

Walros Rolz Notes

Содержание.

1. [Введение](#)
2. [Обзор текущих возможностей бота и платформы.](#)
 1. [Текущие возможности бота.](#)
 2. [Discord.](#)
 3. [Сравнение с другими ботами.](#)
3. [Обзор нового функционала](#)
4. [Работа с markdown.](#)
5. [Стадии разработки.](#)
 1. [Front-end.](#)
 2. [REST-Api.](#)
 3. [Расширение функционала.](#)

Введение

Обсуждение этого момента стоит начать с введения контекста.

Настольные игры, а также ролевые игры типа PnP (Pen and Paper) достаточно популярны по всему миру, однако до сих пор универсальные решения для реализации подобных игр онлайн, для людей разделенных расстояниями, оставляют желать лучшего.

На текущий момент большая часть игроков, особенно в ролевые игры, предпочитает играть "по старинке" в чатовых клиентах, с какой-либо базовой репрезентацией доски. И если ещё примерно пять-десять лет назад наиболее популярны были IRC и jabber чаты, то сейчас люди предпочитают Discord по ряду причин.

В конце 2017 года был написан простенький бот специально для настольных игр. На текущий момент бот поддерживает много разнообразных бросков костей, однако существует набор функционала, не реализованный ни в одном из аналогов.

Функционал Walross Rolz Notes -- часть этого набора, заключающаяся в возможности добавление внутрь бота своих записок в живом режиме. Для игр это даст быстрый доступ к правилам для всех играющих, возможность загружать листы персонажей, заметки и прочие дополнительные материалы.

Обзор текущих возможностей бота и платформы.

Обзор текущих возможностей.

Основная идея бота при его создании заключалась в том, чтобы предоставить возможности для бросков разного вида кубиков, а также трекинга статусов в игре.

На текущий момент бот поддерживает функционал rolz.org, одного из самых больших online-роллеров, а также поддерживает несколько специфичных типов кубов, таких как NWoD-ролл (кубики специфичные для серии игр New World of Darkness от издателя White Wolf) и Warhammer Rolls(кубики, специфичные для ролевых игр по вселенной Warhammer 40k от Fantasy Flight Games).

Для трекинга статусов используется команда value, позволяющая быстро добавить значение и легко за ним следить.

На данный момент какого-либо продвижения бота на платформе не производилось, в связи с чем цифры более чем скромные. Однако, после добавление функционала планируется включения бота в большую часть бот-треккеров.

Discord.

В данном разделе мы рассмотрим почему используется именно платформа Discord. У данной платформы есть множество конкурентов, таких как Telegram, WhatsApp, Slack, Skype. Однако, большая часть нашей аудитории сосредоточена именно тут.

Так вышло, что набор функционала конкретно этой платформы практически идеален для игры в настольные игры. По пунктам можно выделить следующий функционал, сыгравший ключевую роль в выборе платформы:

1. Архитектура платформы. Discord использует всем крайне знакомую архитектуру, напоминающую IRC-серверы. Структурно пространства для чата делятся на "Серверы" и "Каналы". Сервер представляет собой набор каналов для чата, с простым и понятным управлением ролями, значительно упрощающими модерацию. Канал же, по сути своей является стандартной чатовой "комнатой". Итого архитектурно пользователь получает все плюсы IRC, без огромнейшего минуса, состоящего в необходимости самостоятельно администрировать IRC-сервер.
2. Клиент Discord. Клиент разрабатывается полным циклом, и соответственно доступен на любой платформе, плюс работает из браузера. Также, клиент достаточно стабилен и удобен, имея при этом достаточно стандартные характеристики по потреблению ресурсов (приблизительно равные одной вкладке в браузере, чем по сути он и является).
3. Вечное логирование. За всё время существования, сервис Discord ни разу не потерял логи общения клиентов. Возможно для рядового пользователя это кажется маленькой деталью, однако для сообщества игроков в настольные игры эта деталь чрезвычайно важна.
4. Возможности для голосового и видео-общения. Они есть, и выполнены на должном уровне. Такие возможности также крайне важны для игроков в настольные игры, и их отсутствие заставило бы сообщество выбрать другой клиент.

Сравнение с другими ботами.

В качестве сравнения рассмотрим один из популярнейших ботов для платформы Discord, не имеющий какой-то определенной ориентации, а также возьмем узко-специального бота с аналогичной нашему специальностью.

[Mirai](#)

Mirai -- универсальный бот, написанный в качестве учебного проекта студентом первого курса под ником brussell. Бот предлагает крайне обширный функционал, от проигрывания музыки и до базовых бросков кубиков.

Также, по нему предоставляется крайне обширная статистика из первоисточника. На текущий момент ботом пользуются более миллиона пользователей более чем на сорока тысячах серверов. Прямая ссылка на статистику: <https://mirai.brussell.me/stats>

Популярность бота вызвана обширностью его функционала, а монетизация производится путем продажи дополнительного функционала.

Можно сказать, что этот бот является идеальным универсальным ботом, но в нем отсутствуют релевантные функции для целевой аудитории нашего бота.

