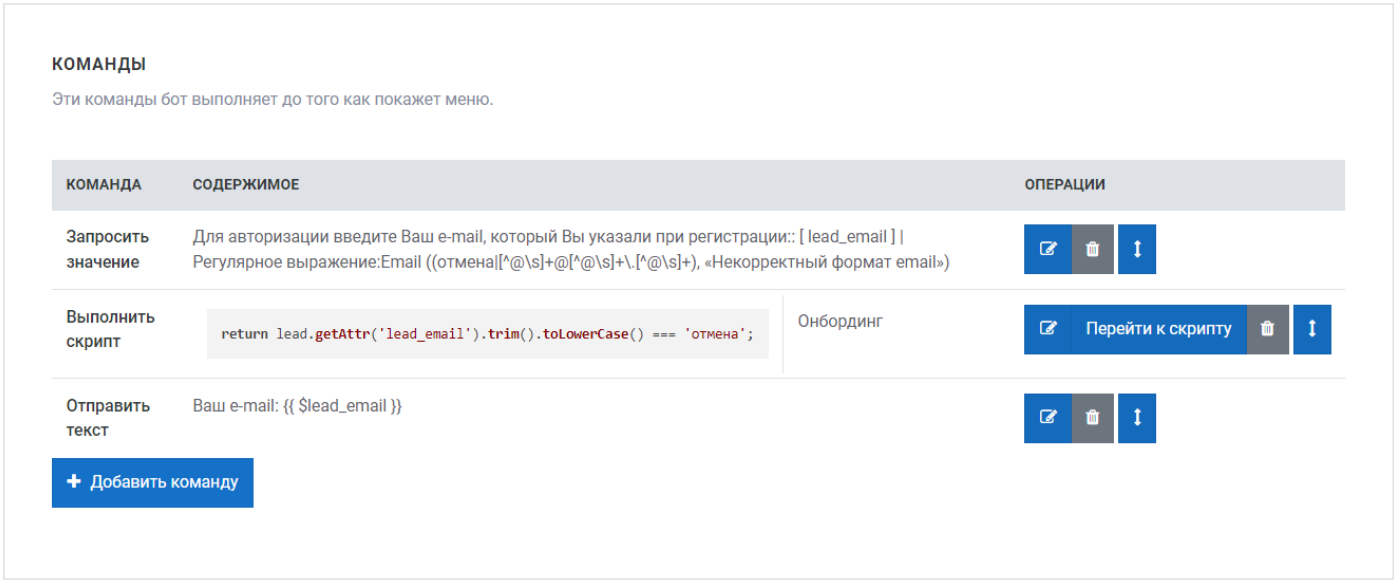
В этот скрипт пользователь попадает из **Скрипта выбора**.

Тут повторяются команды из скрипта запроса почты для регистрации.

