Welcome!

WWCode San Francisco - Backend Study Group

November 2, 2023

- We'll start in a moment :)
- We are **RECORDING** tonight's event
- We may plan to take screenshots for social media
- If you are comfortable, turn the video ON. If you want to be anonymous, then turn the video off
- We'll make some time for Q&A at the end of the presentation
- Feel free to take notes
- Online event best practices:
    - Don't multitask. Distractions reduce your ability to remember concepts
    - Mute yourself when you aren't talking
    - We want the session to be interactive
    - Use the 'Raise Hand' feature to ask questions
- **By attending our events, you agree to comply with our Code of Conduct**

WOMEN WHO CODE /san-francisco

Introduction & Agenda

- Welcome from WWCode!
- Our mission: Empower women to excel in technology careers
- At Backend Study Group, we learn and discuss backend engineering concepts

Prachi Shah
Instructor
Sr. Software Engineer, Unity
Director, WWCode SF

Anjali Bajaj
Co-Host,
Lead, WWCode SF

How to be a good Backend Engineer:

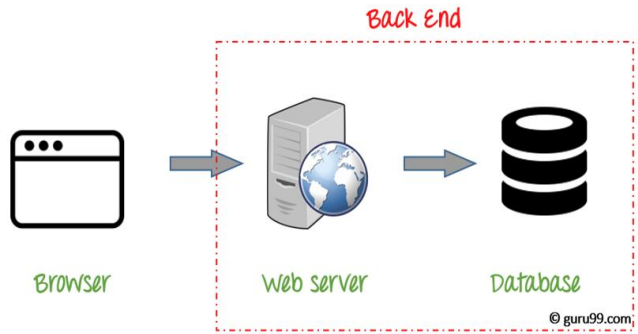
- What is the Backend Engineer role?
- Skills required
- Technology stack
- Career path
- Q & A

WOMEN WHO CODE /san-francisco

Backend Engineering

- Design, build and maintain server-side web applications

- Concepts: Client-server architecture, networking, APIs, web fundamentals, microservices, databases, security, operating systems, etc.



- Tech Stack: Java, PHP, .NET, C#, Ruby, Python, REST, AWS, Node, SQL, NoSQL, etc.

Backend Engineer

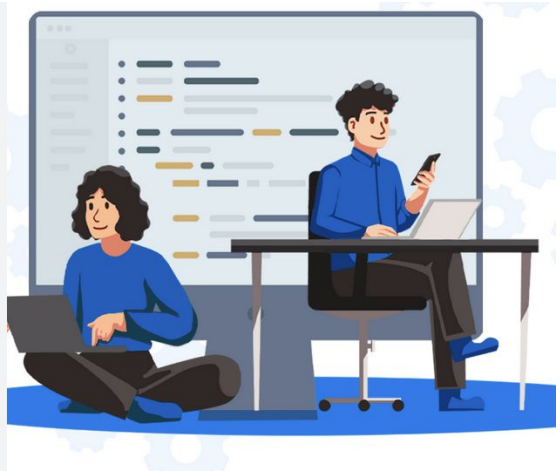
- Server-side development, focusing on how the website works, updates, and changes.

- Create logic, work behind the scenes to power the software or website.

- Build robust and scalable systems, handle data and ensure everything on the client-side actually works.

- Create libraries, write APIs, work with system components to ensure they interact properly, and database management.

- Business logic and domain-specific implementation matters.



Backend Engineer

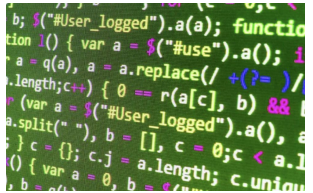
• Specializes in designing, implementing, and maintaining the server, database, and application logic that powers the frontend or user-facing components of a system.

• Database Management: Design, implement, and manage databases. Creating schemas, writing SQL queries, optimizing database performance, and ensuring data integrity.
Example: Implementing a relational database using PostgreSQL to store user profiles for a social networking site.

• Server-Side Logic: Write the logic that processes user requests, interacts with the database, and sends the appropriate response back to the client.
Example: Developing the logic to handle user authentication, where a user submits a username and password, the server verifies it against stored data, and then either grants or denies access.

• API Development: Design and develop APIs (Application Programming Interfaces) that allow different software components to communicate.
Example: Designing a RESTful API that allows a mobile application to fetch user data from a server.

Backend Engineer



• Integration: Integrate various systems and servers to ensure smooth data flow between frontend and backend, or between different backend services.
Example: Integrating a payment gateway like Stripe to handle online transactions on an e-commerce platform.

• Performance Optimization: Ensuring that the server responds quickly to requests and can handle large numbers of simultaneous users. Example: Implementing caching mechanisms using tools like Redis to speed up data retrieval processes.

• Security: Implementing security measures to protect sensitive data and ensure that the server is resistant to common threats and vulnerabilities. Example: Setting up secure HTTPS connections, implementing data encryption, and guarding against SQL injection attacks.

• Infrastructure and Deployment: Setting up and maintaining server infrastructure, which might involve cloud platforms like AWS, Azure, or Google Cloud, and ensuring that software changes are safely deployed to production. Example: Setting up a continuous integration and deployment (CI/CD) pipeline using tools like Jenkins or CircleCI.

Skills

• Programming Languages:
Developing server-side logic using languages like Java or Python which are known for their robustness and scalability.

• Database Management:
Creating and managing a database using SQL or MongoDB to store and organize data efficiently.

