

УДК 004

СЕРВИС ОБМЕНА МГНОВЕННЫМИ СООБЩЕНИЯМИ В РАМКАХ ИНФОРМАЦИОННОЙ СРЕДЫ ОРГАНИЗАЦИИ

Т.А. Решетова, студент гр. ИТб-111, IV курс

Научный руководитель: И.С. Сыркин, к.т.н., доцент

Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

Актуальность

В любой организации существуют свои способы обмена информацией между сотрудниками.

На данный момент в организации существует свой Информационный портал, к которому имеют доступ сотрудники и другие авторизированные пользователи. Используя Портал, сотрудники удаленно могут совершать различные действия с информационными ресурсами организации, однако для связи сотрудников друг с другом и передачи между ними информации используются либо телефонная линия, либо дополнительные программные средства.

Исходя из этого, было решено создать дополнительный сервис позволяющий сотрудникам обмениваться текстовыми сообщениями внутри уже имеющегося информационного портала организации.

Следовательно, цели разработки сервиса:

- Обеспечить сотрудников организации возможностью обмена информацией внутри информационной среды организации.
- Ускорить процесс обмена информацией
- Упростить работу сотрудников в системе

Задачи:

- Разработать сервис обмена сообщениями
- Внедрить сервис в информационную среду организации

Используемые технологии

Для решения задач будут применяться следующие технологии:

1. Паттерн HMVC
2. KONGA
3. Node.js
4. Bootstrap

1. Паттерн HMVC

Паттерн Иерархические-Модель-Вид-Контроллер (HMVC) является расширением MVC, которое позволяет решить множество проблем масштабируемости.

MVC - Шаблон MVC (Модель-Вид-Контроллер или Модель-Состояние-Поведение) описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и конечно же реализуется.

Модель — содержит бизнес-логику приложения и включает методы выборки (это могут быть методы ORM), обработки (например, правила валидации) и предоставления конкретных данных, что зачастую делает ее очень толстой, что вполне нормально.

- Служит для прямого обращения к БД.
- Не взаимодействует с пользователем напрямую, только через контроллер.
- Не генерирует HTML или другой код отображения.
- Одна и та же модель может использоваться в разной части приложения.

Вид — используется для задания внешнего отображения данных, полученных из контроллера и модели.

- Содержит HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных.
- Не обращается к базе данных.
- Не работает с данными, полученными из запроса пользователя.
- Может напрямую обращаться к свойствам и методам контроллера или моделей, для получения готовых к выводу данных.

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение.

- Отвечает за обработку запросов пользователя.
- Не должен содержать SQL-запросов. Их лучше держать в моделях.
- Не должен содержать HTML и другой разметки. Её стоит выносить в виды.

HMVC – это набор традиционных триад MVC, работающих как одно приложение. Каждая триада совершенно независима и может выполняться при отсутствии любой другой. Все запросы к триадам должны использовать интерфейс контроллера (рис. 1).

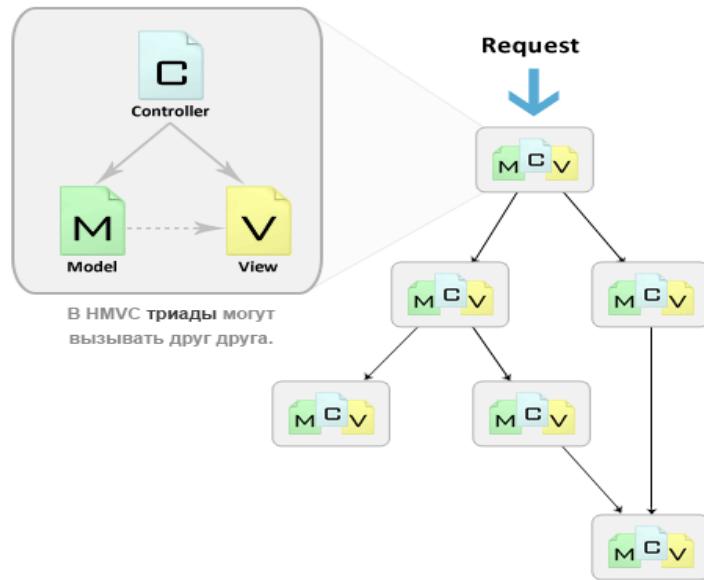


Рисунок 1 - Схема HMVC.

Физическое местонахождение триады на сервере не имеет значения, главное, чтобы она была доступна из всех других частей системы. Характерными особенностями HMVC являются поощрение повторного использования кода, упрощение тестирования отдельных частей системы и гарантия того, что приложение может быть усложнено и расширено без особых сложностей.

