

Impact Factor:

ISRA (India) = 3.117
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
PIHHC (Russia) = 0.156
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2019 Issue: 07 Volume: 75

Published: 08.07.2019 <http://T-Science.org>

QR – Issue



QR – Article



Vadim Andreevich Kozhevnikov

Peter the Great St.Petersburg Polytechnic University
Senior Lecturer
vadim.kozhevnikov@gmail.com

Kirill Vadimovich Yarulin

Peter the Great St.Petersburg Polytechnic University
student
kirillyarulin@mail.ru

INTEGRATING ONLINE COUNSELING SERVICES WITH JUST AI CONVERSATIONAL PLATFORM

Abstract: This article is devoted to the integrating of Zendesk Chat (Zopim), Webim and Bitrix24 online consulting services with Just AI Conversational Platform (JAICP) chatbots platform. A brief description of the subject area, an overview of the technologies necessary for the integration and the main points of the software implementation are presented. The conclusion describes the testing of the results.

Key words: chatbot, Just Ai Conversational Platform, Zendesk Chat, Webim, Bitrix24.

Language: Russian

Citation: Kozhevnikov, V. A., & Yarulin, K. V. (2019). Integrating Online Counseling Services with Just Ai Conversational Platform. *ISJ Theoretical & Applied Science*, 07 (75), 26-35.

Soi: <http://s-o-i.org/1.1/TAS-07-75-7> **Doi:**  <https://dx.doi.org/10.15863/TAS.2019.07.75.7>

Classifiers: Computer science, computer engineering and automation.

ИНТЕГРАЦИЯ СЕРВИСОВ ОНЛАЙН КОНСУЛЬТИРОВАНИЯ С JUST AI CONVERSATIONAL PLATFORM

Аннотация: Данная статья посвящена интеграции сервисов онлайн консультирования Zendesk Chat (Zopim), Webim и Bitrix24 с платформой для создания чат-ботов Just AI Conversational Platform (JAICP). Представлены краткое описание предметной области, обзор необходимых для интеграции технологий и основные моменты программной реализации. В заключении описано тестирование полученных результатов.

Ключевые слова: чат-бот, Just Ai Conversational Platform, Zendesk Chat, Webim, Bitrix24.

Введение

С каждым годом популярность чат-ботов только возрастает. Все больше компаний использует возможности виртуальных помощников для самых различных целей: для составления заказов клиентов, ответов на их вопросы и выполнения множества других рутинных задач обслуживания. На фоне такой популярности, многие компании разрабатывают специальные платформы, позволяющие более удобно и быстро разрабатывать чат-ботов для самых различных целей. Целью нашей работы была интеграция сервисов онлайн

консультирования Webim, Zendesk Chat и Bitrix24 с Just AI Conversation Platform (JAICP).

Чат-бот, виртуальный помощник (англ. chatbot) — это компьютерная программа, осуществляющая диалог с пользователем через текстовый или голосовой интерфейс [1]. Боты позволяют значительно сэкономить средства компании, так они могут общаться с миллионами пользователями одновременно, что позволяет не нанимать большое количество людей в службу поддержки. В 2018 г. в России начался активный рост рынка чат-ботов, поддерживающих технологию NLU (Natural Language Understanding,

Impact Factor:

ISRA (India) = 3.117
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.156
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

понимание естественного языка). Согласно оценкам Business Insider [2], уже к 2020 году около 80% компаний будет использовать чат-боты. По прогнозам Just AI [3], по итогам 2023 г. объем российского рынка чат-ботов достигнет 33 млрд. руб. Такой высокий темп роста рынка чат-ботов с технологией NLU можно объяснить ранней стадией его формирования. Первыми компаниями, которые заинтересовались этим рынком, стали IT- и телеком-компании: доля проектов в денежном выражении в 2017 г. составляла 44%. Далее все более активно начали внедрять речевые технологии банки, страховые и финансовые компании. По итогам 2018 года их доля составляет 28%. Одной из особенностей российского рынка чат-ботов является низкая активность мировых игроков, так как далеко не все их продукты поддерживают русский язык.

Принцип работы чат-бота

Всем чат-ботам приходится работать с естественной речью пользователя. Правильно понять и обработать естественный язык, особенно разговорный стиль, — очень сложная задача. Для решения этой проблемы существует отдельный раздел искусственного интеллекта и математической лингвистики, который называется обработка естественного языка (Natural Language Processing, NLP). Этот раздел изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез — генерацию грамотного текста.