RPBot

RPBot -- один из самых популярных ботов для PnP-ролевых игр на платформе. Используется более чем на двадцати тысячах серверов. Не смотря на крайне скромный набор функционала касательно броска кубиков, получил большую популярность за свою систему работы с менеджментом персонажей.

Однако даже сейчас, этот бот проигрывает Walros Rolz в плане функционала, поскольку весь его функционал реализован, но сверху имеется ещё множество другого.

Обзор нового функционала

Новый функционал заключается в возможности добавления, редактирования и вывода в чат собственных "записок". Этот функционал позволит игроку прямо в чате выводить всю нужную информацию для игры. Образцами такой информации являются:

- Наборы правил.
- Записки и наброски о сюжете игры.
- Листы персонажей.

Причиной добавления такого функционала стало прямое общение с пользовательской базой. На текущий момент такой функционал является самым востребованным. На втором месте из глобальных предложений стоит поддержка макросов и переменных. На третьем же добавление нескольких специализированных кубиков поверх текущего набора функционала.

Новый функционал состоит из двух частей:

1. Сайт для работы с собственными записками в живом режиме.
2. Индексный поиск по содержанию документа.

Сайту нужна система менеджмента записок, позволяющая просто и быстро превращать набор правил в PDF в записку на markdown. Для авторизации на сайте будет использоваться непосредственно Discord-авторизация. Таким образом записки можно будет легко привязать к пользователю.

Индексный поиск же подразумевает вывод текста чатовой командой, например: "!notes search DnD3.5::Character Creation"

Искать пользователь, во всяком случае в изначальном варианте функционала, сможет исключительно по своим запискам.

Работа с Markdown.

Markdown -- типографический язык разметки. Один из самых удачных для своей цели на мой взгляд, например этот документ был отформатирован на нем.

Одним из самых больших плюсов его для данной задачи является тот факт, что документы в форматах PDF и HTML крайне легко конвертируются в Markdown, а также тот факт, что язык легко разбивается на заголовки для поиска, выстраивая тем самым содержание-индекс.

Также, стоит отметить, что сами документы после конвертации легко редактировать, и очень легко писать собственные заметки, а реализация markdown редакторов на JS в браузере вполне себе достойная.

Стадии разработки.

Разделить этот проект можно на три больших части, которые также являются стадиями его разработки.

1. Front-end, что есть разработка сайта.
2. Back-end, что есть разработка RESTful API для этого сайта.
3. Расширение непосредственно функционала бота.

Front-end.

В качестве решения для написания Front-end части был выбран проект Google -- Angular. Основной причиной выбора именно этого фреймворка является наиболее удачный и современный инструментарий этого проекта.

Если сравнивать удачные Front-end решения, то можно прийти к сравнению трех фреймворков:

- React
- Vue
- Angular

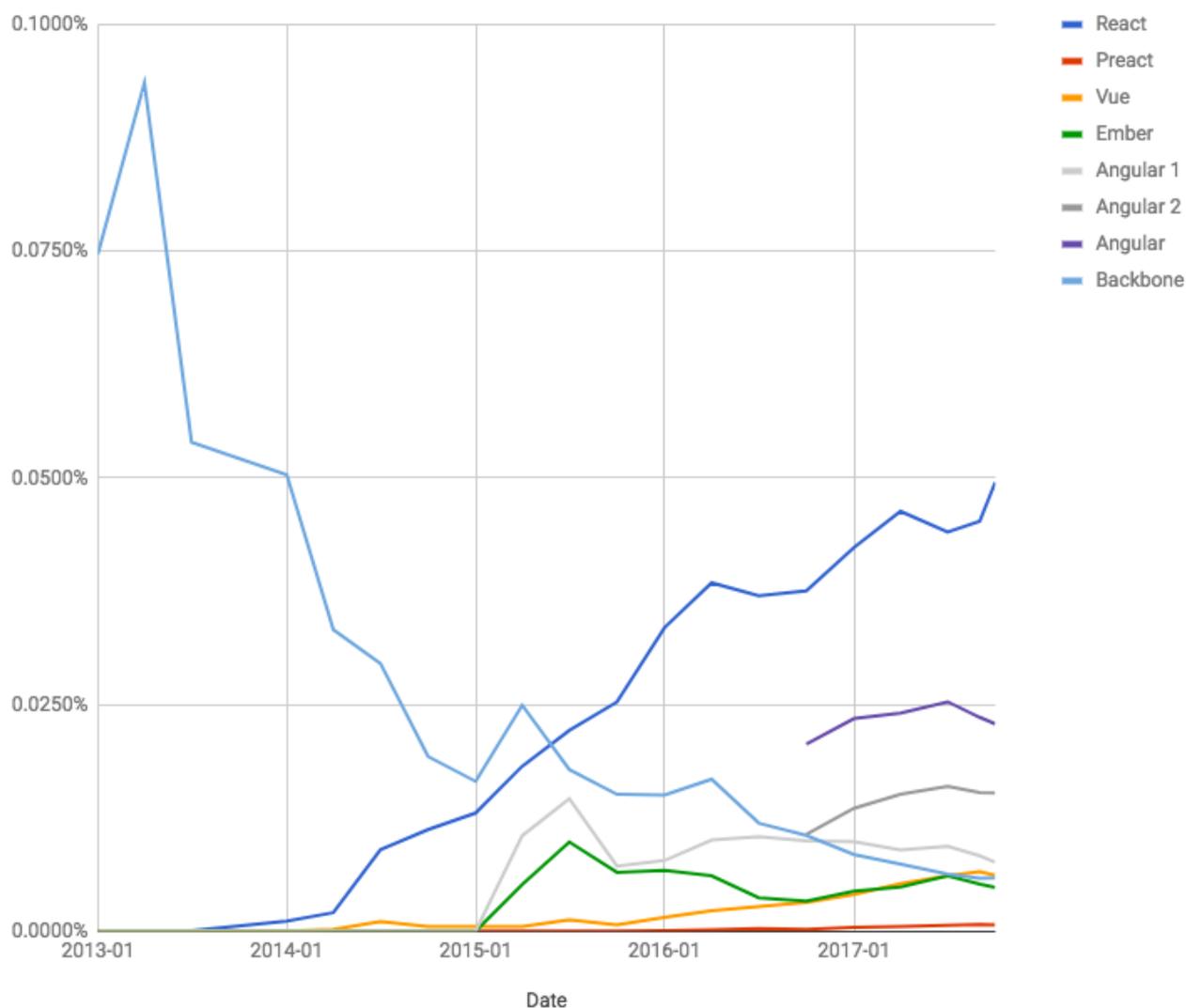
Из трех React является старейшим, вероятнее всего наиболее популярным, хотя статистика из разных источников и может расходиться по глупой причине -- часто в Angular по статистике считается AngularJS. Однако, фреймворк React имеет достаточно много архаизмов, и в принципе тихонечко устаревает.

