Веб-программирование

ИММиКН ЮФУ

Mayer Svetlana (Майер С.Ф.)
sfmayer@sfedu.ru

Современный инструментарий разработчика

Две стороны веб-разработки

FRONTEND (Клиент)	BACKEND (Сервер)
Что видит пользователь • Интерфейс • Дизайн • Анимации • Взаимодействие	**Что не видит пользователь** • Бизнес-логика • Работа с БД • Аутентификация • API (интерфейс прикладного программирования)

Frontend (Клиентская часть)

Frontend — это клиентская часть веб-приложения, которая выполняется в браузере пользователя. Всё, что вы видите на сайте — результат работы фронтенда.

HTML (Структура)	CSS (Стили)	JavaScript (Поведение/Логика)
		• Динамика • Обработка событий • Взаимодействие с сервером • Анимации

Библиотеки и фреймворки фронтенда

Библиотека — набор готовых функций и компонентов, которые можно использовать в своём коде.

Фреймворк — каркас приложения, определяющий архитектуру и структуру кода.

Популярные js-библиотеки

React	• Создание компонентов • Виртуальный DOM • Огромное сообщество
jQuery (устаревает)	• Упрощение работы с DOM • Кросс-браузерность • Меньше кода
D3.js	• Визуализация данных • Графики и диаграммы • Работа с SVG

Популярные frontend-фреймворки

Angular

- Полноценный фреймворк от Google
- TypeScript
- Для крупных enterprise-проектов

Vue.js

- Простота освоения
- Гибкость

React

- Библиотека (часто называют фреймворком)
- Компонентный подход
- Экосистема инструментов

Bootstrap (**CSS-фреймворк**)

Препроцессоры CSS

Расширяют возможности обычного CSS:

```
// SASS/SCSS пример
$primary-col
$spacing: 20

.button {
    backgrou
    padding:

    &:hover
        back
    }

    &-large
        font
        @extend .button;
    }
}
```

Препроцессор	Особенности
SASS/SCSS	Переменные, вложенность
LESS	Похож на SASS, используется в Bootstrap
PostCSS	Плагины, автопрефиксы, будущий CSS

Backend (Серверная часть)

Backend — это серверная часть веб-приложения, которая обрабатывает запросы от клиента, работает с базами данных и выполняет бизнес-логику.

Языки программирования бэкенда

JavaScript (Node.js)

- Node.js
- Единый язык на front/back
- Огромная экосистема (npm)

Python

- Надёжность и производительность
- Spring Boot, Jakarta EE
- Корпоративные системы

PHP

- Специализация на вебе
- Laravel, Symfony, WordPress
- Простое развёртывание

C#

- .NET платформа от Microsoft
- ASP.NET Core
- Windows-интеграция

Go

- Высокая производительность
- Простота и скорость
- Микросервисы

Ruby

- Ruby on Rails
- Конвенция над конфигурацией
- Быстрая разработка

Фреймворки бэкенда

Язык	Фреймворки	Особенности
JavaScript	Express.js, Nest.js, Koa	Минималистичные, гибкие
Python	Django, Flask, FastAPI	Django — "всё включено"
Java	Spring Boot, Micronaut	Мощные
PHP	Laravel, Symfony, Yii	Удобные, быстрая разработка
C#	ASP.NET Core	Кроссплатформенный
Ruby	Ruby on Rails	Соглашения, быстрая разработка
Go	Gin, Echo	Высокая производительность

