

# Understanding Back-End Development Careers and Skills

47 sources

The provided text from Coursera offers a comprehensive overview of **back-end development** as a career path. It **defines the role** of back-end developers in building the unseen infrastructure of websites and applications, contrasting it with front-end development. The article **details the responsibilities and essential skills**, including programming languages, frameworks, databases, and crucial workplace abilities. Furthermore, it touches upon **salary expectations, job outlook, necessary education**, and provides pathways for individuals interested in entering this in-demand field, including recommended Coursera resources.

keep

Save to note

copy\_all

docs

Add note

audio\_magic\_eraser

Audio Overview

map

Mind Map

Can u help me summerize in simply terms

Let's summarize the information from the sources in simple terms.

The YouTube video transcript<sup>1</sup> ... is like a teacher explaining the basics of **backend development using JavaScript**. The instructor, Harsh, plans to teach backend using **Node.js, MongoDB, and Express<sup>1</sup>**. He promises to make it easy to understand<sup>2</sup>.

Here are some key things covered in the video:

- 

**Fundamental JavaScript concepts** that are important for backend development are discussed<sup>8</sup> ....

- 

**Arrays** are introduced as a way to store multiple values<sup>11</sup>.

- 

Important **array methods** like `forEach`, `map`, `filter`, `find`, and `indexOf` are explained with examples<sup>12</sup> .... These methods help you work with lists of data.

- 

`forEach` lets you do something with each item in an array<sup>13</sup> ....

- 

`map` creates a new array by doing something to each item in the original array<sup>17</sup> ....

- 

`filter` creates a new array containing only the items that meet a certain condition<sup>30</sup> ....

- 

`find` returns the first item in an array that meets a certain condition<sup>38</sup> ....

- 

`indexOf` tells you the position of a specific item in an array<sup>40</sup> ....

- 

**Objects** are explained as collections of key-value pairs, which are very important in JavaScript for backend development<sup>42</sup> .... You can access and modify values in objects, and even freeze them to prevent changes<sup>46</sup> ....

- 

**Functions** are blocks of code that can be run<sup>50</sup>. The video explains how the `return` statement works, which sends a value back from a function<sup>52</sup> ....

-

**Asynchronous JavaScript** is introduced with `async` and `await`<sup>56</sup> ... This is used for tasks that might take some time, like fetching data from the internet, so your program doesn't freeze while waiting<sup>60</sup> ...

The Coursera article<sup>74</sup> ... explains **what a backend developer does**. They work on the parts of a website or app that you **can't see**, focusing on how data is processed and stored<sup>74</sup> ...

Here's a breakdown of what backend development involves:

- 

**Server-side software:** Backend developers work on the logic, databases, and servers that make a website function correctly<sup>75</sup> .

- 

**Collaboration:** They work with frontend developers and other team members<sup>76</sup> .

- 

**Key responsibilities** include:

- 

Building and maintaining websites<sup>78</sup> .

- 

Writing high-quality and maintainable code<sup>78</sup> .

- 

Ensuring the website performs well in terms of speed and efficiency<sup>79</sup> .

- 

Troubleshooting and fixing problems<sup>79</sup> .

- 

**Common tools and technologies** used by backend developers include:

- 

**Programming languages:** Python, PHP, JavaScript (with Node.js), Ruby, Java, C#<sup>80</sup> ...

- 

**Frameworks:** Laravel, Django, Spring, Ruby on Rails, Meteor, **Node.js**, Express<sup>80</sup> ...

- 

**Databases:** MongoDB, MySQL, Oracle<sup>80</sup> ...

- 

**Servers:** Apache, NGINX, Microsoft IIS<sup>80</sup> ...

- 

**APIs (Application Programming Interfaces)** are used to help different software communicate<sup>75</sup> ...

- **Essential skills** for backend developers include:

- Technical expertise in programming languages, frameworks, and databases77 ....

- Analytical thinking and problem-solving skills77 ....

- Good communication and collaboration abilities77 ....

- **Job outlook:** Backend developers are in high demand, and the field is expected to grow81 . The typical salary in the US is around \$115,000 per year81 .

- While a degree in computer science can be helpful, it's possible to become a backend developer through other paths by learning the necessary technical skills82 ....

In simple terms, the YouTube series aims to teach you the JavaScript skills and specific technologies like Node.js and MongoDB needed for backend development, while the Coursera article explains what a backend developer does and the broader context of this career path.

keep\_pin

Save to note

copy\_all

thumb\_up

## thumb\_down

who is the person in the video what is the relationship with backend development why should i listen to him

The person in the video is **Harsh**<sup>1</sup>. He introduces himself at the beginning of the video<sup>1</sup>.

Harsh's relationship with backend development is that he is the instructor for a **backend development series**<sup>1</sup>. He states that this series will focus on **Backend Development with Node.js, MongoDB, and Express**, along with "Few More Things"<sup>1</sup> ... He explicitly says, "I will teach you how to write a backend Node App and Express and few more things. How to create good APIs Scalable APIs and Secure APIs That's what I will teach you"<sup>2</sup>. You should consider listening to Harsh for the following reasons, based on the information he provides:

- 

**He claims significant experience in backend development:** Harsh mentions that he has been writing **Node.js** and "few more things" for his brand for years<sup>3</sup>. He also states that they manage **sheryians.com** and all their courses, with thousands of students, indicating practical experience in building and managing backend systems<sup>3</sup>. He further mentions that they have written a lot of **scalable code**, suggesting an understanding of how to build robust applications<sup>4</sup>.

- 

**He aims to teach in an easy-to-understand way:** Harsh explicitly states that he will teach in a way that will make the backend "very easy" to understand, particularly for those who have struggled with it before<sup>5</sup>.

- 

**He intends to provide deeper insights and correct potential misunderstandings:** He highlights that his teaching will be a "little different" and mentions an example with MongoDB IDs to illustrate how he

will explain things in multiple aspects and at a deeper level, potentially correcting common misconceptions<sup>1</sup> ...

- 

**He plans to cover important aspects of backend development:** He explicitly mentions teaching how to create **good APIs, Scalable APIs, and Secure APIs**, which are crucial concepts in backend development<sup>2</sup>.

- 

**He wants to showcase their teaching style:** Harsh mentions that while they also have a backend course, this free series is intended to introduce their style of teaching to a wider audience<sup>2</sup> ... This allows you to experience his teaching before committing to any paid courses.

In summary, you might want to listen to Harsh because he claims substantial practical experience in backend development using the technologies he intends to teach, promises to explain concepts in an accessible manner, aims to provide deeper understanding by addressing potential misconceptions, and plans to cover essential topics like API development.

keep\_pin

Save to note

copy\_all

thumb\_up

thumb\_down

whats the best way to start backend security

Drawing on the sources and our conversation history, here's a breakdown of the best way to start learning backend security:

First, it's important to understand that **security is a fundamental skill for backend developers**<sup>1</sup> .... As KG mentioned their plan to learn more about backend security to make a career out of it<sup>3</sup>, this is a crucial area of focus. Here's a suggested approach to start learning backend security:

- 

**Master a Backend Programming Language:** The foundation of backend development, including security, starts with a strong understanding of a programming language<sup>1</sup> .... Several languages are popular, such as Python, Java, and JavaScript (Node.js)<sup>1</sup> .... According to one source, JavaScript is a widely used language by professionals<sup>8</sup>. Choosing a language and becoming comfortable with its syntax and core concepts is the first step<sup>9</sup>.

- 

**Learn Database Management:** Backend applications heavily rely on databases for storing and retrieving data<sup>10</sup> .... Understanding how databases work, including concepts like data modeling, efficient queries, and importantly, **how to interact with them securely (e.g., preventing SQL injection)**, is crucial<sup>12</sup> .... Familiarity with both SQL databases like MySQL and PostgreSQL, and NoSQL databases like MongoDB, can be beneficial<sup>12</sup> ....

- 

**Understand API Development:** Backend developers often build APIs to enable communication between different parts of an application (like the frontend) or other services<sup>10</sup> .... Learning how to design, build, and **secure these APIs** is essential. This includes understanding authentication (verifying who the user is) and authorization (determining what the user is allowed to do)<sup>2</sup> ....

