

vanced level and www.connectionsacademy.com for elementary level, at school.

Research methods. This science work was designed as a descriptive study with an aim to explore the advantages, value and strong points of using technologies for online reading development, analysis and review of contemporary web-sites.

Theoretical significance: We have done an analytical and comparative work, made a valuable contribution to the studying of English language teaching methods. In our opinion, it will be quite right to research this problem on the basis of works of outgoing scholars such as H. Douglas Brown, J. Harmer, A. Maley, E.S. Post, M.K. Zhaksimbetova, A.S.Kadyrov, E.G. Azimov and others. This research also includes itself the list of widely using electronic manuals.

Structure. The first part of this science work is devoted to the theoretical background of online reading as a component of the process of foreign language teaching and learning.

The second part is devoted to the practice and shows the benefits of using this method for pupils of all ages at school and the appropriate tasks on this method.

In the conclusion, we summarize the results of our investigation and give proof of the theoretical and practical value of the work and make conclusion.

*Керничный А.В.
студент группы ИНФ-408,
КарГУ имени академика Е.А. Букетова
Допира Р.И.
м.т.н., ст. преподаватель,
КарГУ имени академика Е.А. Букетова*

АСИНХРОННОЕ ПРОГРАММИРОВАНИЕ ПРИ СЕТЕВОМ ВЗАИМОДЕЙСТВИИ

Вместе с развитием сети Интернет набирают все большую популярность различные сетевые приложения, облачные сервисы хранения данных, а также облачные вычисления. Это создает все большую потребность в программировании сетевых взаимодействий, вместе с тем создавая острую необходимость в построении эффективных серверных архитектур, способных обслуживать

множество клиентских приложений одновременно и при этом максимально эффективно использовать аппаратные возможности вычислительной машины. В данной работе рассмотрена проблема блокирующего ввода-вывода при сетевом программировании, пути решения: такие как многопоточный и событийно ориентированный подход к написанию серверных приложений. Выявлены преимущества и слабые стороны этих подходов. В качестве примера было разработано приложение web-чата для обмена сообщениями между пользователями с сервером написанном на платформе NodeJS, которая использует событийно ориентированную архитектуру, благодаря языку JavaScript, движку V8, и клиентом созданным с использованием основных web технологий таких как HTML, CSS, JS и Ajax, а так же популярных библиотек упрощающих разработку. Так же в работе рассмотрена основная идея и принцип работы платформы NodeJS.

При сетевом взаимодействии компьютерных программ большая часть времени тратится на ожидание в сети. Таким образом, использование синхронных (блокирующих) методов написания программ становится не разумным, так как использование такого подхода приводит к не эффективному использованию вычислительных мощностей машины. Вычислительная система будет простаивать, ожидая ответа от других систем или клиентского приложения, вместо того что бы заниматься чем-то полезным. Одним из способов решения данной проблемы является использование многопоточности, но проблема этого способа в том, что серверное приложение, использующее многопоточность, будет потреблять слишком много памяти при большом количестве запросов. Таким образом, использование асинхронного подхода при написании сетевых приложений является наиболее эффективным решением данной проблемы, поскольку такой подход решает нашу задачу с минимальными затратами: приложение не простаивает в ожидании получения данных от других систем, а имеет возможность принимать другие запросы, и отвечать лишь в случае получения этих данных, а также расход памяти становится более эффективным.

Актуальность данной работы обусловлена ростом потребности различных сетевых приложений. Быстрый, эффективный и надежный обмен и хранение данных является важной задачей в

современном обществе во всех его сферах. Начиная от образования заканчивая различными сферами общественного хозяйства. Именно поэтому важно задумываться не только о технической стороне вопроса передачи и хранения данных, таких как прокладка сетей и установка надежного оборудования, но и об разработке эффективных программных решений которые мы будем использовать в своей работе. Событийно-ориентированная асинхронная архитектура серверных приложений, в частности платформа NodeJS, уже отлично зарекомендовала себя и продолжает набирать популярность среди разработчиков, благодаря отличным показателям работы написанных на ее основе серверов, способных быстро обрабатывать огромное количество запросов, а также благодаря скорости и удобству разработки в сфере web.

Платформа NodeJS и используемый ею подход, покажет себя наиболее эффективно в задачах где отсутствует потребность в каких либо вычислениях. В задачах где критична скорость обмена данными, а это львиная доля всех интернет ресурсов в сети интернет. Информационные порталы, социальные сети, поисковики, игровые сервера, службы обмена мгновенными сообщениями и другие подобные ресурсы которые мы используем ежедневно, в основном не используют сложные расчеты, а используются как раз для того что бы получить или передавать информацию. Однако использование NodeJS не ограничено сферой web, ее можно с успехом применить во множестве других задач связанных с обменом данными используя более низкоуровневые протоколы связи типа UDP, TCP. Примером того могут послужить технологии интернета вещей. Удаленный сбор данных с показаний с различных датчиков с не сложной их обработкой, и принятие решений могут стать отличной задачей для сервера на NodeJS.

Цель данной научной работы показать использование асинхронного подхода к написанию сетевых приложений на примере создания сетевого веб-чата и сервера для него, используя язык JavaScript и платформу NodeJS.

Асинхронность - это способность вычислительного устройства или программы выполнять множество задач одновременно. Такое свойство очень важно для многих прикладных задач частными случаями которых являются, например, операционные системы. Асинхронное выполнение можно реализовать построением

различных архитектур, так например, многопоточной или событийно-ориентированной архитектурой. Клиентская часть реализована с помощью языка разметки HTML5, HTML шаблонизатора Jade, CSS3, языка JavaScript и библиотеки JQuery, технологии AJAX. Серверная часть реализована с помощью языка JavaScript и платформы NodeJS, фреймворка Express и модуля Jade. Веб интерфейс пользователя показан на рисунке 1.

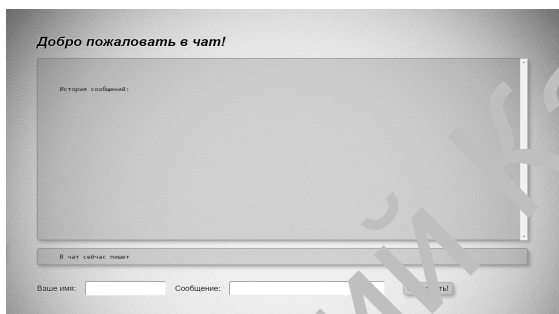


Рисунок 1. Пользовательский интерфейс чата

Веб чат на NodeJS демонстрирует нам пример работы асинхронной событийной модели. А так же демонстрирует производительность такой модели. Сервер написанный на NodeJS, может поддерживать до миллиона одновременных подключений на производительных машинах. Такая производительность важна для множества так называемых real time сервисов, где скорость обмена данными и возможность одновременной работы множества клиентов критична.

Таковыми real time сервисами могут быть сервисы мгновенного обмена сообщениями, различные онлайн игры, социальные сети, службы мониторинга. NodeJS это молодая технология начавшая свой путь в 2009 году, автором которой является Райан Дал. Несмотря на такой не большой путь NodeJS и событийно ориентированная архитектура построения приложений набирает все большую популярность благодаря эффективности и производительности приложений написанных на таких платформах.