Базы данных

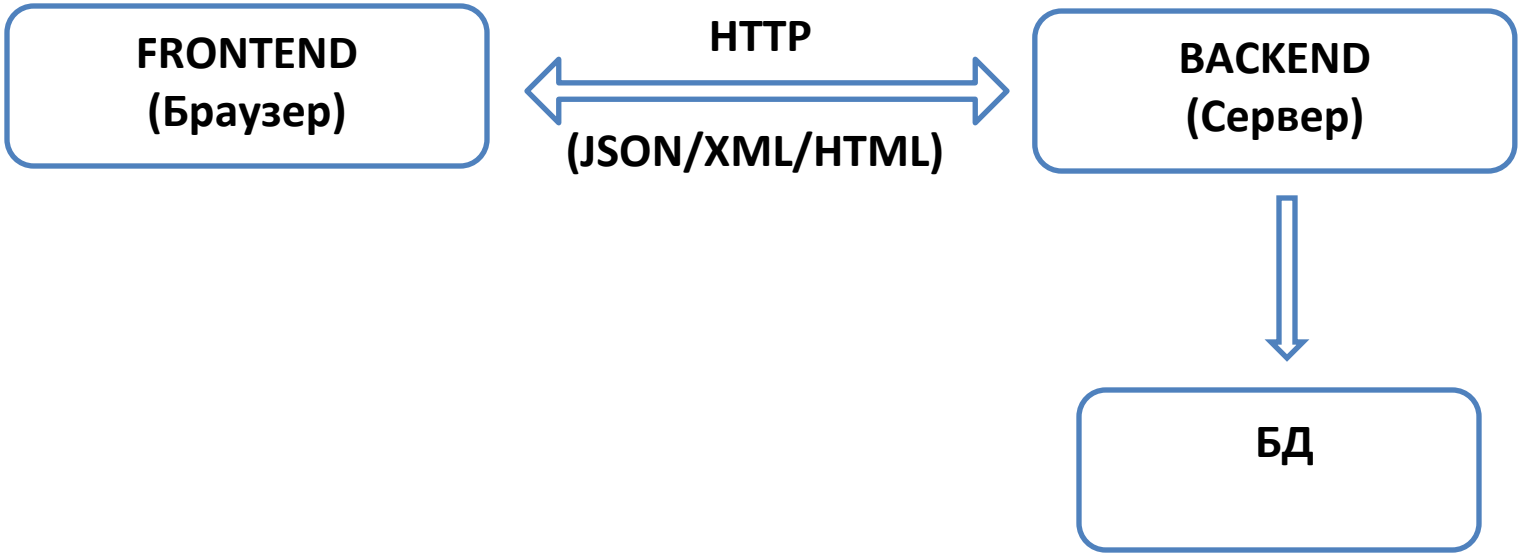
Реляционные (SQL)

- **MySQL** — популярная, бесплатная
- **PostgreSQL** — мощная, расширяемая
- **Oracle** — корпоративная
- **Microsoft SQL Server** — от Microsoft

Нереляционные (NoSQL)

- **MongoDB** — документо-ориентированная
- **Redis** — кэширование, быстрая
- **Cassandra** — распределённая

Взаимосвязь Frontend и Backend



API (Application Programming Interface)

API — это посредник между фронтендом и бэкендом.

REST API (самый популярный)

```
GET    /api/users              # Получить всех пользователей
GET    /api/users/1            # Получить пользователя с id=1
POST   /api/users               # Создать нового пользователя
PUT    /api/users/1             # Обновить пользователя
DELETE /api/users/1             # Удалить пользователя
```

Полный стек технологий (Full Stack)

FRONTEND	BACKEND
• HTML • CSS/SCSS • JavaScript • React/Vue/Angular	• **Язык**: Node.js/Python/Java/PHP • **Фреймворк**: Express/Django/Spring • **База данных**: MySQL/PostgreSQL/Mongo • **API**: REST/GraphQL • **Сервер**: Nginx/Apache

Fullstack

Специалист создает полное решение — от пользовательского интерфейса до серверной логики и базы данных. Fullstack-разработчик — это универсальный специалист, который способен работать как над клиентской (фронтенд), так и над серверной (бэкенд) частью приложения. Он обладает знаниями во всем стеке технологий.

Популярные комбинации (Full Stack варианты)

1.LAMP / LEMP (Классика)

- **Linux** (Операционная система)
- **Apache / Nginx** (Веб-сервер)
- **MySQL / MariaDB** (База данных)
- **PHP / Perl** (Язык программирования)

Для чего: Хостинг CMS (WordPress, Joomla), классические веб-сайты;
для простых сайтов, блогов, корпоративных порталов, где не требуется сложная интерактивность.