Успешное проектирование приложения на основе паттерна HMVC невозможно без разбиения его функций на отдельные системы. Каждая такая система должна представлять собой одну триаду MVC, независимо управляющую методами хранения и представлением внутри более крупного HMVC приложения. В настоящее время существует несколько фреймворков, поддерживающих HMVC без дополнительных модулей. Один из них – Kohana, который изначально создавался с ориентацией на HMVC.

2. Kohana framework.

Kohana (ранее Blue Flame) — это веб-фреймворк с открытым кодом, основанный на PHP5 и использующий концепцию HMVC

Особенности:

- Высокая скорость работы
- Безопасность
- Использование возможностей PHP5
- Большое количество встроенных инструментов
- Простота понимания
- Использование концепции HMVC
- Полная совместимость с UTF-8

3. Node.js

Node или **Node.js** — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код).

Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода.

Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения и даже программировать микроконтроллеры.

В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

В рамках разработки Node будет применяться в качестве web-сервера для обработки запросов пользователей (рис. 2).

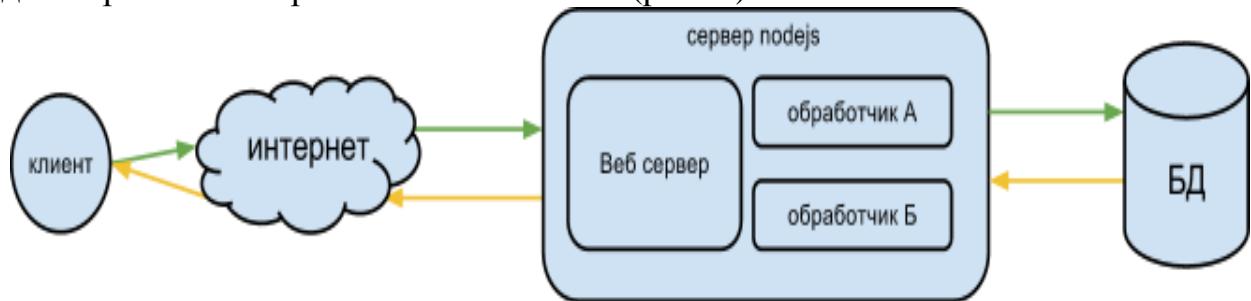


Рисунок 2 – Упрощенная схема работы сервера Node.

Главным преимуществом Node является использование асинхронной модели построенной на очереди событий, т.е. при возникновении некоторого события (пришел запрос, выполнилось считывание файла, пришел ответ от БД) оно помещается в конец очереди. Поток, который обрабатывает эту очередь, берет событие с начала очереди, и выполняет связанный с этим событием код. Пока очередь не пуста процессор будет занят работой. Иначе говоря, запросы, а так же сам Веб сервер находятся в одном node процессе и постоянно висят в памяти.