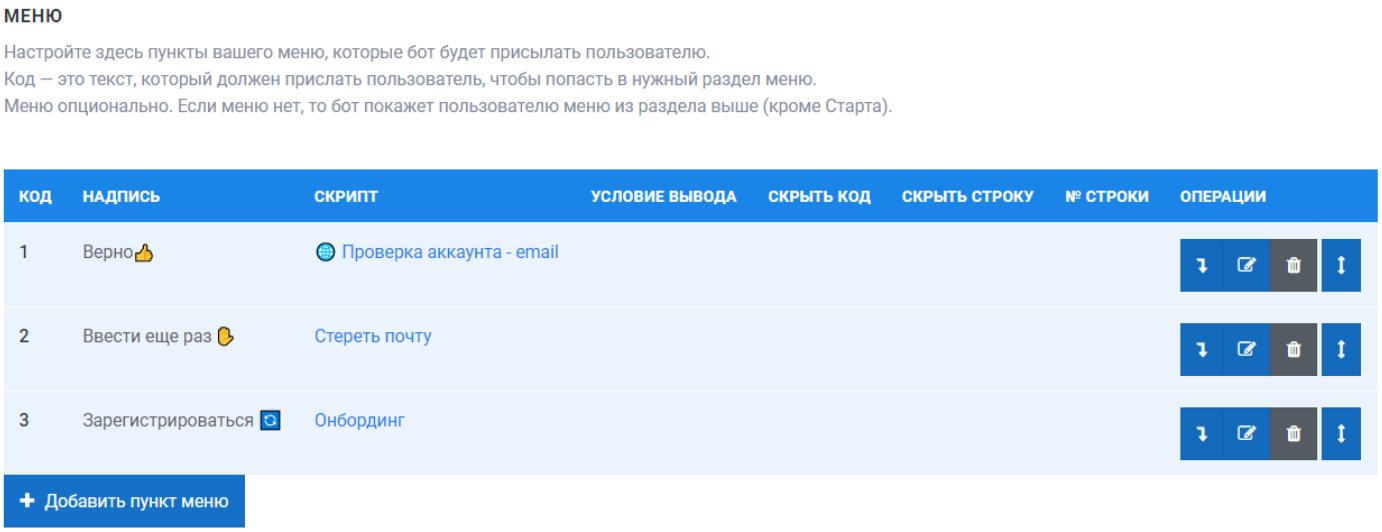
1. Запрашиваем почту у пользователя со следующим регулярным выражением: (отмена|[^@\s]+@[^@\s]+\.[^@\s]+).

2. Далее выполняем проверку на прерывание регистрации.

3. Последней командой добавляем вывод почты для проверки пользователю.



5. В меню данного скрипта добавляем кнопки перехода к следующему скрипту авторизации, замены почты и **Скрипту выбора**.



> Для исправления почты нужно добавить еще один скрипт, в котором будет находиться только команда перехода к скрипту ввода почты.

Скрипт проверки почты

Данный скрипт так же повторяет команды регистрации.

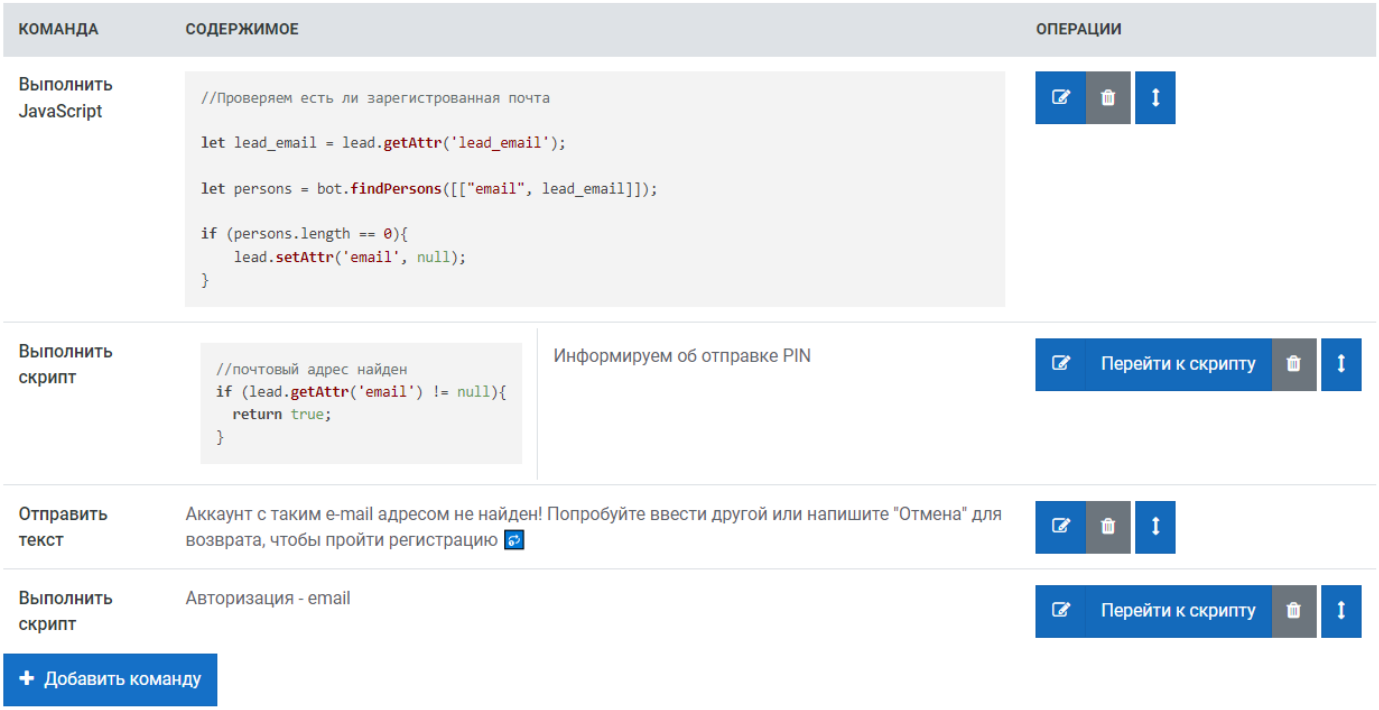
6. Создаем JS скрипт с проверкой на существование персоны с такой почтой и, в зависимости от ответа, выполняем скрипт отправки PIN или запроса почты.

```
//Проверяем есть ли зарегистрованная почта

let lead_email = lead.getAttr('lead_email');

let persons = bot.findPersons([["email", lead_email]]);

if (persons.length == 0){
    lead.setAttr('email', null);
}
```