- 

**Study a Backend Framework:** Frameworks provide tools and structure for building backend applications, often including built-in security features and

best practices<sup>1</sup> .... For example, for Python, Django and Flask are popular<sup>1</sup> ..., for Java, Spring Boot<sup>1</sup> ..., and for JavaScript, Express.js<sup>1</sup> .... Learning a framework can help you implement security measures more efficiently.

- 

### **Focus on Security Skills Specifically:**

- 

**Learn about common web security vulnerabilities:** Understand threats like SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and how to prevent them<sup>15</sup> .... One source explicitly mentions the importance of coding defensively as if the frontend is potentially malicious<sup>35</sup> .

- 

**Study secure coding practices:** Learn how to write code that minimizes security risks. This includes input validation, output encoding, and proper handling of sensitive data like passwords (e.g., using hashing and salting)<sup>32</sup> .

- 

**Understand authentication and authorization mechanisms:** Learn different ways to verify user identities and control access to resources<sup>2</sup> ....

- 

**Familiarize yourself with data encryption techniques:** Understand how to protect data at rest and in transit<sup>2</sup> ....

- 

**Learn about network protocols and web security concepts<sup>25</sup> .**

- 

**Use Learning Resources:** Utilize online courses, tutorials, and documentation to learn these concepts<sup>4</sup> .... Websites like Codecademy and freeCodeCamp offer beginner-friendly tutorials<sup>37</sup> . The Sheryians Coding School YouTube channel aims to teach backend development concepts, including those relevant to security, in an easy-to-understand way<sup>38</sup> .

- 

**Build Hands-On Projects:** Apply what you learn by creating real-world projects that incorporate security best practices<sup>4</sup> .... Start with simple applications and gradually tackle more complex systems, always considering security at each stage. Building a portfolio of these projects can demonstrate your skills to potential employers<sup>39</sup> ....

-



**Stay Updated:** The security landscape is constantly evolving, so continuous learning is crucial<sup>4</sup> .... Follow industry blogs, news, and security-focused resources to stay informed about the latest threats and best practices<sup>4</sup> ....

**Engage with the Developer Community:** Participate in online communities like Stack Overflow or Reddit to ask questions, learn from others, and stay updated on security discussions<sup>4</sup> .... By following these steps, KG and anyone interested can start building a strong foundation in backend security. Remember that **security should be integrated into all aspects of backend development**, not treated as an afterthought<sup>2</sup> ....

keep\_pin  
Save to note

copy\_all

thumb\_up

thumb\_down

i want a better understanding about backend devolment

To gain a better understanding of backend development, let's delve into what it entails based on the information provided in the sources.

## **What is Backend Development?**

Backend development refers to the **server-side of web and software applications**<sup>1</sup> .... It focuses on everything you **can't directly see** on a website or app, such as **databases, servers, and the application's logic**<sup>4</sup> .... Backend developers ensure that websites and applications function smoothly behind the scenes<sup>1</sup> .... They write code that connects the **user interface (frontend)** to the **database** and manage crucial tasks like **user authentication, data storage, and system security**<sup>1</sup> .... Essentially, backend development creates the **foundation that allows web applications to run smoothly, securely, and efficiently**<sup>8</sup> .

## **What Does a Backend Developer Do?**

Backend developers are the **backbone of every application**<sup>6</sup> . They manage processes behind the scenes, ensuring websites and apps function correctly<sup>6</sup> . Their key responsibilities include:

- 

**Designing and maintaining servers and databases**<sup>9</sup> . This involves configuring and managing servers (physical or cloud-based) and designing data models for efficient data storage and retrieval<sup>10</sup> .

- 

**Developing APIs (Application Programming Interfaces) to enable communication between systems**<sup>9</sup> .... APIs act as bridges, allowing the frontend to access server-side data and functionality<sup>11</sup> . This communication between the frontend and backend is crucial for a fully functional application<sup>12</sup> ....

- 

**Ensuring applications are secure, efficient, and scalable**<sup>9</sup> . Security involves implementing measures like data encryption and user authentication<sup>10</sup> .... Efficiency relates to the website performing quickly and responsively<sup>15</sup> . Scalability means the system can handle increasing demand<sup>16</sup> ....

- 

**Collaborating with frontend developers** to create seamless user experiences<sup>9</sup> .... Backend developers need to understand the requirements of the frontend team and provide the necessary support<sup>18</sup> ....

-

**Monitoring and optimizing server performance to prevent downtime**<sup>9</sup> .

•

**Writing high-quality and maintainable code**<sup>21</sup> . This ensures the application is sustainable and easy to update<sup>21</sup> ....

•

**Troubleshooting and debugging issues**<sup>21</sup> ....

•

Sometimes, **training and supporting junior developers**<sup>24</sup> .

**Essential Backend Developer Skills:**

According to one source, there are eight essential skills for backend developers<sup>25</sup> ...:

1.

**Programming Languages:** Mastery of languages like Python, Java, and JavaScript (Node.js) is fundamental<sup>3</sup> .... These languages are used to write the server-side logic<sup>10</sup> .

2.

**Database Management:** Understanding how to design, manage, and interact with databases (both SQL like MySQL and PostgreSQL, and NoSQL like MongoDB) is crucial for efficient data storage and retrieval<sup>10</sup> .... Knowing SQL is often a minimum requirement<sup>32</sup> .

3.

**API Development:** The ability to design, build, and secure APIs is essential for communication between different parts of an application<sup>10</sup> ....

4.

**Development Frameworks:** Frameworks like Django (for Python), Spring Boot (for Java), and Express.js (for Node.js) simplify the development process by providing tools and structure<sup>10</sup> ....

5.

**Security Skills:** Implementing security measures to protect user data and prevent vulnerabilities is a critical aspect of backend development<sup>10</sup> .... This aligns with KG's plan to learn more about backend security<sup>37</sup> .

6.

**Cloud Computing:** Familiarity with cloud platforms and serverless architectures is increasingly important for scalability and cost efficiency<sup>25</sup> ....

7.

**Development Tools:** Mastering tools for version control (like Git<sup>40</sup> ...), debugging, and testing is necessary for efficient development workflows<sup>22</sup> ....

8.

**Soft Skills:** Interpersonal abilities like problem-solving, communication, time management, teamwork, and adaptability are also essential for effective collaboration and navigating the dynamic tech landscape<sup>42</sup> ....

**Backend vs. Frontend:**

It's important to understand the distinction between backend and frontend development. Frontend development focuses on the **user-facing side** of applications – everything users can see and interact with<sup>2</sup> .... The backend, on the other hand, supports the internal infrastructure and operates behind the scenes<sup>2</sup> .... They have a **sybiotic relationship**, depending on each other for the application to function correctly<sup>13</sup> .

**Challenges for Backend Developers in 2025:**

The field of backend development is constantly evolving, presenting several challenges<sup>47</sup> ...:

- 

**Keeping up with new technologies:** The rapid pace of technological advancements requires continuous learning<sup>16</sup> ....

- 

**Scalability and performance optimization:** Designing systems that can handle increasing user loads and maintain optimal performance is a significant challenge<sup>16</sup> ....

- 

**Ensuring data privacy and security:** Protecting sensitive data against evolving security threats is a paramount concern<sup>14</sup> ....

- 

**Integration of AI and Machine Learning:** Incorporating these technologies into applications requires specialized knowledge and collaboration<sup>49</sup> ....

- 

**Cloud computing and serverless architectures:** Adapting to new paradigms of building and managing applications in the cloud is necessary<sup>38</sup> ....

-

**Managing microservices and distributed systems:** As applications become more complex, managing these architectures presents its own set of challenges17 ....

### **How to Start Learning Backend Development:**

Several sources offer guidance on how to begin your journey in backend development26 ...:

- 

**Master a programming language:** Choose a language like Python, Java, or JavaScript and focus on building a strong foundation26 .... Python is often recommended for beginners due to its readability25 ....

- 

**Learn database management:** Understand database concepts and gain practical experience with SQL and NoSQL databases10 ....