Vue же, в свою очередь, имеет крайне скудный инструментарий. При чем в основном Китайский, как следствие с не самой понятной документацией. При том, что фреймворк развивающийся его подводит отсутствие инструментария.

Angular при этом сочетает лучшие качества обоих своих конкурентов -- он имеет крайне обширный инструментарий, во многом написанный Google, но при этом развивается, и крайне быстро.

Статистика от npm:

Major FE frameworks, share of registry



Back-end.

Back-end части приложения будет представлять собой RESTful-API, которой будут передаваться данные в формате JSON при помощи XHR-запросов. Со стороны API будет происходить сериализация разметки в Markdown и загрузка её в базу на MongoDB.

Для этой задачи прекрасно подойдет язык Go, в сочетании с технологией gRPC. Связано это с тем, что:

1. gRPC является нативным инструментом Golang и прекрасно подходит для выполнения задач построения RESTful-API, всерьез упрощая маршalling данных.
2. Golang является современным статически-типизированным языком, заточенным для веб-разработки. Мы получаем всю скорость C, при этом всё удобство Python.
3. Знаний команды в области других подобных языков, конкурирующих с Go -- Crystal и RUST недостаточно для реализации данной задачи.

Расширение функционала.

Сам бот написан на Python 3, и для него требуется расширение, работающее с данными в MongoDB в соответствии с нашим форматом документа. Эта фаза является конечной.

Техническое задание.

Front-end.

Функционал сайта должен включать в себя следующие страницы:

Home -- Основная, домашняя страница сайта. Содержит основную информацию по функционалу бота, а также демо.

About -- Страница, содержащая информацию о авторах бота, их контакты, а такж основные контакты проекта.

Commands -- Страница, описывающая детально все команды, присутствующие у бота.

Login -- Ссылка на форму OAuth2 авторизации дл бота.

Notes -- Страница, содержащая систему менеджмента записок, только для авторизованных пользователей. Также, на этой странице, при переходе на записку открывается Markdown редактор для записок.

Внешний вид можно посмотреть по ссылке на интерактивный макет:

<http://dev.space-walrus.ml>

Back-end.

Функционал RESTful API должен содержать в себе менеджмент OAuth2 авторизации пользователя в соответствии с требованиями Discord, а также работу с приемом данных в markdown в виде строки в одном из полей JSON.

Обработка состоит в превращении строки в Mongo-документ, разбитый структурно по заголовкам. Для вывода обратно в редактор на сайте необходима возможность десериализации данных в обычную Markdown-строку без потери внутренней структуры.

Такая работа создает достаточно большую нагрузку, из-за чего следует тот факт, что писать необходимо в мультипоточном стиле. Избегать использование lock-синхронизации, в связи с сложностью тестирования. Потоки синхронизировать каналами.

Заключение.

Подведем итоги.

Суть задачи -- добавление функционала по обработке markdown в бота Walross Rolz.

Цель разработки расширения -- помочь комьюнити PnP-игроков, собственное пользование, расширение имеющейся платформы, ну и в последнюю очередь возможная монетизация.

Средства разработки -- Python, Golang, TypeScript. Фреймворки -- Angular, discord-py, beego. Форматы данных, используемые в проекте -- JSON, Markdown. Основная СУБД проекта -- MongoDB.