**telegramBotTask**

Бот, который отправляет определенный текст сообщения через Telegram-бота определенным пользователям.

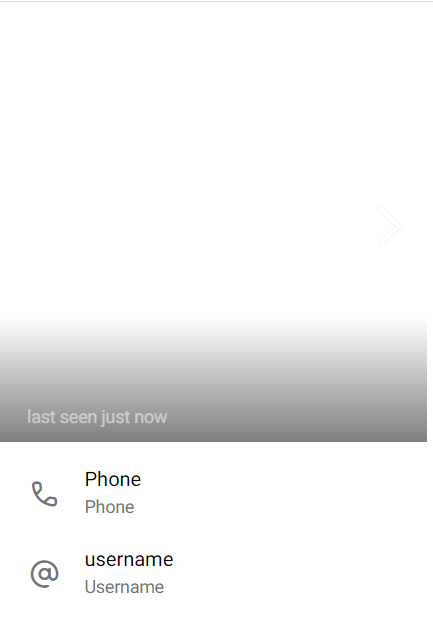
**Необходимые условия для отправки сообщений**

Для отправки сообщения пользователю через контроллер, он должен:

1. Запустить бота. Либо отправив команду «/start», либо отправить любое сообщение боту.



1. Пользователь должен иметь параметр «Username» для отправки ему сообщений через бот.



**Описание кода программы**

1. Контроллер «MessageController»

Код контроллера:

@RestController  
@RequestMapping("/api/message")  
public class MessageController {  
 @Autowired  
 private TelegramBot bot;  
  
 @PostMapping("/sendMessage")  
 public ResponseEntity<String> sendMessage(@RequestBody Message message) throws InterruptedException {  
  
 return bot.sendMessageToChat(message.getUsernamesList(), message.getText());  
 }  
}

Состоит из одного Post-запроса «sendMessage». В него передается Message который состоит из: «text» – сообщение, которое мы хотим отправить и «usernamesList» – коллекция имен пользователей, которым нужно отправить сообщение.

При выполнении Post-запроса идет обращение к методу «sendMessageToChat» класса TelegramBot, передавая туда «usernamesList» и «text».

1. Класс «UserService»

Код класса:

@Service

public class UserService {

@Autowired

public RestTemplate restTemplate;

@Value("${telegram.bot.token}")

private String botToken;

@Autowired

private UserRepository userRepository;

public void setUserChatId(){

String request = String.format("https://api.telegram.org/bot%s/getUpdates", botToken);

List<Root> rootList = restTemplate.getForObject(request, Result.class).getResult();

for (Root root : rootList){

Chat chat = root.getMessage().getChat();

if(!userRepository.UserMap.containsKey(chat.getUsername())){

userRepository.UserMap.put(chat.getUsername(), String.valueOf(chat.getId()));

}

}

}

}

В данном классе 1 метод «setUserChatId». Этот метод делает запрос к API Телеграм и получает все сообщения, отправленные пользователями. Из ответа API можно получить имена всех пользователей и chatId которые запустили бота. Далее эти значения записываются в Map из класса UserRepository, если пользователь ранее был записан цикл этой проверки проходит к следующей итерации. В качестве ключа устанавливается имя пользователя(username), а в качестве значения chatId.

1. Класс «TelegramBot»

Код класса:

@Component

public class TelegramBot extends TelegramLongPollingBot {

@Value("${telegram.bot.token}")

private String botToken;

@Value("${telegram.bot.username}")

private String botUsername;

@Autowired

private UserService userService;

@Autowired

private UserRepository userRepository;

@Override

public void onUpdateReceived(Update update) {

// Здесь будет логика обработки сообщений

}

@Override

public String getBotUsername() {

return botUsername;

}

@Override

public String getBotToken() {

return botToken;

}

public ResponseEntity<String> sendMessageToChat(List<String> usernames, String messageText) throws InterruptedException {

userService.setUserChatId();

for (String username : usernames) {

if(userRepository.UserMap.containsKey(username)){

SendMessage message = new SendMessage();

message.setText(messageText);

message.setChatId(userRepository.UserMap.get(username));

try {

execute(message);

} catch (TelegramApiException e) {

e.printStackTrace();

}

}

else {

return new ResponseEntity(String.format("User with this username %s not found", username), HttpStatusCode.valueOf(404));

}

}

return new ResponseEntity("Message send", HttpStatusCode.valueOf(200));

}

}

Здесь мы конфигурируем своего бота наследуясь от абстрактного класса «TelegramLongPollingBot», переопределяя функции и устанавливая параметры нашего бота которые прописаны в файле application.properties. В функции «sendMessageToChat» мы сначала вызываем метод «setUserChatId» из класса UserService. Далее мы проходим по коллекции имен пользователей, которую мы передали в запросе и проверяем есть ли пользователь с таким именем пользователя в UserMap из класса UserRepository. Если пользователя с таким именем нет возвращается сообщение об ошибке и код ошибки ­– 404, если пользователь есть в Map, то создается экземпляр класса SendMessage и устанавливается текст сообщения и chatId, который мы ищем по ключу имени пользователя(username) в в UserMap из класса UserRepository. Если сообщения отправлены всем пользователям возвращается сообщение об успешной операции и код ответа – 200.