



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004

АРХИТЕКТУРА WEB ПРИЛОЖЕНИЙ

Бубенин Д.Ю.

ФГБОУ ВО "ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ", Воронеж, Россия (394006, город Воронеж, ул. 20-летия Октября, д. 84), e-mail: dmitriy.bubenin@gmail.com

Данная работа даёт всесторонний обзор архитектуры веб-приложений, уделив особое внимание ее критическим компонентам. В введении подчеркивается важность веб-приложений в цифровом мире. Далее излагаются фундаментальные компоненты архитектуры веб-приложений, включая клиентскую часть, серверную часть и базу данных. В нем также рассматриваются ключевые архитектурные шаблоны, такие как MVC, микросервисы и бессерверные, демонстрируя их соответствующие преимущества и варианты использования. Затем в статье рассматриваются стратегии масштабируемости и особое внимание уделяется балансировке нагрузки, кэшированию, CDN и горизонтальному масштабированию. Далее в нем рассматриваются основные меры безопасности, такие как аутентификация, шифрование данных, проверка вводимых данных и защита от киберугроз. В заключение в статье подчеркивается ключевая роль архитектуры веб-приложений в современном цифровом мире. Это подчеркивает важность тщательного планирования и соблюдения лучших практик для создания веб-приложений.

Ключевые слова: Клиентская часть, Серверная часть, Паттерны архитектуры, Безопасность, Масштабируемость и производительность.

ARCHITECTURE OF WEB APPLICATIONS

Bubenin D. Yu.

VORONEZH STATE TECHNICAL UNIVERSITY, Voronezh, Russia (394006, Voronezh, 20-letiya Oktyabrya st., 84), e-mail: dmitriy.bubenin@gmail.com

This paper provides a comprehensive overview of the architecture of web applications, with a particular focus on its critical components. The introduction highlights the importance of web applications in the digital world. The following outlines the fundamental components of the web application architecture, including the client part, the server part and the database. It also examines key architectural patterns such as MVC, microservices and serverless, demonstrating their respective advantages and use cases. The article then discusses scalability strategies and focuses on load balancing, caching, CDN, and horizontal scaling. Next, it discusses basic security measures such as authentication, data encryption, verification of input data and protection against cyber threats. In conclusion, the article highlights the key role of web application architecture in the modern digital world. This highlights the importance of careful planning and adherence to best practices for building web applications.

Keywords: Client side, Server side, Architecture Patterns, Security, Scalability and Performance.

В постоянно меняющемся цифровом ландшафте веб-приложения стали краеугольным камнем современных технологий. Эти динамичные, ориентированные на пользователя программные системы обеспечивают работу всего - от платформ социальных сетей и веб-сайтов электронной коммерции до инструментов повышения производительности и систем управления контентом. За кулисами архитектура веб-приложений играет ключевую роль в

определении их функциональности, производительности, масштабируемости и безопасности. В этой статье мы рассмотрим ключевые аспекты архитектуры веб-приложений, проливающие свет на то, как работают эти сложные системы и как они эволюционировали, чтобы соответствовать требованиям эпохи цифровых технологий.

Понимание архитектуры веб-приложений

Архитектура веб-приложений относится к структуре и принципам проектирования, которые определяют, как организовываются, разрабатываются и развертываются веб-приложения[3]. По своей сути она включает в себя три основных компонента:

- **Клиентская часть:** это та часть веб-приложения, с которой пользователи взаимодействуют напрямую. Он включает в себя элементы пользовательского интерфейса (UI), которые обычно отображаются в веб-браузерах с использованием таких технологий, как HTML, CSS и JavaScript. Клиентская сторона отвечает за представление информации пользователям и обработку пользовательского ввода.
- **Серверная часть:** Серверная часть веб-приложения обрабатывает запросы со стороны клиента, обрабатывает их, взаимодействует с базами данных и возвращает соответствующие ответы. Этот компонент отвечает за логику приложения, бизнес-правила и управление данными. Распространенные серверные технологии включают серверные языки сценариев, такие как PHP, Python, Ruby, и серверные фреймворки, такие как Node.js, Django и Ruby on Rails[2].
- **База данных:** Базы данных хранят данные приложения и управляют ими. Они крайне важны для веб-приложений для хранения пользовательской информации, контента, конфигураций и других данных. Распространенные типы баз данных, используемых в веб-приложениях, включают реляционные базы данных (например, MySQL, PostgreSQL) и базы данных NoSQL (например, MongoDB, Cassandra).