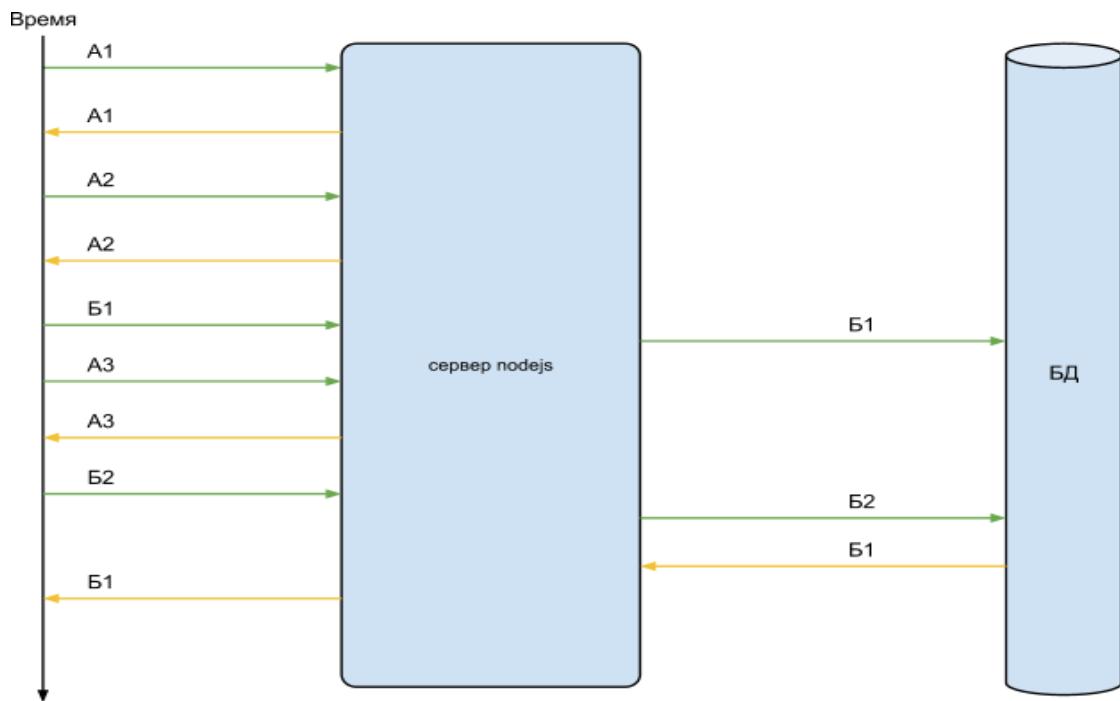


Рисунок 3 – Схема обработки запросов сервером Node.js

Запросы к базе данных не приводят к “подвисанию” сервера в ожидании ответа базы. Сервер получив запрос, просто сформирует и отправит запрос в базу данных, и продолжит отвечать на остальные запросы, как только ответ от базы будет получен, сервер вернет результат клиенту (запросы Б1 и Б2 на схеме). В случае Node.js неважно как и в каком количестве придут запросы, ни один из них не приведет к “подвисанию” в ожидании ответа базы.

4. Bootstrap

Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейсов, включая JavaScript расширения.

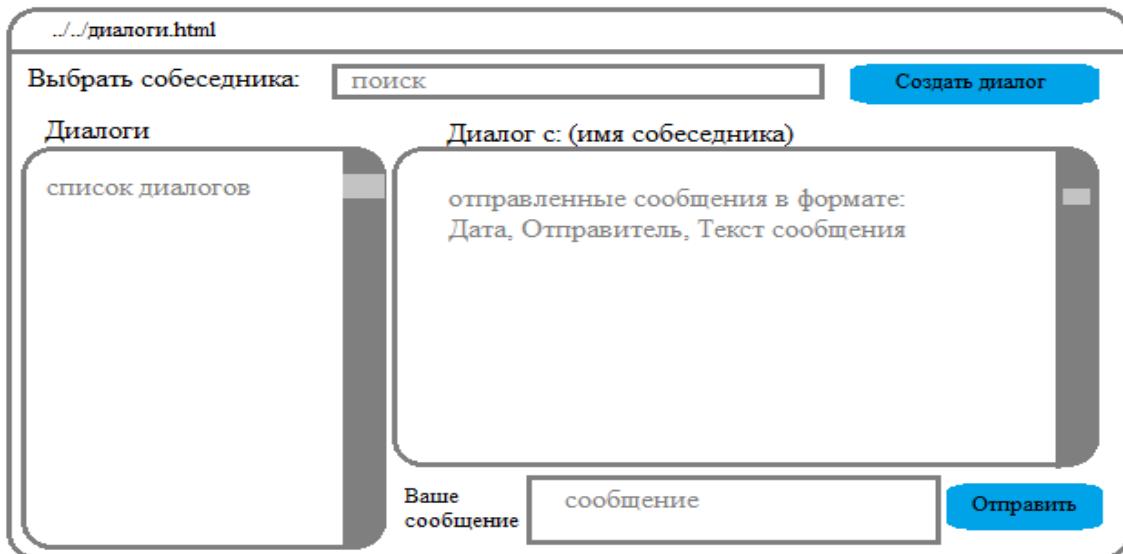
Основные преимущества:

- Экономия времени — Bootstrap позволяет сэкономить время и усилия, используя шаблоны дизайна и классы, и сконцентрироваться на других разработках;
- Высокая скорость — динамичные макеты Bootstrap масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке;
- Гармоничный дизайн — все компоненты платформы Bootstrap используют единый стиль и шаблоны с помощью центральной библиотеки. Дизайн и макеты веб-страниц согласуются друг с другом;

- Простота в использовании — платформа проста в использовании, пользователь с базовыми знаниями HTML и CSS может начать разработку с Bootstrap;
- Совместимость с браузерами — Bootstrap совместим с Mozilla Firefox, Google Chrome, Safari, Internet Explorer и Opera;
- Открытое программное обеспечение — особенность Bootstrap, которая предполагает удобство использования, посредством открытости исходных кодов и бесплатной загрузки.

Поведение итогов

Используя описанные выше технологии будет создан веб-сервис в виде классического чат-приложения, который будет удобен и привычен для использования пользователем.



- Вверху панель поиска собеседника
- Слева список уже созданных ранее диалогов
- Справа поле переписки, поле ввода своего сообщения и кнопка его отправки.

Список литературы:

1. Википедия. Свободная энциклопедия - <http://ru.wikipedia.org>
2. Официальный сайт фреймворка KOHANA - <http://kohanaframework.org>
3. Официальный сайт Bootstrap - <http://getbootstrap.com>
4. Коллективный блог и новостной сайт Habrahabr - <http://habrahabr.ru>
5. Официальный сайт платформы Node.js - <https://nodejs.org/>