- 

**Practice API development:** Learn how to build and consume APIs10 ....

- 

**Study frameworks:** Familiarize yourself with popular backend frameworks for your chosen language20 ....

- 

**Implement security practices:** Start learning about common security vulnerabilities and how to prevent them10 ....

- 

**Learn cloud computing basics:** Understand cloud service models and deployment26 ....

- 

**Master development tools:** Get comfortable with Git for version control and other essential tools26 ....

- 

**Develop soft skills:** Work on your communication, problem-solving, and teamwork abilities26 ....

- 

**Build hands-on projects:** Apply what you learn by creating real-world applications to solidify your understanding and build a portfolio26 ....

- 

**Utilize learning resources:** Enroll in online courses, bootcamps, and use free resources like YouTube tutorials and coding platforms26 .... The Sheryians Coding School YouTube channel, featuring Harsh, aims to teach backend concepts in an accessible way76 .... The "Self Taught Courses"

YouTube channel also offers a full course on databases for backend development<sup>78</sup>.

- 

**Join developer communities:** Engage with other developers online to ask questions and learn from their experiences<sup>26</sup> ...

- 

**Focus on continuous learning:** The tech landscape is always changing, so stay updated with the latest trends and technologies<sup>16</sup> ....

By understanding these fundamental aspects, you can gain a solid foundation in backend development and appreciate its critical role in the functioning of modern web applications.

keep\_pin

Save to note

copy\_all

thumb\_up

thumb\_down

what are the top 5 entry-level job titles in my chosen field and their average starting salaries across different regions?

Based on the sources and our conversation history where you mentioned your plan to pursue a career in backend security<sup>1</sup>, it's important to understand the entry points into the broader field of backend development, as dedicated "entry-level backend security" roles might be less common than security responsibilities being part of junior backend roles.

While the sources don't explicitly list the "top 5 entry-level job titles in backend security" with average starting salaries across different regions, we can infer some common entry-level positions in the backend development field and their potential salary ranges, where security responsibilities would be learned and applied.

Here are some likely entry-level job titles based on the sources:

- 

**Junior Back-End Developer:** This is a common starting point where you would be responsible for server-side code and APIs, database functionalities, and contributing to system security<sup>2</sup> .... A junior backend engineer plays a crucial role in designing and building the server-side logic and ensuring data security and protection<sup>3</sup>.

- 

**Associate Software Engineer (Backend Focus):** Similar to a Junior Back-End Developer, this role emphasizes software development with a focus on the backend. Entry-level back-end engineers need to write viable code, have troubleshooting skills, and understand software network configurations<sup>4</sup>.

- 

**Trainee/Graduate Developer (Backend):** Some companies offer trainee or graduate programs that focus on backend development, providing structured learning and mentorship.

- 

**Junior Database Developer:** Since databases are a critical part of the backend and security, a role focusing on designing and maintaining databases, often using SQL or NoSQL, could be an entry point<sup>5</sup> ....

- 

**Application Developer (Backend Focus):** This title suggests a focus on building the server-side logic of applications, including aspects of data handling and security.

Regarding average starting salaries across different regions, the sources provide some general figures but not specifically broken down for entry-level roles in different geographical areas for these exact titles.

- In the **US**, average salaries for backend developers range from **\$88,000 to \$150,000**<sup>7</sup>. Another source mentions an estimated yearly salary for a backend developer in the US of **\$115,000** as of January 2025<sup>8</sup>. These are general averages and starting salaries would likely be at the lower end of this range. "Interview Kickstart" indicates an average salary for an **Entry-Level Back-end Engineer in New York** as **\$73,318 per year**<sup>9</sup>.

- The "Backend Developers: The Greatest Challenges | GUVI-Blogs" source mentions looking at the **Backend Developer Salary in India & USA**<sup>10</sup>, suggesting salary variations across these regions. However, specific entry-level figures for India are not provided in the given excerpts.

- "Developer Roadmaps" mentions that backend developer salaries can vary depending on location, experience, and company size, with an average base salary in the **United States around \$92,000 per year**. It also notes higher-paying cities like San Francisco, Seattle, and New York having average salaries of **\$120,000 to \$135,000 per year**, again as general averages<sup>11</sup>.

It's important to note that these salary figures are general for backend developers and not specifically for "backend security" entry-level titles. As you gain experience and specialize in security, your job titles and salary expectations would evolve.

To get more specific information on entry-level backend security roles and their salaries in different regions, you might need to consult job boards that allow filtering by entry-level positions and location, as well as conduct more focused research on security-specific career paths within backend development.

keep\_pin

Save to note



copy\_all

thumb\_up

thumb\_down

what are the top 5 entry-level job titles in my chosen field and their average starting salaries across different regions?

Drawing on the sources and our previous conversation where you mentioned your plan to learn more about backend security<sup>1</sup>, here are 5 potential entry-level job titles in the broader field of backend development (as dedicated entry-level backend security roles might be less common initially) along with their average starting salaries where the sources provide information across different regions:

1.

**Junior Back-End Developer:** This role involves responsibility for server-side code and APIs<sup>2</sup>. A junior back-end engineer needs to write viable code, possess troubleshooting skills, and understand software network configurations<sup>3</sup> ...

○

In the **US**, average salaries for backend developers range from **\$88,000 to \$150,000**<sup>5</sup>.

○

Another source estimates the yearly salary for a back-end developer in the US at **\$115,000** as of January 2025<sup>6</sup>.

○  
For an **Entry-Level Back-end Engineer in New York**, the average salary is **\$73,318 per year**<sup>7</sup>.

○  
The average base salary for a backend developer in the **United States** is around **\$92,000 per year**<sup>8</sup>. Higher-paying cities like San Francisco, Seattle, and New York have average salaries of **\$120,000 to \$135,000 per year** for backend developers in general<sup>8</sup>.

2. **Associate Software Engineer (Backend Focus):** This is similar to a Junior Back-End Developer, with an emphasis on software development on the server side<sup>3</sup>. Entry-level back-end engineers need to be able to write code and troubleshoot problems<sup>3</sup> ... Salary information for this specific entry-level title is not detailed separately in the sources but would likely be similar to that of a Junior Back-End Developer.

3. **Trainee/Graduate Developer (Backend):** Some companies offer structured programs for new developers focusing on the backend<sup>9</sup>. These programs often involve learning on the job. Specific salary information for these roles is not broadly available in the sources, as it can vary significantly by company and location.

4. **Junior Database Developer:** Database design and maintenance are crucial aspects of backend development and security<sup>10</sup> ... Database developers specialize in this area, often working with SQL or NoSQL databases<sup>12</sup> ....

○  
According to Salary.com, the average salary for a Database Developer in the **US** is **\$98,664.00**<sup>14</sup> ... Entry-level salaries would likely be lower than this average.

5. **Application Developer (Backend Focus):** This title suggests a role focused on building the server-side logic of applications, which includes data handling and security considerations<sup>3</sup>. Similar to Associate Software Engineer, specific entry-level salary details for this title across different regions are not explicitly provided in the sources but would align with the general entry-level backend developer salary ranges.

It's important to note that the sources provide general salary ranges for backend developers and specific entry-level figures are limited to certain locations like New York<sup>5</sup> ... While one source mentions "Backend Developer Salary in India & USA"<sup>16</sup> and another notes an "Average Fresher Salary" in **India** for back-end development as **4 LPA (Lakhs Per Annum)**<sup>10</sup>, detailed breakdowns by entry-level job title across different regions are not consistently available within these sources.

As you progress in your career with a focus on backend security, your job titles might evolve to be more security-specific, such as Junior Security Engineer<sup>17</sup>, but the initial entry point is often through these broader backend development roles where you will learn and apply security principles<sup>3</sup> ....

keep\_pin  
Save to note

copy\_all

thumb\_up

thumb\_down

how much would i make per hour

Based on the sources and our conversation history where you mentioned your plan to pursue a career in backend security<sup>1</sup>, the sources provide information on annual salaries for backend developers, which is a likely entry point into roles with security responsibilities. However, the sources **do not explicitly state hourly rates** for entry-level positions in this field. Here's a breakdown of what the sources indicate about potential earnings on an annual basis, which we can then use to estimate a rough hourly rate:

- In the **US**, average salaries for backend developers range from **\$88,000 to \$150,000 per year**<sup>2</sup>. Another source estimates the yearly salary for a backend developer in the US at **\$115,000** as of January 2025<sup>3</sup>.