Скрипт отправки PIN

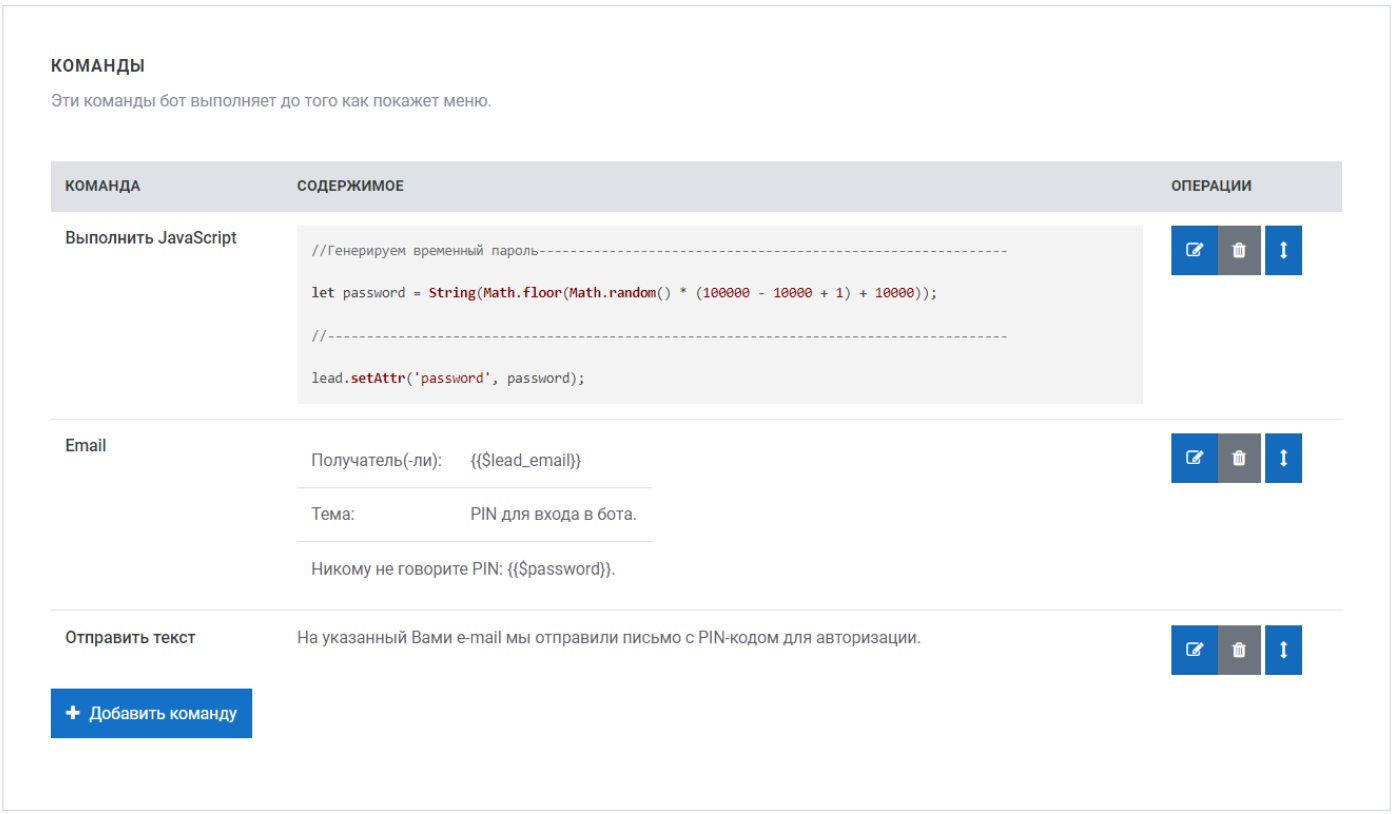
7. В данном скрипте следует создать генератор PIN-кода и команду отправки письма на почту.

```
//Генерируем временный пароль-----------------------------------------------------------