Так как чат-бот является интеллектуальной системой, он содержит свою базу знаний. В простейшем случае, база знаний — это множество возможных вопросов пользователя и соответствующих ответов бота на них. Наиболее популярные методы выбора ответа — следующие:

- Полное совпадение фразы пользователя с фразой из базы знаний;
- Ключевые слова. После анализа текста программа выявляет ключевые слова и, основываясь на них, определяет более подходящий ответ;
- Анализ контекста. Также выбор ответа зависит от предыдущих ответов пользователя. Ответ на одну и ту же фразу может быть различным в разных частях диалога;
- Анализ других факторов. Чат-бот может обращаться в другие сервисы, чтобы выбрать более подходящий ответ. Например, бот может обратиться к социальным сетям, чтобы попытаться собрать более полную информацию о пользователе.

Чат-боты имеют множество преимуществ по сравнению с обычными людьми. Бот может параллельно общаться с большим количеством

пользователей одновременно, и, следовательно, пользователю не нужно ждать, когда освободится реальный человек. Чат-боту не нужно отдыхать, и он может общаться с пользователем 24 часа в сутки. Также виртуальный помощник может в разы быстрее анализировать вопросы пользователя и давать ответ, особенно когда для ответа нужны дополнительные вычисления или обращения к сторонним ресурсам.

Несмотря на все преимущества чат-ботов, у них также есть и существенные недостатки. На сегодняшний день искусственный интеллект еще не развит до уровня человека, и поэтому зачастую бот не может правильно понять пользователя и дать ему интересующий его ответ. Поэтому зачастую у современных чат-ботов есть возможность переводить диалог на реального человека, в случае если он не знает, как ответить пользователю.

Каналы общения

Общение пользователя с чат-ботом может происходить через различные каналы общения. Список самых популярных каналов общения:

- мессенджеры (Telegram, Viber, VK, WhatsApp и др.);
- сервисы онлайн консультирования (виджет на сайте или в приложении);
- социальные сети (Facebook, Вконтакте, Одноклассники и др.);
- голосовые ассистенты (Яндекс Алиса, Siri и др.);
- умные устройства (Яндекс станция, Amazon Echo, Google Home);
- телефония.

Сервис онлайн консультирования (онлайн консультант) — это сервис, который позволяет компаниям встраивать в свой сайт или мобильное приложение чат поддержки. Он может быть представлен в виде виджета или части интерфейса самого продукта.

Такие сервисы имеют множество преимуществ:

- возможность быстро дать посетителю помощь в текстовом виде, что избавляет пользователя от звонка по телефону;
- удобное приложение для операторов;
- сохранение данных о пользователях и истории переписки в базе;
- возможность оценки качества сервиса и диалога с оператором;
- настройка внешнего вида виджета;
- ведение статистики;
- интеграция с социальными сетями и мессенджерами.

Платформы для создания чат-ботов

Платформа для создания чат-ботов — это сервис, позволяющий просто и быстро создавать

Impact Factor:

ISRA (India) = 3.117
 ISI (Dubai, UAE) = 0.829
 GIF (Australia) = 0.564
 JIF = 1.500

SIS (USA) = 0.912
 РИНЦ (Russia) = 0.156
 ESJI (KZ) = 8.716
 SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
 PIF (India) = 1.940
 IBI (India) = 4.260
 OAJI (USA) = 0.350

чат-ботов различного уровня сложности. Зачастую, с помощью такого сервиса, для создания бота пользователь не должен обладать навыками программирования. Важной характеристикой хорошей платформы является возможность работать с различными каналами общения как с единым интерфейсом. Это означает, что пользователь может реализовать

одного бота и подключать его к разным каналам коммуникации.

Так как нашей целью являлось добавление новых каналов общения в платформу создания чат-ботов JAICP, мы провели сравнительный анализ различных платформ относительно поддерживаемых каналов общения. Результаты исследования можно посмотреть на рис. 1.