- For an **Entry-Level Back-end Engineer in New York**, the average salary is **\$73,318 per year**<sup>4</sup>.

- The average base salary for a backend developer in the **United States** is around **\$92,000 per year**<sup>5</sup>. In higher-paying cities like San Francisco, Seattle, and New York, the average salary can range from **\$120,000 to \$135,000 per year** for backend developers in general<sup>5</sup>.

- In **India**, the average fresher salary for a back-end developer is **4 LPA (Lakhs Per Annum)**<sup>6</sup>.

To estimate an hourly rate from these annual salaries, we can make a rough calculation assuming a standard full-time work year (e.g., 40 hours per week for 52 weeks, totaling 2080 hours).

- Based on the New York entry-level salary of \$73,318 per year, a rough hourly estimate would be  $\$73,318 / 2080 \text{ hours} \approx \textbf{\$35.25 per hour}$ .

- Using the lower end of the US average range (\$88,000 per year), the hourly estimate would be  $\$88,000 / 2080 \text{ hours} \approx \textbf{\$42.31 per hour}$ .

- Using the higher end of the US average range (\$150,000 per year), the hourly estimate would be  $\$150,000 / 2080 \text{ hours} \approx \textbf{\$72.12 per hour}$ .

- Based on the overall US average of \$92,000 per year, the hourly estimate would be  $\$92,000 / 2080 \text{ hours} \approx \textbf{\$44.23 per hour}$ .

- Using the estimated yearly salary of \$115,000<sup>3</sup>, the hourly estimate would be  $\$115,000 / 2080 \text{ hours} \approx \$55.29 \text{ per hour}$ .

**It is crucial to understand that these are very rough estimates.** Actual hourly rates for entry-level positions, especially those with a security focus, can vary significantly based on several factors not fully detailed in the sources, including:

- **Specific job title and responsibilities:** Dedicated "entry-level backend security" roles might have different pay scales compared to general junior backend developer roles.

- **Your location:** The cost of living and demand for developers can significantly impact pay rates across different regions and even within cities<sup>5</sup>.

- **Company size and type:** Larger companies or those in specific industries might offer higher compensation.

- **Your skills and experience:** Even at the entry level, specific skills or prior experience (e.g., internships, personal projects with a security focus) could influence your starting pay.

To get a more accurate understanding of potential hourly rates for entry-level roles in backend security, it is recommended to:

- **Research specific job titles** on job boards in your desired locations. Many listings may include hourly rates or salary ranges.

- **Network with professionals** in the field to gain insights into typical compensation for entry-level positions in your area.

- Consider looking at salary surveys that might provide more granular data for specific entry-level security-related roles, although the sources provided do not contain this level of detail.

keep\_pin  
Save to note

copy\_all

thumb\_up

thumb\_down

whats missing from my research that would help me make a more informed decision about this career path

Drawing on the sources and our conversation history where you expressed your plan to learn more about backend security and make a career out of it<sup>1</sup>, there are several areas where further research could help you make a more informed decision about this career path. While the sources provide a good overview of backend development and touch upon security, a deeper dive into the following aspects would be beneficial:

- 

### **The Specific Landscape of "Backend Security" as an Entry-Level**

**Role:** The sources discuss entry-level *backend developer* roles<sup>2</sup> ... and the responsibilities of a *Junior Security Engineer*<sup>5</sup> ... However, dedicated entry-level "backend security engineer" positions might be less common initially. Understanding how security responsibilities are integrated into junior backend roles versus stand-alone junior security roles would be valuable. You could research if companies typically hire directly for junior backend

security roles or if it's more common to specialize in security after gaining experience as a backend developer.

- 

### **Detailed Skill Requirements for Backend Roles with a Security Focus:**

While general backend skills like knowing programming languages (JavaScript, Python, Java are mentioned<sup>7</sup> ...), databases (SQL, NoSQL like MongoDB<sup>9</sup> ...), and web frameworks (Django, Spring Boot, Express.js<sup>7</sup> ...) are important<sup>12</sup> ..., understanding the **specific security skills** expected even in entry-level backend roles would be helpful. The "Responsibilities: Junior Backend Engineer" source mentions ensuring data security and protection<sup>16</sup> and being familiar with techniques like input validation, output encoding, and parameterized queries to prevent common security risks like SQL injection, XSS, and CSRF<sup>17</sup>. Researching other common backend security vulnerabilities and the practices to mitigate them, as mentioned in the Reddit thread on learning backend security from scratch (like CSRF, parameter pollution, cache poisoning<sup>18</sup>) and the OWASP Top Ten<sup>19</sup>, would be crucial.

- 

**Career Progression Specifically within Backend Security:** The "Career Roadmap of a Back-end Engineer"<sup>2</sup> and "Potential Career Advancements" for a Junior Security Engineer<sup>20</sup> offer general paths. However, understanding how a backend developer can specialize and progress within the security domain (e.g., moving from a junior backend role to a security-focused role, or progressing within a security engineering team with a backend specialization) would provide a clearer long-term vision.

- 

**Day-to-Day Responsibilities of Entry-Level Backend Roles with Security Aspects:** The descriptions of junior backend engineers mention tasks like writing code, performing development tests, and addressing minor bugs<sup>2</sup> ... Gaining insights into how security considerations fit into these daily tasks (e.g., incorporating security testing into development, writing secure code, participating in security code reviews) would offer a more realistic picture.

- 

**Industry-Recognized Security Certifications Relevant to Backend Development:** The "Responsibilities: Junior Security Engineer" source mentions certifications like CompTIA Security+, CEH, or CISSP<sup>21</sup>. Researching which security certifications are most valued or relevant for

backend developers who want to specialize in security, even at the entry level, could guide your learning path.

- 

### **Practical Application of Security Principles in Backend Development:**

Understanding how general security principles (like confidentiality, integrity, and availability) are specifically applied in the context of backend development (e.g., secure data storage, secure API design, secure communication protocols) would bridge the gap between general security knowledge and its practical implementation in your chosen field.

- 

### **Common Security Challenges and Mistakes in Backend Development:**

The "Top 10 Mistakes Backend Developers Make in 2025"<sup>22</sup> highlights issues like missing automatic database backups<sup>22</sup>. Further research into common security-related mistakes that backend developers make and how to avoid them would be highly valuable.

By exploring these areas, you can gain a more nuanced and practical understanding of what a career in backend security entails at the entry level and how to best prepare for it. Remember that continuous learning and staying updated with the latest security trends and vulnerabilities are essential in this field<sup>23</sup> ....

keep\_pin

Save to note

copy\_all

thumb\_up



thumb\_down

can you explain api to me

An **API**, which stands for **Application Programming Interface**, is fundamentally a **set of rules and definitions that allow different software applications to communicate and interact with each other**1 .... Think of it as a contract or an interface that one piece of software exposes so that other software can use its functionality or data1 .... In the context of web development, which is likely what you're most interested in given our focus on backend security, an API acts as the **communication bridge between the front-end (what users see and interact with in a web browser) and the back-end (the server-side logic, databases, and infrastructure that power the application)**1 .... Here's a breakdown of key aspects of APIs in web development:

- 

**Interface for Applications:** The back-end has an interface (the API) that is used by other applications, typically the front-end application, to talk to it15 .... This interface is not directly used by the end-user15 .

- 

**Requests and Responses:** When the front-end needs data or wants to perform an action on the server, it sends a **request** to the back-end through the API16 .... The API defines how these requests should be formatted (e.g., what data to include)3 .... The back-end then processes the request and sends back a **response**, which often includes the requested data or confirmation of the action16 ....

- 

**HTTP Protocol:** In web-based APIs, communication often happens using the **HTTP (Hypertext Transfer Protocol)**, which is the foundation of data communication on the internet27 .... Common HTTP methods or **verbs** used in APIs include **GET** (to retrieve data), **POST** (to create new data), **PUT** (to update existing data), and **DELETE** (to remove data)27 .... These

operations are often related to **CRUD** (Create, Read, Update, Delete) functionalities for data<sup>29</sup>.

