

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

РЕФЕРАТ

Тема: Описание предполагаемого способа решения

Студентка гр. 4303

Попова Д. Е.

Преподаватель

Санкт-Петербург

2019

1. ОПИСАНИЕ АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ

1.1. Описание используемых технологий

Разрабатываемый инструмент представляет собой веб-сайт. В качестве языка программирования был выбран PHP, потому что основная CRM — 1С-Битрикс.

1.1.1. 1С-Битрикс

Платформа предлагает некоторые решения интернет магазинов, в частности недвижимости. Здесь продуман весь путь клиента от просмотра товара до его покупки, а стандартные компоненты платформы обеспечивают высокую конверсию в продажи.

Корзина Битрикса пересчитывает стоимость в режиме реального времени, имеет возможность отложить товары без удаления, повторить заказ в личном кабинете, а также выполнить поиск по корзине. Одностраничное оформление улучшает пользовательский опыт, а количество товаров не влияет на скорость работы корзины.

Платформа адаптируется под все устройства с учётом принципа mobile first.

Для хранения данных организована БД при помощи Битрикс API, где имеется поддержка SQL.

2.1.2. SOAP — протокол обмена структурированными сообщениями в распределённой вычислительной среде. Он используется для обмена произвольными сообщениями в формате XML. SOAP является расширением протокола XML-RPC.

SOAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTP, HTTPS и др. Однако его взаимодействие с каждым из этих протоколов имеет свои особенности, которые должны быть определены отдельно. Чаще всего SOAP используется поверх HTTP.

В данном проекте SOAP используется для передачи xml данных в 1С.

1.1.3. RESTful API

REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы. В определённых случаях (интернет-магазины, поисковые системы, прочие системы, основанные на данных) это приводит к повышению производительности и упрощению архитектуры. В широком смысле компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов. Вызов удалённой процедуры может представлять собой обычный HTTP-запрос (обычно «GET» или «POST»; такой запрос называют «REST-запрос»), а необходимые данные передаются в качестве параметров запроса.

Несмотря на то, что REST не является стандартом сам по себе, большинство RESTful-реализаций используют такие стандарты, как HTTP, URL, JSON и XML.

При разработке данного сервиса был использован REST для общения сервиса KVADO — комплексное решение для организаций ЖКХ и состоящее из нескольких функциональных модулей, каждый из которых решает определённый круг задач — с KVS.

1.1.4. RabbitMQ

Сервер RabbitMQ по сути является менеджером очередей, который имеет следующие преимущества:

- в случае некорректного завершения работы сервера, данные в очереди не теряются. И при последующем запуске обработка продолжается с того места, где был обрыв;
- распределить задачи на несколько очередей, т.е. создать распараллеливание на уровне сообщений
- если результат обработки не удовлетворяет, задачу можно послать в очередь повторно;

- существует несколько режимов работы очереди: рассылка типа точка-точка(*direct*), рассылка сообщений по шаблону(*topic*), широковещательная рассылка сообщений(*fanout*);
- возможность синхронизировать работу клиента и сервера, своего рода реализация RPC
- количество хранимых в очереди сообщений неограничено
- сервер сообщений может быть расположен удаленно как по отношению к продюсеру, так и по отношению к консьюмеру.

RabbitMQ использует протокол AMQP. Чтобы использовать RabbitMQ необходимо поставить клиентскую и серверную части.

1.1.5. Логгирование в Slack

Для проверки целостности полученных и отправленных данных был реализован бот для Slack, отправляющий сообщения в чат при помощи POST-запросов. В настройках бота имеется раздел *Add features and functionality*:

Incoming Webhooks — позволяет настроить отправку сообщений в чат через специальный адрес

Interactive Messages — используется для добавления кнопок в сообщениях

Slash Commands — в этом разделе можно задать адреса для обработки специальных команд вида «/command»

Event Subscriptions — позволяет подписать ваше приложение на события. Например, «пользователь покинул чат» или «ответ/реакция на сообщение»

Bots — раздел для создания бота, который будет получать уведомления о событиях, активированных в «Event Subscriptions»

Permissions — в этом разделе задается список привилегий, которые будет запрашивать приложение при установке (добавлении в команду).

В разделе «Install your app to your team» настраивается авторизация приложения через протокол OAuth. Раздел «Manage Distribution» используется

для того, чтобы ваше приложение попало в общий каталог, где его сможет найти любой пользователь.

Webhook — адрес, на который можно слать запросы для отправки сообщений в чат от лица приложения и который непосредственно логирует информацию о принятых и отправленных данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Основные сведения // Документация для разработчиков URL:
https://dev.1c-bitrix.ru/api_help/ (дата обращения: 21.10.2019).
2. RabbitMQ для начинающих // Ajaxblog URL:
<http://ajaxblog.ru/php/rabbitmq-tutorial/> (дата обращения: 03.11.2019).
3. Разработка под Slack: отправляем сообщение в чат // CodeX URL:
<http://ajaxblog.ru/php/rabbitmq-tutorial/> (дата обращения: 07.11.2019).
4. SOAP // Wikipedia URL: <https://ru.wikipedia.org/wiki/SOAP> (дата обращения: 07.11.2019).
5. REST // Wikipedia URL: <https://ru.wikipedia.org/wiki/REST> (дата обращения: 07.11.2019).
6. КВАДО.РУ - простое решение для сложных задач // КВАДО.РУ URL:
<http://kvado.ru/> (дата обращения: 29.10.2019).