Linux + Apache + MySQL + PHP

Бэкенд:

- **PHP** — язык программирования для серверной логики.
- **MySQL** — реляционная СУБД для хранения данных.
- **Apache** — веб-сервер, обрабатывает HTTP-запросы.

Фронтенд:

- **HTML/CSS** — разметка и стилизация интерфейса.
- **JavaScript** — интерактивность на стороне клиента.
- **jQuery** — библиотека JavaScript для упрощения работы с DOM, AJAX и анимациями.

Дополнительно:

- **phpMyAdmin** — веб-интерфейс для управления MySQL.

2. MEAN / MERN / MEVN (JavaScript везде)

- **MongoDB** (База данных - NoSQL)
- **Express.js** (Backend фреймворк для Node.js)
- **Angular / React / Vue.js** (Frontend фреймворк/библиотека)
- **Node.js** (Среда выполнения на сервере)

Для чего: Современные одностраничные приложения (SPA), приложения реального времени, стартапы. Использует один язык (JS/TS) на всех уровнях.

MongoDB + Express.js + Angular/React/Vue + Node.js

Бэкенд:

- **Node.js** — среда выполнения JavaScript на сервере.
- **Express.js** — фреймворк для Node.js, упрощает маршрутизацию и обработку запросов.
- **MongoDB** — документоориентированная NoSQL СУБД, хранит данные в формате JSON-подобных документов.

Фронтенд:

- **React** — библиотека для построения пользовательских интерфейсов, использует компонентный подход и виртуальный DOM.
- **Angular –** фреймворк по принципу SPA
- **Vue.js** - фреймворк
- **JSX** — синтаксис для описания структуры интерфейса в React.
- **Axios** — библиотека для выполнения HTTP-запросов к API.

Сборка и инструменты:

- **Webpack** — сборщик модулей, объединяет код и зависимости.
- **Babel** — транспилятор, преобразует современный JavaScript в совместимый с браузерами код.

3. JAMstack (Современный подход к статике)

- **JavaScript** (Динамика на клиенте)
- **APIs** (Серверные функции как микросервисы или serverless)
- **Markup** (Заранее сгенерированный HTML)

Технологии: Gatsby, Next.js, Nuxt.js + Headless CMS (Strapi, Sanity) + CDN.

Для чего: Блоги, маркетинговые сайты, интернет-магазины с высокой скоростью загрузки; для корпоративных систем, банковских приложений, высоконагруженных сервисов с требованиями к безопасности и стабильности.

4. Python-стеки (Django/Flask + ...)

- **Django / Flask / FastAPI** (Backend фреймворк)
- **PostgreSQL / MySQL** (База данных)
- **React / Vue.js / Alpine.js** (Frontend)

Для чего: Проекты, где важна скорость разработки и наличие мощных библиотек для Machine learningL/Artificial Iintelligence. Отлично подходит для стартапов и MVP.

Стек на Python (Django + React)

Бэкенд:

- **Python** — универсальный язык программирования.
- **Django** — высокоуровневый фреймворк с «батарейками в комплекте»: ORM, админ-панель, система аутентификации.
- **PostgreSQL** — мощная реляционная СУБД с поддержкой сложных запросов и транзакций.
- **DRF (Django REST Framework)** — расширение для Django, позволяет быстро создавать RESTful API.

Фронтенд:

- **React** — для построения динамичного интерфейса.
- **Redux** — библиотека управления состоянием приложения, обеспечивает предсказуемое изменение данных.
- **SCSS/SASS** — препроцессор CSS, добавляет переменные, миксины и вложенность.

Деплой и инфраструктура:

- **Docker** — контейнеризация приложения для единообразной работы в разных средах.
- **Nginx** — веб-сервер и обратный прокси для статических файлов и балансировки нагрузки.
- **Когда подходит:** для сложных веб-приложений с развитой бизнес-логикой, административными панелями, аналитикой.