	JAICP	Dialogflow	RASA	Botkit	Amazon Lex	IBM Watson	LUIS + MBF	ChatFlow	Chatfuel	Pandorabots	ChattyPeople	Botsify	Beep Boop	Facebook MP	FlowXO	Smooch	botmother	Chatgun	Dexter	gupshup	
Веб-виджет	+	+	-	+	-	+	+	-	-	+	-	-	-	-	+	+	-	-	+	+	
Facebook Messenger	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	+
Alexa	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Skype	-	+	-	-	-	-	+	+	-	+	-	-	-	-	-	-	-	-	-	-	+
Slack	+	+	+	+	+	+	+	+	-	+	-	-	+	-	+	-	-	-	-	+	+
Telegram	+	+	+	-	-	-	+	+	-	+	-	-	-	-	+	+	+	-	-	-	+
Twitter	-	+	-	-	-	-	-	+	-	+	-	-	-	-	-	+	-	-	-	+	+
Viber	+	+	-	-	-	-	-	-	-	+	-	-	-	-	-	+	+	-	-	-	+
VK	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	-	-	+
WhatsApp	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	+	-	-	-	-	+
Instagram	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Google Assistant	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Google Hangouts	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+
Microsoft Cortana	-	+	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-
Yandex.Alisa	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Livetex	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Webim	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Zendesk Chat	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bitrix24	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Одноклассники	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-
SIP	+	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	+
Kik	-	+	-	-	+	-	+	+	-	+	-	-	-	-	-	-	-	-	-	-	+
Line	-	+	-	-	-	-	-	+	-	+	-	-	-	-	-	+	-	-	-	-	+
Twilio	-	+	+	+	+	-	-	+	-	+	-	-	-	-	+	-	-	-	-	+	-
Mattermost	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Rocket.Chat	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Cisco Spark	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Cisco Webex Teams	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+
Cisco Tropo	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Intercom	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Рис. 1. Сравнение поддерживаемых каналов у различных платформ

Выводом, полученным в результате этого сравнительного анализа, является то, что ни у одного из конкурентов нет поддержки сервисов Zendesk Chat, Webim и Bitrix24, интеграцию которых мы собрались осуществить.

Краткий обзор технологий для разработки.

Выбор технологий играет очень важную роль при разработке современных веб-приложений. Так, правильно подобранные технологии могут значительно сократить время на разработку и поддержку программного продукта. В этой главе мы сделаем обзор основных технологий, которые используются для разработки веб-приложений и которые потребуются для реализации интеграции выбранных нами сервисов онлайн

консультирования с платформой JAICP, и выберем наиболее подходящие для реализации нашей задачи.

Выбор языка программирования - очень важная задача. От этого выбора зависит весь остальной стек технологий. Существует большое количество языков, на которых можно написать веб-приложение, но в результате сравнения наш выбор остановился на языке программирования Java.

На чистом Java можно написать практически все что угодно, но современные фреймворки позволяют значительно сократить время и простоту разработки. На сегодняшний день существует множество Java фреймворков для создания веб-приложений. После анализа существующих фреймворков наш выбор

Impact Factor:

ISRA (India) = 3.117
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.156
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

остановился на Spring Framework (или коротко Spring) — универсальном фреймворке с открытым исходным кодом для Java-платформы. Spring — это семейство проектов, которые осуществляют конкретную функциональность. Благодаря Spring, разработчик приложений может пользоваться широким спектром инструментов с открытым кодом, не занимаясь написанием больших объемов кода и не привязывая создаваемое приложение слишком тесно к какому-то конкретному инструменту. Например, Spring Data обеспечивает взаимодействие приложения с данными: реляционными и NoSQL базами данных, key-value хранилищами и т.п., Spring Security помогает реализовать безопасность приложения и многое другое [4]. Spring Boot является проектом на уровне IO Execution (уровень выполнения) IO Spring Framework. Он позволяет намного легче создавать приложения на Spring, так как неявно выполняет многие конфигурации, которые требует Spring. Он не использует кодогенерацию, не использует XML для конфигурации, все конфигурируется через аннотации, но, если необходимо, можно с легкостью поменять конфигурацию по умолчанию для более специфической настройки [5]. Spring предлагает широкий спектр классов, предназначенных для поддержки создания веб-приложений. Благодаря Spring, разработчик получает максимальную гибкость при выборе способа реализации пользовательского интерфейса для веб-приложения.

Наиболее популярным шаблоном, который используется при разработке веб-приложений, является MVC. В последних версиях Spring постепенно развилась от простой веб-платформы до полноценной реализации MVC. Spring Web MVC — это оригинальный веб-фреймворк, построенный на API-интерфейсе Servlet и с самого начала включенный в Spring Framework. Он оперирует понятиями контроллеров, маппингов запросов, различными HTTP абстракциями и т.п.