• API Design and Development:
Designing a RESTful API that allows different software components to communicate with each other.

• Understanding of Frontend Technologies:
Collaborating with front-end developers to ensure the backend logic works well with the front-end display.

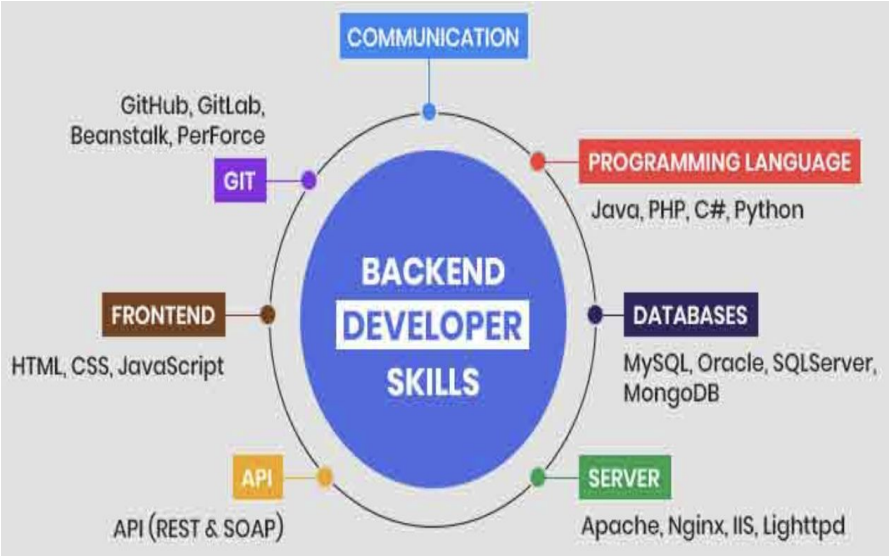
• Problem-Solving Ability:
Debugging a complex issue where data isn't being saved to the database correctly.

Skills

• Version Control Systems:
Using Git to track changes in your code over time and collaborate with other developers.

• Security:
Ensure application code and data is secure, privacy and compliance is maintained.

• Networking:
Knowledge of networking protocols.



• Operating System:
Knowledge of operating systems and devices (mobile, computer, tablet, etc.)

• CS Fundamentals:
Algorithms and data structures.

Soft Skills

- Project Management:

Task prioritization, estimates and execution.

- Team Player:

Collaborate on ideas and solutions.

- Patience:

Development is slower compared to front-end development.
Things take time! :)

- Problem Solving:

Articulate the right problems to solve and approach to debugging.

- Analytical:

Tackle complicated technical problems and gathering information.

- Attention to Detail: Focus on tasks and ensure highest quality of code.

Technology Stack

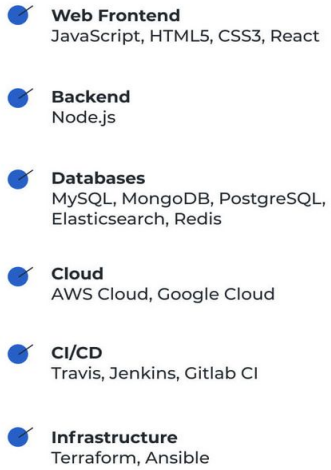
• IDEs:
IntelliJ IDEA, PyCharm, Eclipse, or Visual Studio Code.
Example: Using IntelliJ IDEA for developing and debugging Java applications.

• Database Tools:
MySQL Workbench, MongoDB Compass.
Example: Utilizing MySQL Workbench for designing and documenting database architecture.

• API Testing Tools:
Postman, Insomnia.
Example: Employing Postman to test API endpoints and ensure they're functioning as expected.

• Version Control Tools:
Git, GitHub, Bitbucket.
Example: Collaborating on a project with a team using GitHub to manage code versions and review code.

Technology Stack



• MEAN/MERN:
(MongoDB, Express.js, Angular/React, Node.js)
Example: Building a dynamic web application using the MERN stack which is known for its ease of use and robustness.

• Ruby on Rails:
Rails platform and Ruby programming language.
Example: Developing a social media platform using Ruby on Rails for its rapid development capabilities.

• Spring Boot:
Convention-over-configuration framework for development in Spring and Java.
Example: Developing a microservices architecture using Spring Boot for a large-scale enterprise application.

Career Path

• Entry-Level Backend Developer:
Gaining experience and mastering the necessary skills and tools.

• Mid-Level Backend Developer:
Taking on more complex projects, mentoring junior developers.

• Senior Backend Developer/Lead Backend Developer:
Overseeing projects, making architectural decisions, leading teams.

• Backend Architect/Engineering Manager:
Designing system architectures, managing engineering teams.

• Director of Engineering/CTO:
Strategic planning, overseeing engineering operations, driving technological advancements.

Backend Study Group

References:

- Backend Engineer Fundamentals
- Backend Engineer vs. Backend Developer

Backend Study Group:

- Presentations on GitHub and session recordings available on WWCode YouTube channel
- January 18th 2024: Web Development 101
- More events in February and March! RSVP at meetup.com/women-who-code-sf

Women Who Code:

- Technical Tracks and Digital Events for more events
- Join the Digital mailing list for updates about WWCode
- Contacts us at: contact@womenwhocode.com
- Join our Slack workspace and join *#backend-study-group*!

You can unmute and talk or use the chat

WOMEN WHO CODE /san-francisco