

UnicChat

Настройка ботов

21/03/2024

Настройка среды разработки ботов

Требования

Разработка ботов на UnicChat требует от вас получения и настройки нашего шаблонного кода. Для этого на вашей машине должно быть следующее.

Git

Node.js

Конфигурация бота

Независимо от типа бота, UnicChat SDK требует от вас создания конфигурации с некоторыми переменными для вашего окружения. Эта конфигурация может быть сохранена в файле `.env`, например, для локальной разработки.

Переменные окружения для ботов

Ниже приведен список всех переменных окружения, которые вы можете использовать в своем проекте.

Переменные, помеченные звездочкой (*), являются обязательными.

Переменная	Описание
UNICCHAT_URL *	URL-адрес экземпляра UnicChat, к которому необходимо подключиться. Может быть указан как <code>host:port</code> , <code>http://host:port</code> или <code>https://host:port</code> .
UNICCHAT_USE_SSL	Заставляет бота подключаться по протоколу SSL. Если значение не задано, он будет пытаться определить протокол по URL.
UNICCHAT_AUTH	Метод авторизации для бота. По умолчанию: пароль. Установите значение <code>ldap</code> , чтобы включить LDAP-логин для пользователей бота.
UNICCHAT_USER *	Имя пользователя бота (имя учетной записи, с которой пользователи будут вызывать бота). Должен быть зарегистрирован на вашем сервере UnicChat и иметь роль бота.
UNICCHAT_PASSWORD *	Пароль пользователя бота.

UNICCHAT_ROOM	Бот взаимодействует с перечисленными каналами. По умолчанию: GENERAL. Принимается список, разделенный запятыми. Позволяет боту прослушивать и отвечать на сообщения от всех вновь созданных частных групп, в которых пользователь бота был добавлен в качестве члена. Должен быть пустым, если LISTEN_ON_ALL_PUBLIC=true
RESPOND_TO_DM	Бот получает сообщения личную переписку (true/false). По умолчанию: false.
RESPOND_TO_EDITED	Бот получает отредактированные сообщения (true/false). По умолчанию: false.
ROOM_CACHE_SIZE	Размер кэша (LRU) для поиска в комнатах (ID или название комнаты).
ROOM_CACHE_MAX_AGE	Максимальное время кэша для поиска в комнатах.
DM_ROOM_CACHE_SIZE	Размер кэша для поиска в комнатах личной переписки.
DM_ROOM_CACHE_MAX_AGE	Максимальное время кэша для поиска в комнатах личной переписки.
INTEGRATION_ID	Идентификатор, применяемый к объекту сообщения для источника интеграции. По умолчанию: js.SDK
ADMIN_USERNAME	Имя пользователя-администратора для вызовов API, используемых в тестах SDK.
ADMIN_PASS	Пароль пользователя-администратора для вызовов API, используемых в тестах SDK.

Архитектура ботов

Архитектура ботов состоит из двух основных частей: UnicChat host и bot host. Эти две части связаны друг с другом с помощью методов UnicChat SDK. Бот-хост работает вне хоста UnicChat и состоит из инфраструктуры бота. Инфраструктура бота, в свою очередь, состоит из SDK, адаптера и фреймворка.

Бот, будучи подписанным на сообщения комнаты (все или определенные), обрабатывает их и отвечает пользователю в интерактивном режиме. Разговорная модель может быть разработана и управляться с помощью инструментария и фреймворков на хосте бота.

Сегодня многие боты создаются с помощью бот-фреймворков или платформ на основе популярных языков программирования, таких как NodeJS и Python.

Как работает архитектура ботов

Создатели ботов разрабатывают логику бота с помощью своего любимого фреймворка. Эта логика определяет поведенческое взаимодействие с пользователями, наделяя бота

способностью распознавать контекст разговора. Таким образом, бот может сосредоточиться на том, чего хотят добиться пользователи, и даже использовать проактивный подход, чтобы найти правильное решение и создать лучший пользовательский опыт.

Благодаря возможности прослушивать сообщения и отвечать на них, бот может быть интегрирован в хост UnicChat, чтобы взаимодействовать с пользователями в разговорной манере. Для такой интеграции создатели ботов используют адаптеры, которые могут перевести логику, заданную с помощью полномочий фреймворка, в формат, понятный хосту UnicChat. Адаптеры используют методы UnicChat SDK для взаимодействия с хостом UnicChat.

Пользователи и боты

Для работы ботов в UnicChat требуются учетные записи пользователей с ролью бота. Кроме наличия определенного набора разрешений, учетные записи ботов рассматриваются как обычные учетные записи пользователей на хосте UnicChat. Однако, чтобы предотвратить возможную путаницу между пользователями, сообщения от онлайн-ботов по умолчанию помечаются ярлыком "Бот".

Администратор бота

Администраторы (и только администраторы) могут создавать учетные записи ботов и настраивать права роли. Это включает в себя настройку учетных данных, которые бот использует для подключения, а также полей пользователя, к которым боты имеют доступ для обеспечения конфиденциальности и безопасности. Команда UnicChat работает над улучшением представления управления ботами и повышением будущего потенциала активируемых пользователем ботов.

Хост ботов

Бот-хост - это машина, на которой работают боты. Это может быть пустой сервер, сетевой кластер, виртуализированная среда или облако.