По сравнению с другими фреймворками (например, с JSF) преимуществом Spring MVC является ориентированность на запросы. Это более четко реализует паттерн MVC. На наш взгляд, когда разработчик четко понимает, какие запросы и куда отправляются со стороны клиента, это приводит к более рациональному проектированию приложения и к более оптимизированному и быстрому коду. Серверный и клиентский код лучше разделяется.

В качестве инструмента сборки проекта мы выбрали Maven. По сравнению с другими инструментами для сборки приложений (например, Gradle) он более легок в освоении и является более популярным среди Java-разработчиков.

Технологии, необходимые для интеграции с сервисом Zendesk Chat

API сервиса Zendesk Chat задействует несколько необычных технологий для подобных сервисов, а именно протокол WebSocket и GraphQL [6].

WebSocket (стандарт RFC 6455) — это протокол связи поверх TCP-соединения, предназначенный для двунаправленного взаимодействия сервера и клиента в реальном времени. С помощью этого протокола можно отправлять сообщение на сервер и получать ответ без выполнения HTTP запроса, при этом этот процесс будет событийно-управляемым [7]. Протокол WebSocket состоит из двух частей: рукопожатие (handshake) между клиентом и сервером, а также последующая передача данных. После того, как клиент и сервер отправили свои рукопожатия, и в случае, если они прошли успешно, начинается передача данных. Это двусторонний канал связи, где каждая сторона может отправлять данные независимо от другой стороны. Клиент и сервер передают данные в концептуальных единицах, называемых сообщениями. Сообщение может состоять из одного или нескольких кадров (frames). В текущей версии протокола (RFC 6455) определено 6 типов кадров.

Зачастую поверх WebSocket используется протокол STOMP (Simple/Streaming Text-Oriented Messaging Protocol). STOMP — это простой протокол обмена текстовыми сообщениями. Основан на фреймах по образцу HTTP. Фрейм состоит из команды, необязательных заголовков и необязательного тела. Использование STOMP поверх WebSocket позволяет более удобно осуществить двунаправленное взаимодействие между сервером и клиентом.

Также необходимо помнить, что хоть протокол WebSocket и приносит новые возможности, он поддерживается далеко не всеми браузерами. С учетом того, что приложение должно продолжать работу с пользователями, в браузерах которых поддержка WebSocket отсутствует, понадобится задействовать какую-то запасную технологию для максимально близкой эмуляции желаемой функциональности.

Такой технологией, например, является SockJS. SockJS — это библиотека JavaScript, которая предоставляет объект, который имитирует WebSocket. SockJS предоставляет последовательный кросс-браузерный API Javascript, который создает полнодуплексный междоменный канал связи с низкой задержкой между браузером и веб-сервером, с WebSocket или без него, в случае если браузер не поддерживает WebSocket.

GraphQL — это язык запросов для API и серверная среда исполнения для выполнения этих

Impact Factor:

ISRA (India) = 3.117
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.156
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

запросов к вашим данным. Является проектом с открытым исходным кодом, разработанным компанией Facebook. Он создавался как более эффективная альтернатива REST, для разработки и использования программных интерфейсов приложений [8].

Эта технология имеет множество отличий от REST:

- Синтаксис GraphQL позволяет запрашивать именно те данные, которые необходимы, игнорируя ненужные;
- Синтаксис GraphQL является сильно типизированным. Это дает возможность до исполнения проверить правильность запроса в рамках системы типом этого синтаксиса;
- При работе с GraphQL необходим только один API endpoint - URL - для обращения к API.

Синтаксис GraphQL позволяет выполнять множество видов запросов. Стандарт GraphQL набирает популярность. В рамках опроса State of JavaScript, проведенного среди JS-разработчиков, более половины респондентов указали, что слышали об этой технологии и хотели бы с ней ознакомиться, а пятая часть уже её использует и не намерена отказываться. Если верить этому тренду, GraphQL ожидает активное развитие [9].

Технологии, необходимые для интеграции с сервисом Webim

Интерфейс взаимодействия сервиса Webim является типичным примером реализации REST API. REST (Representational State Transfer) — архитектурный стиль взаимодействия клиента (приложение/сайт) с сервером на основе протокола HTTP. Обычно все взаимодействие с сервером сводится к четырем операциям, которым соответствует конкретный HTTP метод [10]:

- 1) получение данных с сервера (GET-запрос);
- 2) добавление новых данных (POST-запрос);
- 3) обновление существующих данных (UPDATE-запрос);
- 4) удаление данных (DELETE-запрос).