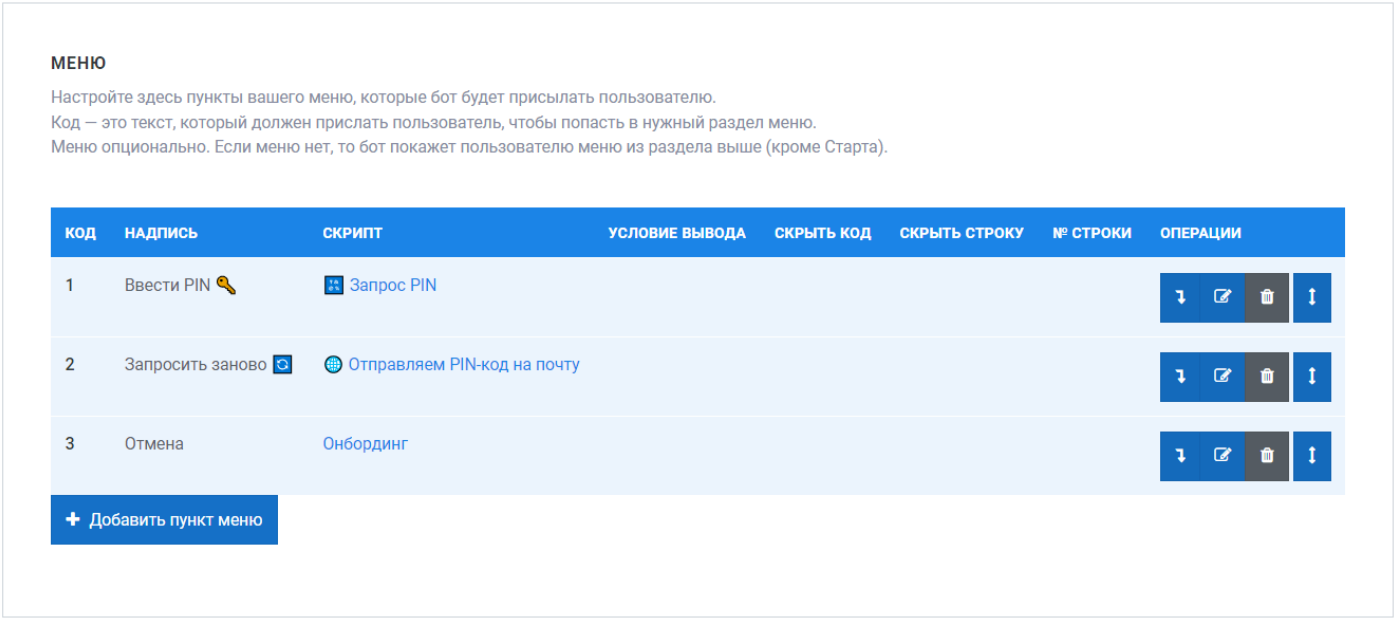
let password = String(Math.floor(Math.random() * (100000 - 10000 + 1) + 10000));

//-------------------------------------------------------------------------------------

lead.setAttr('password', password);
```