5. .NET Stack (C#)

- **ASP.NET Core** (Backend фреймворк)
- **MS SQL Server / PostgreSQL** (База данных)
- **React / Angular / Blazor** (Frontend)

Для чего: Крупные корпоративные приложения, финансовый сектор, проекты, интегрированные с экосистемой Microsoft.

Стек на .NET (ASP.NET Core + Blazor)

Бэкенд:

- **.NET Core** — кроссплатформенная среда разработки от Microsoft.
- **ASP.NET Core** — фреймворк для создания веб-API и веб-приложений.
- **Entity Framework Core** — ORM для работы с реляционными базами данных.
- **MS SQL Server** — СУБД от Microsoft, интегрирована с экосистемой .NET.

Фронтенд:

- **Blazor** — фреймворк для построения интерактивных веб-UI на C#.
  Может работать на стороне сервера (Blazor Server) или в браузере через WebAssembly (Blazor WebAssembly).
- **Bootstrap** — CSS-фреймворк для адаптивной вёрстки.

Инструменты:

- **Visual Studio** — IDE для разработки под .NET.
- **Azure DevOps** — CI/CD и управление проектами.

6. Ruby on Rails (Монолит)

- **Ruby** (Язык)
- **Rails** (Фреймворк, включающий всё)
- **PostgreSQL / MySQL** (База данных)
- **Stimulus / Hotwire** (Frontend)

Для чего: Стартапы (известен скоростью вывода на рынок), MVP, проекты, где важна "магия" Rails.

7. Стек на Java (Spring Boot + Angular)

Бэкенд:

- **Java** — строго типизированный язык с кроссплатформенностью.
- **Spring Boot** — фреймворк для быстрого создания микросервисов и REST API, упрощает конфигурацию.
- **Hibernate** — ORM-фреймворк, связывает объекты Java с таблицами базы данных.
- **Oracle DB или PostgreSQL** — промышленные СУБД для надёжного хранения данных.

Фронтенд:

- **Angular** — полноценный фреймворк от Google для создания сложных SPA.
- **TypeScript** — надмножество JavaScript с строгой типизацией, улучшает качество кода.
- **RxJS** — библиотека реактивного программирования для работы с потоками данных.

Инфраструктура:

- **Maven/Gradle** — системы сборки и управления зависимостями для Java.
- **Kubernetes** — оркестрация контейнеров для масштабирования и управления микросервисами.

Fullstack-разработка

Плюсы подхода: Широкий кругозор, возможность вести проект от идеи до запуска, высокая востребованность на рынке.

Сложности: Необходимость постоянно поддерживать актуальность огромного массива знаний.

Интересный факт: Технология Node.js, которая позволила JavaScript "спуститься с клиента на сервер", была создана в 2009 году Райаном Далем. Это произвело революцию, так как позволило использовать один язык программирования и для фронтенда, и для бэкенда, что значительно упростило жизнь fullstack-разработчикам.

Краткий сравнительный анализ

Критерий	MERN	Django + React	Spring Boot + Angular	.NET + Blazor
Язык бэкенда	JavaScript	Python	Java	C#
СУБД	MongoDB (NoSQL)	PostgreSQL (SQL)	Oracle/PostgreSQL	MS SQL Server
Скорость разработки	Высокая	Средняя	Низкая	Средняя
Производительность	Средняя	Высокая	Очень высокая	Высокая
Сообщество	Большое	Большое	Огромное	Большое
Типизация	Динамическая	Динамическая	Статическая	Статическая
Подходит для стартапов	Да	Да	Нет	Частично

Как выбрать стек

При выборе учитывать:

1. **Требования к проекту.** Для высоконагруженных систем — Java или .NET; для стартапов — Node.js или Python.
2. **Опыт команды.** Если разработчики знают Java, не стоит навязывать им PHP.
3. **Масштабируемость.** Микросервисы проще строить на Spring Boot или .NET Core.
4. **Бюджет.** Open-source решения (Python, Node.js) дешевле, чем лицензии на Oracle и MS SQL.
5. **Сроки.** MERN и Django позволяют быстрее запустить MVP.