- 

**Endpoints (URLs):** APIs expose specific **endpoints**, which are essentially web addresses (URLs) that the front-end can send requests to in order to access different functionalities or resources on the back-end<sup>4</sup>.... For example, an API for a blog might have an endpoint like `/api/posts` to get all blog posts or `/api/users/{id}` to get information about a specific user<sup>20</sup>....

- 

**Abstraction and Simplification:** APIs **hide the complex inner workings of the back-end** from the front-end developer<sup>8</sup>.... The front-end only needs to know how to make requests to specific API endpoints and understand the format of the responses, without needing to understand the underlying database queries or server-side logic<sup>2</sup>.... This promotes modularity and makes development easier<sup>36</sup>....

- 

**Standardized Communication:** APIs provide a **standardized way for different systems or programs to talk to each other**<sup>2</sup>.... This allows developers to build applications that can easily integrate with other services without needing to know the specifics of how those services are implemented<sup>3</sup>....

- 

**Types of APIs:** There are different types of APIs based on their architecture and how they communicate. Two common types are **REST APIs** and **GraphQL APIs**<sup>30</sup>....

- 

**REST (Representational State Transfer)** is an architectural style that uses standard HTTP methods to interact with resources identified by URLs<sup>30</sup>.... It's widely adopted and leverages the existing web infrastructure<sup>27</sup>....

- 

**GraphQL** is a query language for your API that allows clients to request exactly the data they need, making data fetching more efficient<sup>34</sup>....

- 

**Security:** **API security** is crucial to protect APIs from unauthorized access, data breaches, and cyber threats<sup>45</sup>.... It involves measures to ensure the **confidentiality, integrity, and availability** of data transmitted and

accessed through APIs<sup>45</sup>. Common API cyberattacks include Man-in-the-Middle (MitM) attacks, brute force attacks, API scanning, data scraping, content spoofing, and API abuse<sup>46</sup>. Best practices for securing APIs include encrypting communication via HTTPS<sup>48</sup>. In simpler terms, imagine a restaurant<sup>36</sup> ... You (the front-end) want to order food (data or an action). You communicate your order to the waiter (the API) using a menu (the API documentation). The waiter takes your order to the chef (the back-end), who prepares the food. The waiter then brings the food back to you (the API response). You don't need to know how the chef cooks the food or where the ingredients come from; you only need to know how to order from the menu. Understanding APIs is fundamental to backend development and, as you are interested in backend security, you'll learn how to design and secure these interfaces to protect the sensitive data and functionalities they expose<sup>45</sup> ...

keep\_pin  
Save to note

copy\_all

thumb\_up

thumb\_down

top 5 entry level jobs

Based on the sources provided, here are five prominent entry-level job types in the programming and backend development fields:

- 

**Backend Engineer Intern:** Several companies are offering **remote entry-level backend engineer internships**<sup>1</sup> .... These internships provide opportunities to gain hands-on experience in building the server-side logic of applications<sup>13</sup>. Responsibilities may include designing and developing MVPs<sup>2</sup>, supporting AI-driven data solutions<sup>2</sup>, backend development for SaaS startups<sup>3</sup>, and focusing on AWS and Shopify integration<sup>5</sup>. Some internships may not require a degree<sup>3</sup> .... The average salary for remote entry-level backend engineers (with 0 years of experience) is **\$98,525** based on data from 4 job openings<sup>14</sup>.

- 

**Junior Developer:** This role involves designing and developing applications and programs, typically under the supervision of more senior developers<sup>15</sup>. There are several types of software developers, including those specializing in the **back-end**<sup>16</sup> ... and **full-stack**<sup>16</sup>. A **Junior Back-End Developer** is responsible for overseeing the development and maintenance of the server side of an app or website, working closely with front-end developers, and creating, maintaining, testing, and debugging all back-end web applications<sup>17</sup>. They often have experience with programming languages like Java, Ruby, PHP, or Python and a basic understanding of front-end technologies<sup>18</sup>. The average annual US salary for a **Junior developer** is **\$105,060**<sup>16</sup>. The job outlook for software developers is projected to grow by **17 percent from 2023 to 2033**<sup>16</sup>.

- 

**Programmer Analyst:** A **programmer analyst** develops and maintains software, often working on business applications<sup>19</sup>. Their tasks can include designing programs, testing software, debugging, writing, and analyzing code, and translating business requirements into technical specifications<sup>19</sup>. Employers may prefer candidates with a bachelor's degree in computer science, but a combination of relevant work experience and Professional Certificates may also be accepted<sup>20</sup>. The average annual

US salary for a **programmer analyst** is **\$101,505**<sup>20</sup>. The job outlook for programmer analysts is projected to grow by **11 percent from 2023 to 2033**<sup>20</sup>.

• **Web Developer (Entry-Level/Junior):** While the source<sup>15</sup> specifically mentions "Junior developer" and describes full-stack developers, the overall context of entry-level programming jobs implies that **entry-level web developer** positions exist. These roles would likely involve working on either the front-end or back-end of web applications, depending on specialization. The job outlook for **web developers and digital designers** is expected to grow by **8 percent between 2023 and 2033**<sup>15</sup> ... Relevant skills include interpersonal skills, creativity, technical proficiency, and attention to detail<sup>15</sup>.

• **Full-stack Developer (Junior):** **Junior full-stack developers** work on both the front and back end of an application, possessing a wide variety of skills to build and design all aspects of an application<sup>16</sup>. The average annual US salary for a **junior developer** (which would include full-stack) is **\$105,060**<sup>16</sup>. The job outlook for software developers (including full-stack) is projected to grow by **17 percent from 2023 to 2033**<sup>16</sup>. It's worth noting that the term "entry-level" generally refers to roles requiring the least amount of training or expertise and are ideal for individuals new to the occupational field<sup>22</sup>. Employers may look for a computer science degree or equivalent experience<sup>22</sup>, experience working in a collaborative environment, and knowledge of popular programming languages like Python and JavaScript<sup>23</sup>.

☐ keep\_pin  
Save to note

copy\_all

thumb\_up

thumb\_down

Compare responsibilities between two entry-level roles in my field Backend Engineer Intern:vs Junior Developer what skills are unique to each. Based on the sources, here's a comparison of the responsibilities and unique skills for the entry-level roles of Backend Engineer Intern and Junior Developer:

### **Backend Engineer Intern**

#### **Responsibilities:**

- 

**Designing and developing MVPs** (Minimum Viable Products) for specific platforms, such as Moladin's mobility fintech platform focusing on web applications<sup>1</sup>.

- 

**Supporting AI-driven data solutions** and gaining hands-on experience in a dynamic startup environment<sup>1</sup>.

- 

Contributing to **backend development for SaaS startups**, potentially involving technologies like Bootstrap, Elixir, Erlang, ERP, Firebase, Haskell, Java, JavaScript, Laravel, MySQL, PHP, Python, Redis, Rust, and SQL<sup>2</sup>.

- 

**Creating browser solutions** while gaining significant AI experience as a C++ Developer intern, potentially using Django, Flask, JavaScript, NoSQL, Postgres, React, Redux, and SQL<sup>3</sup>.

- 

Gaining **tech skills in software development remotely**<sup>4</sup>.

- Focusing on **AWS and Shopify integration**<sup>5</sup> .

- Engaging in **backend development and data science** to enhance social connections within a community-based platform<sup>6</sup> .

- Developing planning software using **.NET technologies** for high-profile clients<sup>7</sup> .

- Building an **AI recruiting co-pilot** while learning from experts, potentially using Angular, Cloud, ElasticSearch, Ionic, Java, Kafka, Microservices, MongoDB, MySQL, Node.js, Postgres, Redis, and SQL<sup>8</sup> .

- Focusing on **backend development for scalable solutions in FinTech applications**, potentially using AWS, Cloud, ElasticSearch, JavaScript, Node.js, Postgres, Redis, and TypeScript<sup>9</sup> .

- Developing **scalable backend solutions for AI-powered SaaS platforms**, potentially using Django and Python<sup>10</sup> .