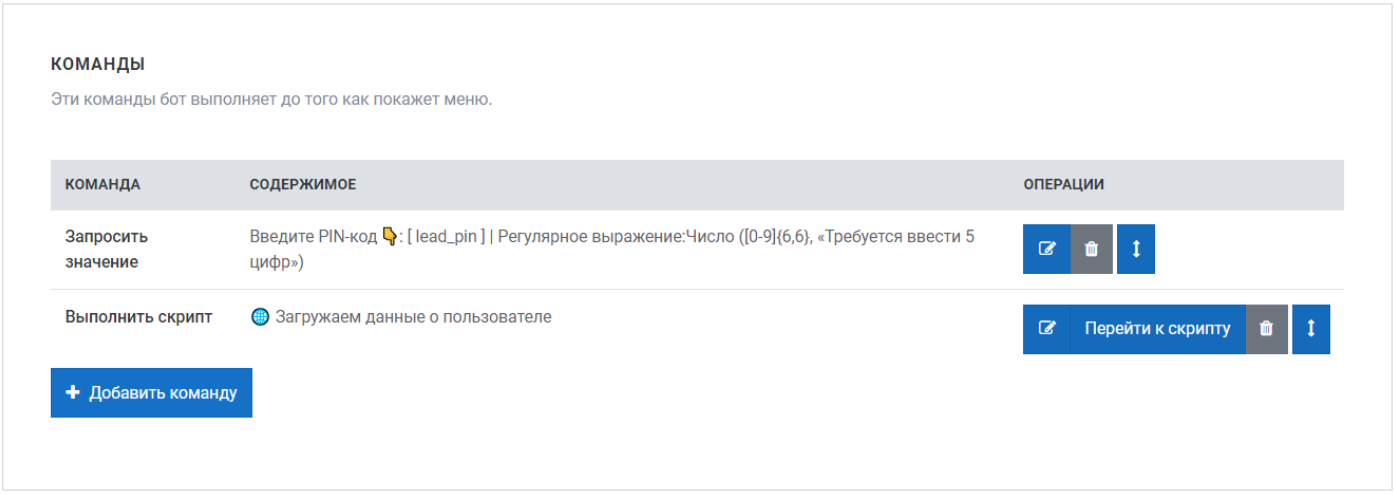
8. В меню добавляем кнопки перехода к следующему скрипту авторизации, повторной отправки PIN и **Скрипту выбора**.



> Для повторной отправки PIN нужно добавить еще один скрипт, в котором будет находиться только команда перехода к скрипту отправки PIN.

Скрипт ввода PIN

9. В данном скрипте нужны команды запроса PIN и перехода к скрипту проверки PIN.