Все необходимые данные передаются с помощью параметров запроса, либо, в случае POST и UPDATE запросов, с помощью тела запроса. Одной из особенностей REST является то, что сервер не запоминает состояние пользователя, т.е. необходимо в каждом запросе отправлять идентифицирующую информацию, посредством заголовков или параметров запроса.

Технологии, необходимые для интеграции с сервисом Bitrix24

Bitrix24, аналогично Webim, также реализует REST API для интеграции со сторонними сервисами. Для осуществления безопасности Bitrix24 использует протокол OAuth 2.0 [11]. OAuth - открытый протокол авторизации, который

позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль. Этот протокол является очень популярным для осуществления авторизации и его используют большое количество сервисов. Он позволяет клиенту получить доступ к API от имени конкретного пользователя конкретного портала. Для сервера авторизация - это указание на то, что пользователь дал доступ приложению, приложение предоставляет свой секрет. Портал это всё объединяет и выдает приложению соответствующий тип доступа.

Некоторые детали программной реализации

Общим для всех интеграций является интерфейс ConfigurableChatService, который содержит общие методы для управления ботами и каналами. Этот интерфейс содержит такие методы как processHuman(String token, T request), который обрабатывает сообщение пользователя, написанное в каком-либо канале связи, addBot(String accountId, ChannelChatConfig config) который вызывается при создании канала, и так далее. Также общим для всех каналов является абстрактный класс BaseAsyncAdapterService, который реализует интерфейс ConfigurableChatService. Для каждого канала будет реализован класс <имя_канала>ServiceImpl, который будет определять необходимые методы этого интерфейса и абстрактного класса. Это позволит при работе с каналами в других частях платформы не завязываться на конкретной реализации.

Первым шагом для реализации интеграции с сервисом Zendesk Chat является создание классов моделей данных. API Zendesk Chat принимает и отправляет данные в формате JSON. Spring Framework включает в себя библиотеку Jackson, которая позволяет преобразовывать текст в формате JSON в Java объекты и обратно. Правила преобразования можно указать с помощью аннотаций. В большинстве случаев, это преобразование Spring может осуществлять неявно, но, когда нужно выполнить преобразование вручную, можно воспользоваться методами класса ObjectMapper. Также для удобства в своем проекте мы использовали библиотеку Lombok. Она позволяет не писать конструкторы классов, методы получения и установки полей, а также другие методы. Достаточно указать соответствующие аннотации, и тогда, в процессе компиляции, библиотека сгенерирует необходимые методы автоматически. Например, аннотация @AllArgsConstructor создает конструктор класса, в котором аргументами являются все его поля. Для работы с

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	РИИЦ (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

протоколом WebSocket мы использовали библиотеку okhttp3, с помощью которой можно удобно установить двунаправленное соединение с сервером Zendesk Chat. В данном случае такое соединение используется для того, чтобы сервер мог сразу же уведомлять нас о новом сообщении пользователя или других событиях.

Аналогично реализациям других каналов, для Webim для удобного взаимодействия с данными мы создали классы модели. Альтернативой использования протокола WebSocket для получения входящих сообщений от сервера является указание на нем webhook URL. При возникновении какого-либо события, например, при новом сообщении пользователя, сервер будет отправлять запрос на этот URL со всей необходимой информацией. Также одной из задач, связанной с интеграцией сервиса Webim, был сбор логов диалога пользователя с оператором. Суть в том, что изначально пользователь переписывается с ботом, и все сообщения записываются в базу данных платформы. В случае, если бот не может ответить на сообщение пользователя, к диалогу подключается реальный оператор, и Webim перестает отправлять уведомление о новом сообщении на webhook URL для этого диалога. Для того, чтобы можно было просматривать весь диалог с конкретным пользователем, мы реализовали отдельный сервис, который периодически обращается к Webim и сохраняет все недостающие записи логов.