Бот-хост может управлять и запускать множество различных ботов, а также множество экземпляров любого бота.

Потоки сообщений

Каждый пользователь UnicChat имеет коллекцию своих сообщений, которые могут быть переданы потоком сервера. Боты подписываются на коллекцию сообщений для своего аккаунта, создавая поток, который обновляется каждый раз, когда сообщение отправляется либо непосредственно им, либо в любую комнату, к которой они присоединились.

UnicChat SDK

SDK - это низкоуровневый программный модуль, предоставляющий внешним потребителям интерфейс для подписки на потоки сообщений, отправки сообщений и вызова методов на сервере UnicChat через WebSockets/DDP.

С помощью SDK пишутся адаптеры для фреймворков ботов.

Адаптер фреймворка

Адаптер (также называемый коннектором или промежуточным ПО) - это программный компонент, который используется для связи фреймворка с хостом UnicChat.

Адаптер позволяет создателям ботов проектировать и строить взаимодействие ботов в соответствии с правилами и практиками фреймворка, не вникая в детали низкоуровневого программирования и методы взаимодействия с хостом UnicChat.

Адаптеры используют UnicChat SDK для взаимодействия с хостом UnicChat.

Бот-фреймворк

Фреймворки ботов обеспечивают уникальные подходы к созданию разговорных интерфейсов и позволяют разработчикам сосредоточиться на проектировании и управлении сложной логикой ботов.

Большинство фреймворков можно запускать на собственном хосте ботов.

Масштабирование ботов

В производстве сотни или тысячи пользователей могут вести беседы с ботом одновременно. Возможность масштабирования бота от одной беседы до множества одновременных бесед очень важна.

Фреймворки/платформы для ботов поддерживают разные степени масштабируемости. Чтобы рост числа пользователей не вызвал проблем с архитектурой и производительностью, следует грамотно выбрать фреймворк для бота. Например, рассмотреть потенциальные проблемы масштабируемости еще до того, как вы начнете реализовывать логику бота

Внешние сервисы

Ценность большинства производственных ботов зависит от их способности получать доступ к внешним данным и системам (сервисам) и манипулировать ими. Внешние сервисы, такие как базы данных, системы планирования, CRM и NLP, могут использоваться для предоставления данных или обработки разговоров.

Фреймворк упрощает связь бота с этими сервисами и обеспечивает легкий доступ к ним через хост бота.

Разработка бота UnicChat SDK

Этот бот - простой пример того, как можно использовать методы UnicChat JS SDK. Бот является базовым и не обрабатывает ошибки, различные типы сообщений, перезагрузки сервера и другие ситуации возникающие в процессе работы.

Краткое руководство по запуску UnicChat SDK Bot

UnicChat Bot - это пакет для Node.js который работает с Node и npm.

Установка пакета UnicChat SDK

Откройте терминал в директории, в которой вы хотите разработать своего бота, и выполните следующую команду

```
npm init -y
```

```
npm install @unicchat/sdk
```

Создание файлов бота

Для запуска самого простого бота необходимо создать два файла: первый отвечает за работу сервера бота, а второй – за ответы бота на сообщения пользователей.

- создайте файл server.js со следующим содержимым:

```
const { driver } = require('@unicchat/sdk');
const respmap = require('./reply');

//параметры окружения
const HOST = '<UNICCHAT HOST>';
const USER = '<BOT USER NAME>';
const PASS = '<BOT USER PASS>';
const BOTNAME = '<UNIC CHAT BOT ALIAS>';
const SSL = '<SSL USAGE>';
const ROOMS = ['<UNICCHAT CHANNEL>'];
var myUserId;

const runbot = async () => {
  const conn = await driver.connect({ host: HOST, useSsl: SSL })
  myUserId = await driver.login({ username: USER, password: PASS });
  const roomsJoined = await driver.joinRooms( ROOMS );
  console.log('joined rooms');
```

```

const subscribed = await driver.subscribeToMessages();
console.log('subscribed');

const msgloop = await driver.reactToMessages( processMessages );
console.log('connected and waiting for messages');

const sent = await driver.sendToRoom( BOTNAME + ' is listening ...', ROOMS[0]);
console.log('Greeting message sent');
}

const processMessages = async(err, message, messageOptions) => {
if (!err) {
  if (message.u._id === myUserId) return;
  const roomname = await driver.getRoomName(message.rid);

  console.log('got message ' + message.msg)
  var response;
  if (message.msg in respmap) {
    response = respmap[message.msg];
  }
  const sentmsg = await driver.sendToRoomId(response, message.rid)
}
}

runbot()

```

Настройте содержимое раздела «параметры окружения» в соответствии с вашим сервером и учетными данными пользователей.

Убедитесь, что <BOT USER NAME> имеет роль BOT на сервере.

- создайте файл reply.js со следующим содержимым:

```

const respmap = {
  "Привет" : "привет",
  "Ты бот?" : "Да, я бот"
};

module.exports = respmap;

```

Запуск бота

Запустите следующую команду в терминале:

```
node server.js
```

После выполнения последней команды бот UnicChat SDK попытается подключиться к экземпляру UnicChat и прослушать сообщения в комнате “general”.

Взаимодействие с ботом

Войдите в чат как обычный пользователь (не пользователь типа BOT), перейдите в общую комнату и поговорите с только что созданным ботом.