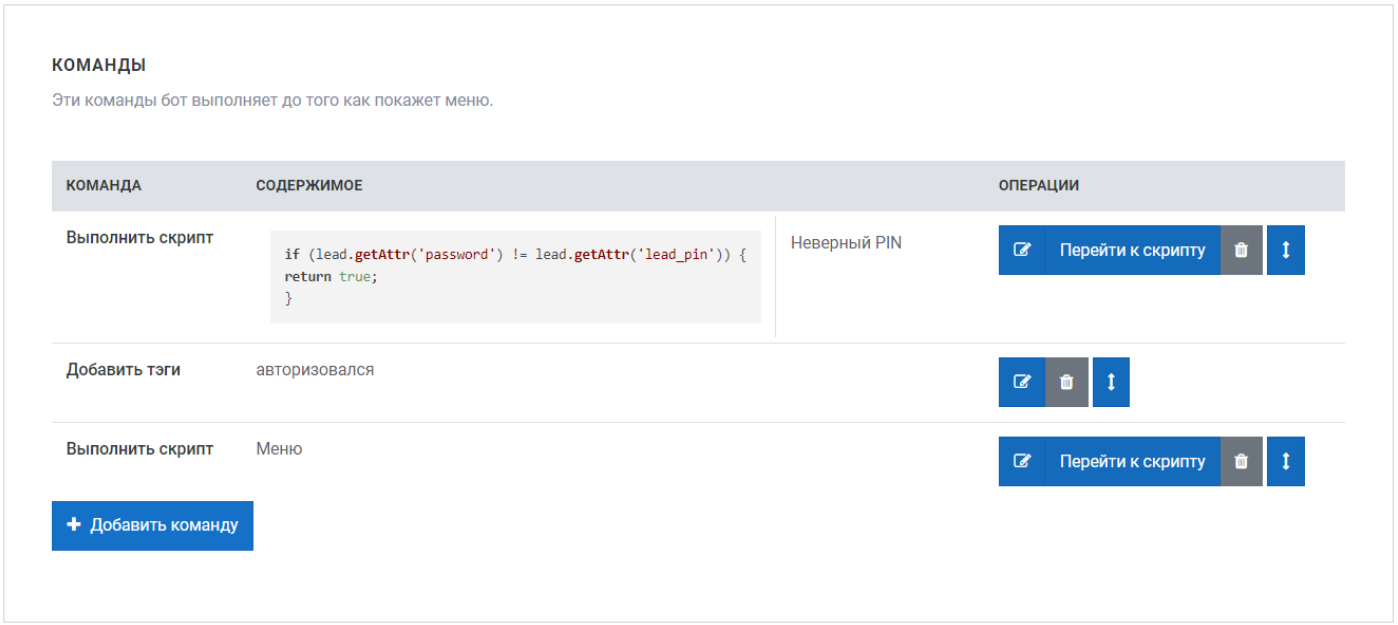
Скрипт проверки PIN и "Неверный PIN"

В скрипте проверки сравниваются отправленный на почту PIN и PIN введенный пользователем.

10. Для проверки добавляем команду с условием при выполнении которого будет вызываться скрипт "Неверный PIN".

Можно так же добавить тег **авторизовался**.

11. Последним делом добавляем команду перехода в меню.

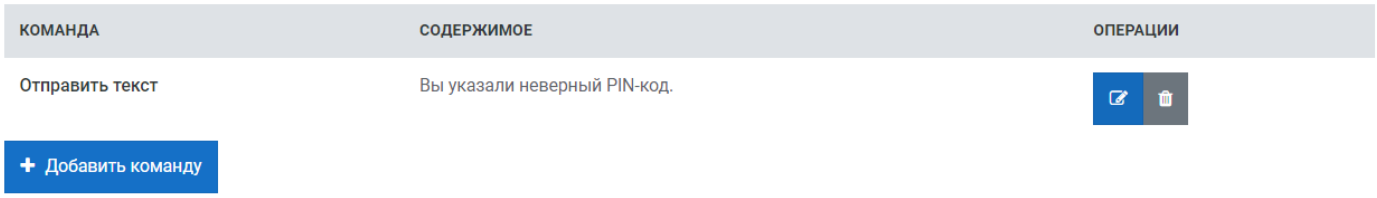


12. В скрипте "Неверный PIN" добавляем команду отправки текста с оповещением о неправильном PIN-коде.

13. В меню добавляем следующие кнопки:

- Запрос PIN;

- Отправка PIN;
- Запрос почты;
- Скрипт выбора регистрации\авторизации.