- Developing **software solutions and assisting the team in technological advancement**, potentially using JavaScript and Python<sup>11</sup> .

- Contributing to **innovative projects** as an entry-level Full Stack Java Developer<sup>12</sup> .

- Building **robust server-side components for AI applications**, potentially using Django, Flask, NoSQL, Postgres, and Python<sup>13</sup> .

**Junior Developer (specifically focusing on Junior Back-End Developer)**

**Responsibilities:**

- **Overseeing the development and maintenance of the server side of an app or website**<sup>14</sup> ....

- Working **closely with front-end developers** to improve the visitor experience and ensure system consistency<sup>15</sup> .

**Creating, maintaining, testing, and debugging all back-end web applications<sup>15</sup> ....**

- 

Being involved in the **entire product development lifecycle**, including design, development, deployment, and maintenance of new and existing features<sup>17</sup> .

- 

Writing **reusable and maintainable code<sup>17</sup> .**

- 

Potentially performing **UI tests to optimize performance<sup>17</sup> .**

- 

Developing **server-side logic<sup>18</sup> .**

- 

Developing **APIs<sup>16</sup> ....**

- 

Implementing **database functionalities<sup>18</sup> ....**

- 

Ensuring **data security and protection<sup>18</sup> ....**

- 

Collaborating with the frontend team to ensure **seamless flow of data<sup>23</sup> ....**

- 

Writing **clean and efficient code** in backend languages like Java, Python, or Ruby<sup>25</sup> ....

- 

**Debugging and problem-solving<sup>25</sup> ....**

- 

Integrating **user-facing elements** with server-side logic<sup>28</sup> .

- 

Designing **robust APIs to support mobile and desktop clients<sup>29</sup> .**

- 

Managing and optimizing **scalable distributed systems in the cloud<sup>29</sup> .**

- 

Optimizing **web applications for performance and scalability<sup>29</sup> .**

- 

Developing **automated tests<sup>29</sup> .**

- 

Writing reports mentioning the state of the software application<sup>30</sup> .

-



Performing **development tests**<sup>30</sup> .

- 

Checking the **internal system's performance**<sup>30</sup> .

- 

Studying the **end user's needs and requirements**<sup>30</sup> .

- 

Addressing **minor software bugs**<sup>30</sup> .

**Skills Unique to Each Role (Based on Emphasis in Sources):**

**Backend Engineer Intern:**

- 

**Exposure to a wider variety of specific technologies and platforms** is often highlighted in internship descriptions, suggesting a learning and exploratory focus (e.g., specific combinations of languages and frameworks like Go with IoT, Elixir with ERP, or focus on specific services like AWS and Shopify)<sup>1</sup> ....

- 

Internships often emphasize **gaining hands-on experience** and supporting more senior team members, suggesting a more learning-oriented role with potentially less independent ownership of complete features<sup>1</sup> .

- 

Some internships explicitly state "**No degree required**", suggesting opportunities for individuals without formal computer science degrees<sup>2</sup> ....

- 

The focus on **designing and developing MVPs** in some descriptions indicates a potential involvement in initial product stages<sup>1</sup> .

- 

Internships may be more focused on **specific problem domains** like AI-driven data solutions or FinTech<sup>1</sup> ....

**Junior Developer:**

- 

A more holistic understanding of the entire back-end development **lifecycle** is expected, from design to deployment and maintenance<sup>17</sup> .

- 

**Stronger emphasis on independent code writing, testing, and debugging skills** as a core responsibility<sup>15</sup> ....

-

Expectation of a **good understanding of the entire product development process**<sup>31</sup>.

- 

Basic knowledge of **front-end technologies like HTML, JavaScript, and CSS** is often mentioned as beneficial for collaboration<sup>31</sup> ....

- 

**Technical and consumer awareness** might be expected as they contribute to the user experience indirectly through the backend<sup>31</sup>.

- 

More direct responsibility for ensuring the **stability and quality of web applications**<sup>17</sup>.

- 

A focus on writing code that is **maintainable and scalable**<sup>17</sup> ....

- 

A need to have **analytical and time management skills**<sup>31</sup>.

- 

**Seamless communication with team members** is highlighted as a crucial soft skill<sup>33</sup>.

- 

Understanding of **software network configurations**<sup>34</sup>.

In essence, a **Backend Engineer Intern** role appears to be more focused on learning specific technologies and gaining initial practical experience under guidance, often with a more project-based or support-oriented set of responsibilities. A **Junior Developer** role, on the other hand, involves more direct responsibility for the back-end's functionality, requiring a broader understanding of the development process, stronger coding and problem-solving abilities, and the capacity for more independent work and collaboration with frontend teams. While both are entry-level, the Junior Developer typically has a higher expectation of foundational knowledge and a greater degree of ownership over development tasks.

keep\_pin

Save to note

copy\_all

thumb\_up

thumb\_down

What tools, platforms, and programming languages are most commonly expected in entry-level job descriptions for junior developer?

Based on the sources, entry-level job descriptions for junior developers commonly expect knowledge of certain programming languages, databases, frameworks, and tools<sup>1</sup> ....

#### **Programming Languages:**

- 

**Python** is frequently mentioned as a highly in-demand language, with over 40 percent of recruiters actively seeking candidates with Python skills<sup>2</sup> . It is also recommended for beginners due to its easy and readable syntax<sup>5</sup> .... Several internships also list Python as a required skill<sup>7</sup> .

- 

**JavaScript** is another highly sought-after language, also with over 40 percent of recruiters looking for candidates with this skill<sup>2</sup> . It is used for both frontend and backend development (Node.js)<sup>5</sup> .... Some entry-level backend engineer internships also list JavaScript<sup>7</sup> ....

- 

**Java** is highlighted as a robust language for backend development, particularly for large-scale enterprise applications<sup>5</sup> .... It is often taught in

computer science programs<sup>5</sup> ... Several entry-level backend engineer internships also require or mention Java<sup>7</sup> ...

- 

Other languages mentioned in the context of backend development and potentially entry-level roles include **SQL** (for database interaction)<sup>7</sup> ..., **Go**<sup>5</sup> ..., **Ruby**<sup>5</sup> ..., **PHP**<sup>7</sup> ..., **C++**<sup>5</sup> ..., and **.NET/C#**<sup>5</sup> ...

### Databases:

- 

**SQL** and understanding of **relational databases** like **MySQL** and **PostgreSQL** are commonly expected<sup>1</sup> ... Several entry-level backend engineer internships specifically mention SQL, MySQL, or Postgres<sup>7</sup> ...

- 

Knowledge of **NoSQL databases** like **MongoDB** is also beneficial<sup>1</sup> ... Some entry-level backend engineer internships mention MongoDB<sup>14</sup> .

### Frameworks:

- 

Knowledge of **backend frameworks** relevant to the chosen programming language is important<sup>3</sup> ...

- 

For **Python**, **Django** and **Flask** are mentioned<sup>10</sup> ...

- 

For **Java**, **Spring** or **Spring Boot** are key frameworks<sup>5</sup> ...

- 

For **JavaScript**, **Node.js** and **Express.js** are popular<sup>5</sup> ...

- 

For **PHP**, **Laravel** is mentioned<sup>7</sup> ...

- 

For **Ruby**, **Ruby on Rails** is a significant framework<sup>5</sup> ...

- 

For **.NET/C#**, **ASP.NET Core** is relevant<sup>5</sup> ...

- 

While primarily frontend, basic knowledge of **frontend technologies** like **HTML**, **CSS**, and **JavaScript** is often expected for junior *backend* developers to facilitate collaboration with frontend teams<sup>3</sup> ...

### Tools and Platforms:

- 

**Git** for version control is an essential skill for all developers, including entry-level and junior roles<sup>1</sup> ...

- Familiarity with **APIs** and **web services**, particularly **RESTful APIs**, is generally expected<sup>1</sup> .... Entry-level backend developers will be involved in developing and maintaining APIs<sup>40</sup> ....

- Basic knowledge of **server management** and potentially **cloud computing platforms** like **AWS**, **Azure**, or **Google Cloud Platform** can be beneficial<sup>26</sup> .... Some entry-level internships mention AWS and Cloud<sup>7</sup> ....