При разработке веб-приложений для эффективной организации и структурирования кода обычно используется несколько архитектурных шаблонов. Вот три широко признанных паттерна:

Model-View-Controller (MVC): MVC - это популярный архитектурный шаблон, который разделяет приложение на три взаимосвязанных компонента: модель (данные и бизнес-логика), представление (пользовательский интерфейс) и контроллер (обрабатывает пользовательский ввод и управляет взаимодействиями между моделью и представлением). MVC способствует модульности кода и его сопровождаемости.

Микросервисы: В архитектуре микросервисов веб-приложение разделено на набор небольших независимых сервисов, которые взаимодействуют через API. Каждая служба фокусируется на определенной функции или фичах, что упрощает ее масштабирование, обслуживание и обновление. Микросервисы обеспечивают гибкость и часто используются в крупномасштабных и сложных приложениях [4].

Бессерверная архитектура абстрагирует управление сервером, позволяя разработчикам сосредоточиться исключительно на написании кода, не беспокоясь о подготовке и обслуживании сервера. Такие сервисы, как AWS Lambda, функции Azure и облачные функции Google, являются примерами бессерверных платформ, которые выполняют код в ответ на события, что делает их экономически эффективным и масштабируемым подходом.

Масштабируемость и производительность

Веб-приложения должны разрабатываться с учетом масштабируемости и производительности, чтобы соответствовать различным уровням трафика и требованиям пользователей. Такие стратегии, как балансировка нагрузки, кэширование, сети доставки контента (CDN) и горизонтальное масштабирование (добавление большего количества серверов), помогают гарантировать, что веб-приложения смогут справляться с возросшими рабочими нагрузками без снижения производительности.

Безопасность

Безопасность имеет первостепенное значение в архитектуре веб-приложений. Общие меры безопасности включают:

Аутентификация и авторизация: Внедрение механизмов безопасного входа в систему и контроля доступа для защиты пользовательских данных и конфиденциальных операций.

Шифрование данных: Шифрование данных при передаче (HTTPS) и в состоянии покоя (шифрование базы данных) для предотвращения утечки данных.

Проверка вводимых данных: проверка пользовательских вводимых данных для предотвращения внедрения SQL, межсайтового скриптинга (XSS) и других уязвимостей в системе безопасности.

Брандмауэры и защита от DDoS-атак: Использование брандмауэров и служб предотвращения DDoS-атак для защиты от атак [1].

Заключение

Архитектура веб-приложений - это многогранная тема, которая сочетает в себе клиентские и серверные технологии, базы данных и различные архитектурные шаблоны для обеспечения мощного, отзывчивого и безопасного пользовательского интерфейса. Успешная разработка веб-приложений требует глубокого понимания этих архитектурных принципов и приверженности лучшим практикам, гарантируя, что ваши цифровые творения выдержат испытание временем в нашем постоянно подключенном мире. Поскольку технологии продолжают развиваться, архитектура веб-приложений будет оставаться на переднем крае, адаптируясь к вызовам и возможностям цифрового будущего. Овладевая этими принципами, разработчики и архитекторы могут продолжать создавать инновационные, эффективные и безопасные веб-приложения, которые обогащают наш онлайн-опыт и продвигают цифровую революцию вперед.

Список литературы

1. Госмен, Джошуа. "О безопасности веб-приложений." O'Reilly Media, 2019.
2. Гоффин, Джеймс Д., и Дэвид Р. Стоут. "Веб-разработка с использованием Node.js, Express и MongoDB." O'Reilly Media, 2014.
3. Соуверс, Стивен, и Лоуренс Дэвидсон. "Программная архитектура: современные методы проектирования." ООО "И. Д. Вильямс", 2016.
4. Фаулер, Мартин, и Дэвид Райс. "Архитектура микросервисов." O'Reilly Media, 2015.

References

1. Gossman, Joshua. "On Web Application Security." O'Reilly Media, 2019.
 2. Goffin, James D., and David R. Stoa. "Web development using Node.js, Express and MongoDB." O'Reilly Media, 2014.
 3. Souvers, Stephen, and Lawrence Davidson. "Software architecture: modern design methods." I. D. Williams LLC, 2016.
 4. Fowler, Martin, and David Rice. "Architecture of microservices." O'Reilly Media, 2015.
-