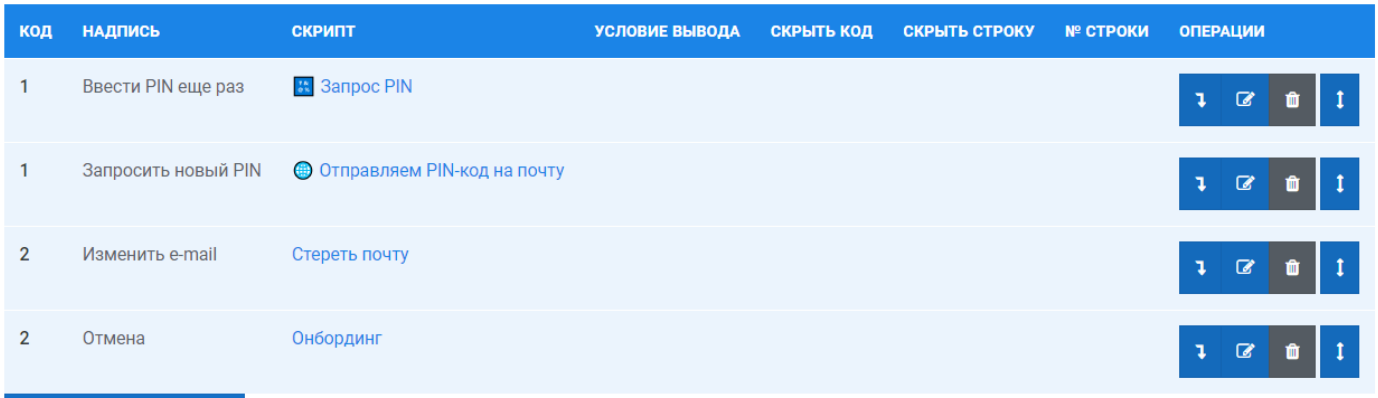


КОМАНДА	СОДЕРЖИМОЕ	ОПЕРАЦИИ
Отправить текст	Вы указали неверный PIN-код.	

+ Добавить команду

МЕНЮ

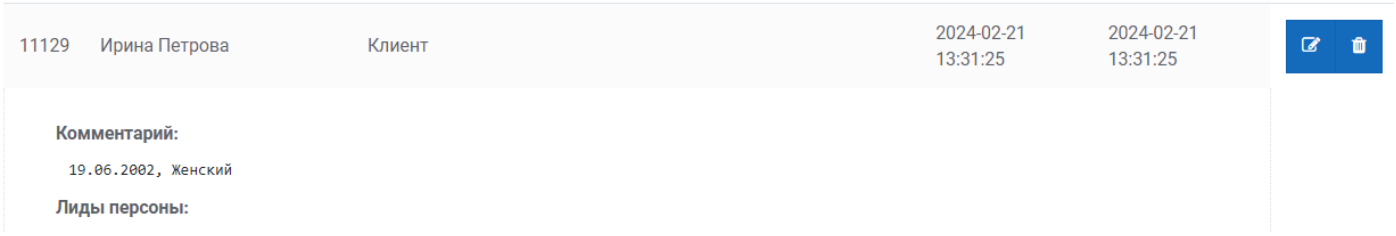
Настройте здесь пункты вашего меню, которые бот будет присылать пользователю.
Код — это текст, который должен прислать пользователь, чтобы попасть в нужный раздел меню.
Меню опционально. Если меню нет, то бот покажет пользователю меню из раздела выше (кроме Старта).



КОД	НАДПИСЬ	СКРИПТ	УСЛОВИЕ ВЫВОДА	СКРЫТЬ КОД	СКРЫТЬ СТРОКУ	№ СТРОКИ	ОПЕРАЦИИ
1	Ввести PIN еще раз	Запрос PIN					
1	Запросить новый PIN	Отправляем PIN-код на почту					
2	Изменить e-mail	Стереть почту					
2	Отмена	Онбординг					

+ Добавить пункт меню

Бот завершен! Теперь при регистрации в нем на вкладке **Персоны** платформы будут создаваться персоны:



11129	Ирина Петрова	Клиент	2024-02-21 13:31:25	2024-02-21 13:31:25

Комментарий:

19.06.2002, Женский

Лиды персоны:

Поздравляем вас с прохождением урока!

Версия #3
Ирина Петрова создал 2024-02-21 10:45:49 UTC
Ирина Петрова обновил 2024-03-20 08:58:09 UTC