Аналогично Zendesk Chat и Webim для интеграции с сервисом Bitrix24 [2] был создан

класс Bitrix со всеми классами моделями для данных и с различными константами. Аналогично Webim, о всех событиях сервер Bitrix24 объявляет с помощью запросов, отправленных на заданный webhook url. При входящем сообщении вызывается метод processHuman, который преобразует запрос в унифицированный для всех каналов вид и отправляет этот запрос в другой компонент для получения ответа бота на этот запрос. При получении ответа вызывается метод processBot, в котором этот ответ уже преобразуется в вид, специфичный для Bitrix24, и, используя объект класса BitrixApi, вызывается нужный метод.

Анализ результатов и тестирование

Для того, чтобы пользователь мог удобно подключать реализованные мной каналы общения, мы добавили в интерфейс платформы Just AI Conversation Platform новые элементы.

Первым делом в окно создания каналов были добавлены три новые кнопки с названиями сервисов и их логотипами (рис. 2).

Вторым шагом было создание для каждого канала общения индивидуальной формы. Эти формы позволяют указать необходимые данные, чтобы было возможно подключить канал. Кроме специфичных для канала полей также было необходимо добавить в формы общие для всех каналов данные, такие как название канала, ветка в репозитории со сценарием, а также режим деплоя - автоматический или нет.

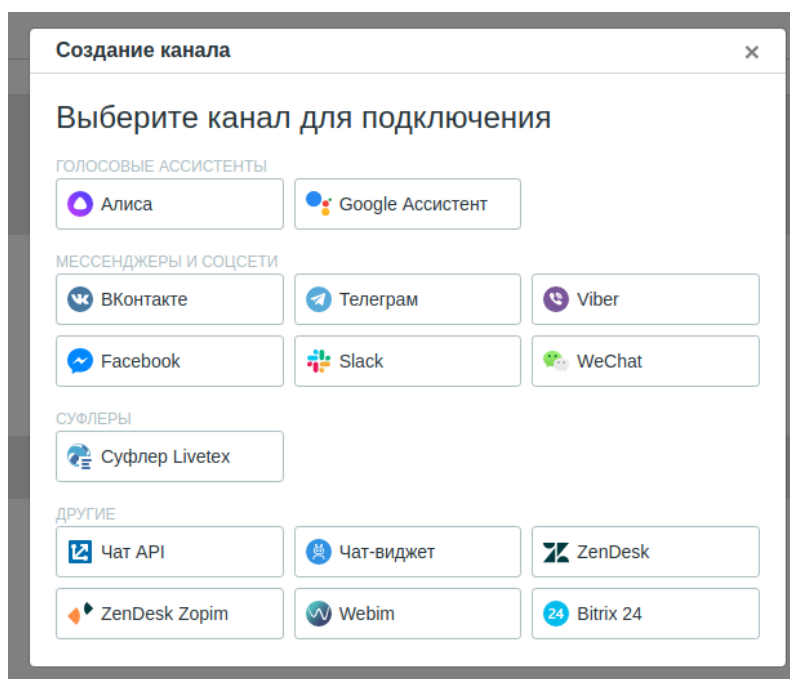


Рис. 2. Окно добавления канала

Impact Factor:

ISRA (India) = 3.117
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.156
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

Для канала Zendesk Chat от пользователя необходимо только одно поле - токен доступа. Форму создания этого канала можно увидеть на рис. 3.

Для сервиса Webim, кроме токена доступа к API, необходимо также указать домен аккаунта.

Для того, чтобы осуществить сбор логов между пользователем и оператором, необходимо указать электронную почту и пароль администратора. Эти два параметра не являются обязательными. Форму создания канала Webim можно увидеть на рис. 4.

Рис. 3. Окно добавления канала Zendesk Chat

Рис. 4. Окно добавления канала Webim

Bitrix24 имеет очень гибкую настройку для создания ботов. Для них можно указать имя, фамилию, отчество, год рождения, аватар и многое другое. Для более дружелюбного интерфейса было решено оставить настройку только основных параметров бота, таких как имя, фамилия, электронную почту и аватар бота. Для того, чтобы обращаться к API сервиса,

пользователю также необходимо указать код и ключ приложения. Внешний вид получившейся формы можно увидеть на рис. 5.

Для того, чтобы можно было проверить реализованные интеграции, мы решили написать в платформе JAICP тестового бота, который будет представлять собой игру в города. К этому боту

Impact Factor:

ISRA (India) = 3.117
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.156
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

мы подключим реализованные нами каналы и проверим корректность их работы.

Платформа Just AI Conversation Platform имеет свой язык для написания сценариев работы бота. Он имеет огромный функционал, и с помощью него можно реализовать сценарий

любой сложности. В сценарии объявляются состояния, в которых может находиться бот в каждый момент времени, и в зависимости от состояния выполняется конкретная логика.

