

Основные шаги:

1) Создать бота, завести API-шлюз, все соединить

- Создаем бота в ТГ
- Создаем функцию в Yandex Cloud и добавляем в нее токен ТГ. Функцию можно сделать и публичной, тогда не придется создавать Api Gateway. Если все же с Api Gateway, то для него нужно создать сервисный аккаунт с правами editor и serverless.functions.invoker. В Api Gateway нужно указать function.id, function.name, id сервисного аккаунта.
- Соединить все в единое целое через запрос в браузере

Сервисный аккаунт это как пользователь с определенными правами доступа. Yandex Cloud Function - это файл index.py. Сторонние библиотеки прописываются в файле requirements.txt. В моем случае это numpy и ydb (для работы с яндексовской базой)

2) Доступ к API: лучше тестировать не в Yandex Cloud Functions, а в colab или юпитеровском ноутбуке

3) сбор обратной связи

- текст
- команда
- кнопка меню (клавиатура) <https://core.telegram.org/bots/api#inlinekeyboardbutton>.
Типы клавиатур <https://core.telegram.org/bots/features#keyboards>

В моем боте использована стандартная клавиатура (проще это делать в json, а потом конвертировать). Можно сделать и с помощью внешних библиотек, но нужно помнить, что установка библиотек в облаке потребляет ресурсы.

4) Реализация алгоритма бандита (Логика лучше вынести в отдельный файл. Но в моей реализации пока все в одном файле)

Возникшие ошибки при реализации:

- Ошибка, из-за которой не срабатывал класс в отдельной функции — забыл загрузить библиотеку numpy 🤪 в requirements.txt. Т.е. просто прерывалось выполнение сразу же.
- Текущую ручку нужно делать либо глобальной (и не забыть это указать), либо еще одним атрибутом класса eGreedy. Второй вариант удобнее, если хочется потом сделать отдельный сбор предпочтений для каждого chat_id (Это как раз реализовано. Сбор данных идет по каждому пользователю и каждому пользователю выводятся персонализированные картинки на основе его истории)
- Если возникла ошибка 504 Execution timeout exceeded – значит, что функция считалась слишком долго. Можно регулировать при настройке функции (параметр Таймаут – максимальное время выполнения функции, после которого сервис прервет выполнение, не дожидаясь ответа.)
- История сбрасывается примерно через минуту неактивности, поэтому имеет смысл сохранять результат (реализовал)