- Understanding of **data structures and algorithms** is considered a critical foundation, especially for junior developers<sup>3</sup> ....

- Experience with **development tools** like **Docker** for containerization and **Postman** for API testing can be advantageous<sup>26</sup> ....

In summary, entry-level junior developer job descriptions commonly look for candidates with proficiency in at least one popular backend language like **Python**, **Java**, or **JavaScript**, a good understanding of **SQL** and **relational databases**, familiarity with a relevant **backend framework**, and essential tools like **Git** and knowledge of **APIs**. Basic understanding of frontend technologies and cloud platforms can also be beneficial<sup>1</sup> ....

keep\_pin

Save to note

copy\_all

thumb\_up

thumb\_down

list out all the 47 sources

Here is a list of all 47 sources provided:<sup>1</sup>

•

**Excerpts from "36 Remote Entry-level Backend Engineer Jobs – Apply Now"**<sup>2</sup>

•

**Excerpts from "36 Remote Entry-level Backend Engineer Jobs – Apply Now"** (This appears to be a continuation of the same article)<sup>3</sup>

•

**Excerpts from "5 Entry-Level Programming Jobs + How to Get One | Coursera"**<sup>4</sup>

•

**Excerpts from "5 Entry-Level Programming Jobs + How to Get One | Coursera"** (This appears to be a continuation of the same article and includes citations)<sup>5</sup>

•

**Excerpts from "50 Popular Backend Developer Interview Questions and Answers - Developer Roadmaps"**<sup>6</sup>

•

**Excerpts from "50 Popular Backend Developer Interview Questions and Answers - Developer Roadmaps"** (This appears to be a continuation of the same article)<sup>7</sup>

•

**Excerpts from "8 Best Practices for Securing APIs | NinjaOne"**<sup>8</sup>

•

**Excerpts from "8 Best Practices for Securing APIs | NinjaOne"** (This appears to be a continuation of the same article)<sup>9</sup>

•

• **Excerpts from "8 Trends In Backend Development You Can't Ignore In 2025 - Arunangshu Das Blog"**<sup>10</sup>

• **Excerpts from "A Beginner's Guide to Creating the Best Back-End Developer Portfolio - Pesto Tech"**<sup>11</sup>

• **Excerpts from "A Beginner's Guide to Creating the Best Back-End Developer Portfolio - Pesto Tech"** (This appears to be a continuation of the same article)<sup>12</sup>

• **Excerpts from "A Beginner's Guide to Creating the Best Back-End Developer Portfolio - Pesto Tech"** (This appears to be a continuation of the same article)<sup>13</sup>

• **Excerpts from "A Beginner's Guide to Creating the Best Back-End Developer Portfolio - Pesto Tech"** (This appears to be a continuation of the same article)<sup>14</sup>

• **Excerpts from "API Security Best Practices: Protecting Your Data in Transit - Semaphore"**<sup>15</sup>

• **Excerpts from "API Security Best Practices: Protecting Your Data in Transit - Semaphore"** (This appears to be a continuation of the same article)<sup>16</sup>

• **Excerpts from "API Security Best Practices: Protecting Your Data in Transit - Semaphore"** (This appears to be a continuation of the same article)<sup>17</sup>

• **Excerpts from "API Security Best Practices: Protecting Your Data in Transit - Semaphore"** (This appears to be a continuation of the same article)<sup>18</sup>

• **Excerpts from "Backend learning roadmap : r/webdev - Reddit"**<sup>19</sup>

• **Excerpts from "How to identify and address performance bottlenecks - Product-Led Alliance"**<sup>20</sup>

Excerpts from "How to identify and address performance bottlenecks - Product-Led Alliance" (This appears to be a continuation of the same article)<sup>21</sup>

•

Excerpts from "How to identify and address performance bottlenecks - Product-Led Alliance" (This appears to be a continuation of the same article)<sup>22</sup>

•

Excerpts from "I'm soo confused about where and why API's come into play : r/learnprogramming - Reddit"<sup>23</sup>

•

Excerpts from "I'm soo confused about where and why API's come into play : r/learnprogramming - Reddit" (This appears to be a continuation of the same discussion)<sup>24</sup>

•

Excerpts from "I'm soo confused about where and why API's come into play : r/learnprogramming - Reddit" (This appears to be a continuation of the same discussion)<sup>25</sup>

•

Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"<sup>26</sup>

•

Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses" (This appears to be a continuation of the same transcript)<sup>27</sup>

•

Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses" (This appears to be a continuation of the same transcript)<sup>28</sup>

•

Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses" (This appears to be a continuation of the same transcript)<sup>29</sup>



•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**30**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**31**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**32**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**33**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**34**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**35**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**36**

•  
**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End**

**TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**37**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**38**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**39**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**40**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**41**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**42**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**43**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**44**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**45**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**46**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**47**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**48**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**49**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**50**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**51**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End**

**TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**52**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**53**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**54**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**55**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**56**

•

**Excerpts from the transcript of the video "Introduction to Databases for Back-End Development FULL COURSE II Databases for Back-End TUTORIAL" uploaded on the YouTube channel "Self Taught Courses"**  
(This appears to be a continuation of the same transcript)**57**

•

**Excerpts from "Junior Back-End Developer Job Description [Template] - Homerun"****58**

•

**Excerpts from "Learn Backend Development for Free: Top Websites to Get You Started - DEV Community"****59**

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"****60**

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[61](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource)[62](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[63](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[64](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image related to frontend)[65](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[66](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[67](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[68](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image related to backend)[69](#)

•

**Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)[70](#)

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource)**71**

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)**72**

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)**73**

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)**74**

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image showing company logos)**75**

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource)**76**

- **Excerpts from "Learning Path- Back End Development - Board Infinity"** (This appears to be a continuation of the same resource, possibly an image)**77**

- **Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School"****78**

- **Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School"** (This appears to be a continuation of the same transcript)**79**

- **Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the**

**YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[80](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[81](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[82](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[83](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[84](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[85](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[86](#)

•

**Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)**[87](#)

•



Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)[88](#)

•

Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)[89](#)

•

Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)[90](#)

•

Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)[91](#)

•

Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)[92](#)

•

Excerpts from the transcript of the video "Master JavaScript Essentials for Backend Development I Part 1" uploaded on the YouTube channel "Sheryians Coding School" (This appears to be a continuation of the same transcript)[93](#)

•

Excerpts from "Microservices Architectures 101: How APIs Drive System Integration - Ambassador Labs"[94](#)

•

Excerpts from "Resources to learn backend security from scratch : r/webdev - Reddit"[95](#)

•



• **Excerpts from "Resources to learn backend security from scratch : r/webdev - Reddit" (This appears to be a continuation of the same discussion)**<sup>96</sup>

• **Excerpts from "Responsibilities: Junior Security Engineer - Remotely"**<sup>97</sup>

• **Excerpts from "Reverse Proxy VS Forward Proxy: Understanding the Difference - Geonode"**<sup>98</sup>

• **Excerpts from "Table of Contents Software Security Principles Overview - Exploit-DB"**<sup>99</sup>

• **Excerpts from "Table of Contents Software Security Principles Overview - Exploit-DB" (This appears to be a continuation of the same document)**<sup>100</sup>

• **Excerpts from "Table of Contents Software Security Principles Overview - Exploit-DB" (This appears to be a continuation of the same document)**<sup>101</sup>

• **Excerpts from "Table of Contents Software Security Principles Overview - Exploit-DB" (This appears to be a continuation of the same document)**<sup>102</sup>

• **Excerpts from "Table of Contents Software Security Principles Overview - Exploit-DB" (This appears to be a continuation of the same document)**<sup>103</sup>

• **Excerpts from "The 5 Best Backend Development Languages to Master (2025)"**<sup>104</sup>

• **Excerpts from "The 5 Best Backend Development Languages to Master (2025)" (This appears to be a continuation of the same article)**<sup>105</sup>

• **Excerpts from "Top 10 Mistakes Backend Developers Make in 2025 - Brainhub"**<sup>106</sup>

• **Excerpts from "Top 10 Mistakes Backend Developers Make in 2025 - Brainhub" (This appears to be a continuation of the same article)**[107](#)