Создание канала

24 Bitrix 24

Название: Bitrix 24

Токен: Введите токен

Имя: Имя

Фамилия: Фамилия

Email: Email

Открытые линии: Открытые линии

Код приложения: Код приложения

Ключ приложения: Ключ приложения

Аватар бота: Аватар бота

Аватар бота: [Upload icon]

Ветка: master

Публикация: Автоматически Вручную

Назад Отмена Создать

Рис. 5. Окно добавления канала Bitrix24

В сервисе Zendesk Chat (Zopim) была создана учетная запись, получен токен доступа для возможности подключения бота и скопирован скрипт виджета сервиса, который можно вставить на свою страницу. Результат диалога пользователя и бота можно увидеть на рис. 6.

В этом канале общения есть возможность переводить диалог на реального оператора.

Процесс подключения бота для Bitrix24 немного отличается. В отличие от Zendesk Chat первым делом канал создается в платформе JAICP. После создания в платформе генерируется webhook адрес, который уже необходимо вставить в Bitrix24. После установления этого адреса бот становится доступным. На рис. 7 представлен процесс общения пользователя с ботом, используя канал Bitrix24.

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	РИИЦ (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

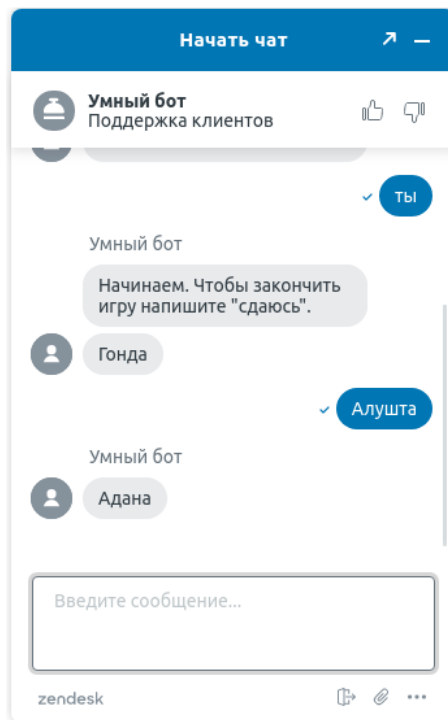


Рис. 6. Виджет сервиса Zendesk Chat

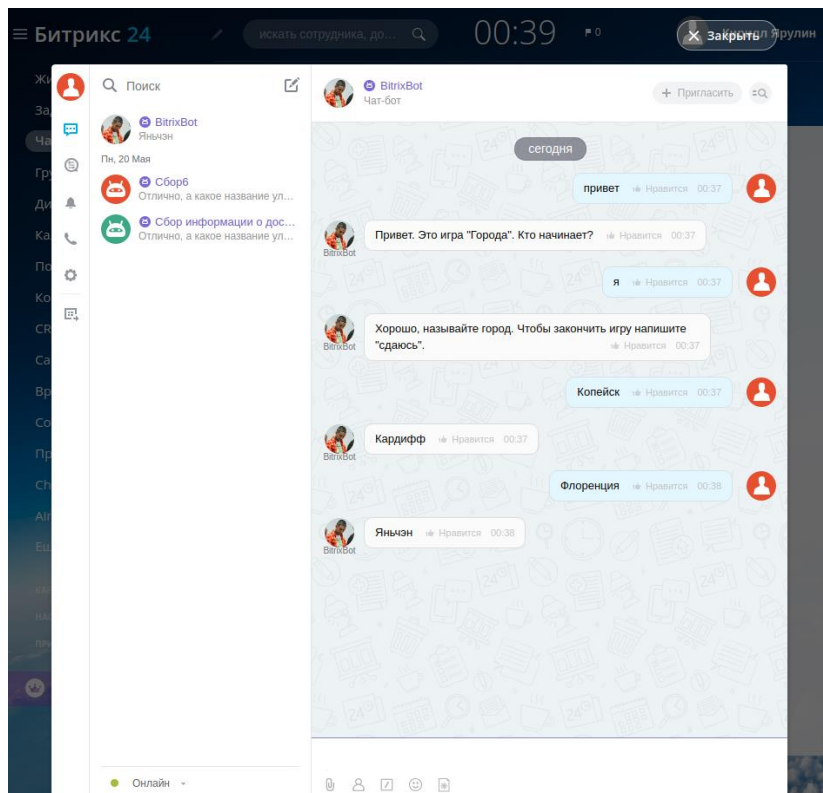


Рис. 7. Чат в сервисе Bitrix24

По сравнению с Zendesk Chat и Bitrix24, проверку интеграции с сервисом Webim пришлось проводить другим способом. Сложность оказалась в том, что в Webim невозможно создать бесплатный тестовый аккаунт. Исходя из этого,

для проверки нашей реализации, мы написали тесты, которые проверяют, что на каждый возможный запрос от Webim в ответ отправляется корректный JSON.

Impact Factor:

ISRA (India) = 3.117	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.156	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

Так как для получения ответа бота активно используется другой компонент, для тестирования функционала сервисов пришлось написать mock класс, который будет имитировать работу реального компонента.

Чтобы в тестах можно было проверить, какие именно данные попадают во внешний компонент, мы создали класс `TestEventConsumer` с пустым методом `process`, который вызывается в классе `EchoBotServerConnector`. Далее в наших тестах с помощью библиотеки `Mockito` мы проверяем, с какими именно данными вызывался метод `process`.

В каждом тесте с помощью объекта `mockMvc` выполняется запрос на `webhook` адрес канала и проверяется статус и тело ответа.

Заключение

В результате работы была успешно осуществлена интеграция сервисов онлайн

консультирования `Zendesk Chat` (`Zopim`), `Webim` и `Bitrix24` с платформой создания чат-ботов `Just AI Conversation Platform` (`JAICP`). Также, кроме реализации серверной части, были добавлены формы создания новых каналов общения в интерфейсе платформы `JAICP`.

Чтобы проверить корректность работы интеграции сервиса `Webim`, были написаны тесты, которые проверяют, что на каждый возможный запрос от `Webim` возвращается верный ответ. Так как для сервисов `Zendesk Chat` (`Zopim`) и `Bitrix24`, в силу сложности их API, написание тестов оказалось затруднительно, для тестирования работы этих каналов было принято другое решение. В платформе `JAICP` был написан тестовый сценарий бота, который представляет собой игру в "Города". Далее к нему были подключены каналы `Zendesk Chat` и `Bitrix24`, и осуществлено ручное тестирование.

References:

- (n.d.). *Chatbot*. [Online]. Retrieved 01.07.2019 from <https://en.wikipedia.org/wiki/Chatbot>
- (n.d.). *80% of businesses want chatbots by 2020*. [Online]. Retrieved 01.07.2019 from <https://www.businessinsider.de/80-of-businesses-want-chatbots-by-2020-2016-12?r=US&IR=T>
- (n.d.). *Rynek chat-botow v tsifrakh i faktakh. Infografika*. [Online]. Retrieved 01.07.2019 from <https://www.plusworld.ru/daily/tehnologii/403076-2/>
- Cosmina, I., Harrop, R., Schaefer, C., & Ho, C. (2017). *Pro Spring 5. An In-Depth Guide to the Spring Framework and Its Tools*. (p.849). Apress.
- (n.d.). *Spring Boot Reference Guide*. [Online]. Retrieved 01.07.2019 from <https://docs.spring.io/spring-boot/docs/current/reference/html/>
- (n.d.). *Zendesk Chat API Documentation*. [Online]. Retrieved 01.07.2019 from https://developer.zendesk.com/rest_api/docs/chat/introduction
- (n.d.). *RFC 6455 - The WebSocket Protocol*. [Online]. Retrieved 01.07.2019 from <https://tools.ietf.org/html/rfc6455>
- (n.d.). *GraphQL Documentation*. [Online]. Retrieved 01.07.2019 from <https://graphql.org/learn/>
- Maldonado, L. (n.d.). *A Beginner's Guide to GraphQL*. [Online]. Retrieved 01.07.2019 from <https://www.freecodecamp.org/news/a-beginners-guide-to-graphql-86f849ce1bec/>
- (n.d.). *Representational state transfer*. [Online]. Retrieved 01.07.2019 from https://en.wikipedia.org/wiki/Representational_state_transfer
- (n.d.). *Bot platforma Bitrix24*. [Online]. Retrieved 01.07.2019 from https://dev.1c-bitrix.ru/learning/course/?COURSE_ID=93&INDEX=Y