• **Excerpts from "Top 10 Mistakes Backend Developers Make in 2025 - Brainhub" (This appears to be a continuation of the same article)**[108](#)

• **Excerpts from "Top 10 Mistakes Backend Developers Make in 2025 - Brainhub" (This appears to be a continuation of the same article)**[109](#)

• **Excerpts from "Types of APIs | Types Of API Calls & REST API Protocol - Stoplight"**[110](#)

• **Excerpts from "Types of APIs | Types Of API Calls & REST API Protocol - Stoplight" (This appears to be a continuation of the same article)**[111](#)

• **Excerpts from "What Is A Database Developer? Required Skills, Salary, & More! - Skillcrush"**[112](#)

• **Excerpts from "What Is A Database Developer? Required Skills, Salary, & More! - Skillcrush" (This appears to be a continuation of the same article)**[113](#)

• **Excerpts from "What is Backend Development? - Developer Roadmaps"**[114](#)

• **Excerpts from "What is version control | Atlassian Git Tutorial"**[115](#)

• **Excerpts from "What is version control | Atlassian Git Tutorial" (This appears to be a continuation of the same tutorial)**[116](#)

• **Excerpts from "the complete back end developer roadmap - Code with Mosh"**[117](#)

• **Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)**[118](#)

Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)**119**

•

Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)**120**

•

Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)**121**

•

Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)**122**

•

Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)**123**

•

Excerpts from "the complete back end developer roadmap - Code with Mosh" (This appears to be a continuation of the same roadmap)

keep\_pin

Save to note

copy\_all

thumb\_up

thumb\_down

give me the summary off all we talked will now including all the specifications and details

Based on the sources provided, here is a summary of the key aspects of backend development and related concepts that we have likely discussed:

## APIs (Application Programming Interfaces)

- 

An API is an interface that allows different software applications to communicate and interact with each other<sup>1</sup> .... For web applications, this often involves **REST APIs** that utilize HTTP<sup>4</sup> ....

- 

A **REST API** (Representational State Transfer) is a common type of API<sup>5</sup> . RESTful APIs have several key characteristics:

- 

Resources are identified by **URIs (Unique Resource Identification)**<sup>6</sup> .

- 

Resources are modified through their representation<sup>6</sup> .

- 

Messages are **self-descriptive**, containing enough information to understand their processing<sup>6</sup> .

- 

Clients can discover available actions for a resource using the server's response (**HATEOAS - Hypermedia as the Engine of Application State**)<sup>6</sup>

- 

- 

Requests are **stateless**, meaning each request contains all necessary information for processing<sup>6</sup> .

- 

It should be a **layered system** where intermediaries don't affect client-server communication<sup>6</sup> .

- 

Responses should be **cacheable**<sup>6</sup> .

-

Optionally, the server can send code to the client for execution (**Code on Demand**)<sup>6</sup>.

- 

The **HTTP request/response cycle** is fundamental to how web APIs function<sup>7</sup>:

- 

**Open the connection:** The client establishes a TCP connection to the server (port 80 for HTTP, 443 for HTTPS)<sup>7</sup>.

- 

**Send the request:** The client sends an HTTP request containing:

- 

An **HTTP method** (e.g., GET, POST, PUT, DELETE)<sup>7</sup>.

- 

A **URI** specifying the resource location<sup>7</sup>.

- 

The **HTTP version** (e.g., HTTP/1.1, HTTP/2)<sup>7</sup>.

- 

**Headers** providing extra request-related data<sup>7</sup>.

- 

An optional **body** containing data<sup>7</sup>.

- 

**Request processed by the server:** The server handles the incoming request<sup>7</sup>.

- 

**Send the HTTP response:** The server sends back a response containing:

- 

The **HTTP Version**<sup>8</sup>.

- 

A **status code** indicating the request's outcome<sup>8</sup>.

- 

**Headers** with extra response-related data<sup>8</sup>.

- 

An optional **body** containing data<sup>8</sup>.

- 

**The connection is closed:** Usually the final step, though newer protocols allow keeping the channel open<sup>8</sup>.

-

APIs can be **synchronous** (requiring a wait for a response) or **asynchronous** (responses may come later)<sup>5</sup>. Some APIs may not reply, requiring periodic checks<sup>5</sup>.

- **API design** involves defining resources, data formats, relationships, and available methods, often captured in an **API definition** using specifications like OpenAPI<sup>9</sup>. Documentation is a crucial part of API design<sup>10</sup>.

- **API security** is essential for protecting data in transit<sup>11</sup>.

## **Backend Development Fundamentals**

- Backend development involves creating the server-side logic of applications, handling databases, and ensuring data integrity<sup>12</sup> ....

- Key aspects include working with programming languages, frameworks, and databases<sup>14</sup>.

- **Eventual consistency** is a consistency model in distributed systems where data will become consistent across all servers eventually, requiring data synchronization and potential conflict resolution<sup>15</sup>.

- Effective **backend to frontend communication**, often through APIs, is crucial for service development<sup>16</sup>.

## **Databases**

- A **database** is an electronic storage system where data is organized systematically for efficient management and retrieval<sup>12</sup> .... Data within a database can be related<sup>12</sup> ....

- Common database actions include storing, connecting, filtering, searching, updating, and deleting data (**CRUD operations**)<sup>12</sup> ....

- **Relational databases** organize data into **tables** (also known as entities or relations) with **rows** (records) and **columns** (fields or attributes)<sup>21</sup> ....

- Tables can be related using **keys**<sup>23</sup>.

- Each column has a unique name and a **data type** (e.g., numeric, string, date/time) that defines the type of values it can hold<sup>24</sup> ....

- A **primary key** uniquely identifies each row in a table24 ...

- A **foreign key** in one table references the primary key of another table, establishing relationships between tables27 ...

- **SQL (Structured Query Language)** is a standard language used to interact with relational databases18 ... SQL includes subsets like:

- **DDL (Data Definition Language)** for defining database schema and objects (e.g., CREATE, ALTER, DROP)19 .

- **DML (Data Manipulation Language)** for manipulating data (e.g., INSERT, UPDATE, DELETE)19 ...

- **DQL (Data Query Language)** for querying data (e.g., SELECT)19 ...

- **DCL (Data Control Language)** for controlling data access and security19 ...

- **Database schema** is the organization or blueprint of how data is stored and related within a database34 ... It can be logical (how data is organized into tables and their relationships) or physical (how data is physically stored)37 .

- **Database normalization** is the process of organizing database tables to minimize data redundancy and improve data integrity38 ... It involves different normal forms (1NF, 2NF, 3NF) that address issues like atomicity, repeating groups, functional dependency, partial dependency, and transitive dependency38 ...

## **Backend Development Roadmap and Trends**

- Becoming a backend developer involves learning various technologies beyond just a programming language, including databases, version control (Git), and understanding client-server communication14 .

- Online courses and learning platforms can be valuable resources for backend development education43 .

- **Versioning** of APIs is crucial as they evolve, with semantic versioning being an effective approach to communicate changes (MAJOR for breaking changes, MINOR for backward-compatible enhancements)<sup>44</sup>.

- **Self-documenting APIs** using tools like Swagger/OpenAPI improve usability<sup>44</sup>.

### **Version Control (Git)**

- **Version control** systems (like Git) track changes to files over time, allowing for collaboration, reverting to previous versions, and understanding the history of code modifications<sup>45</sup> ...

- Key benefits include a complete long-term change history, branching and merging capabilities, and traceability of changes<sup>46</sup> ...

### **Junior Backend Developer Skills and Responsibilities**

- A junior backend developer should be comfortable with the programming language and frameworks used by the team<sup>48</sup>.

- **Git** is an essential skill for version control<sup>48</sup>.

- Understanding of **SQL and databases** is crucial<sup>49</sup>.

- Other important skills include basic security awareness, understanding of networking and client-server interaction<sup>49</sup>.

- Responsibilities can include communicating with stakeholders, contributing to long-term technology roadmaps, developing and maintaining workflows, producing quality code, and recommending system solutions<sup>50</sup>.

This summary encapsulates the core concepts from the provided sources that would likely be central to our discussions about backend development.