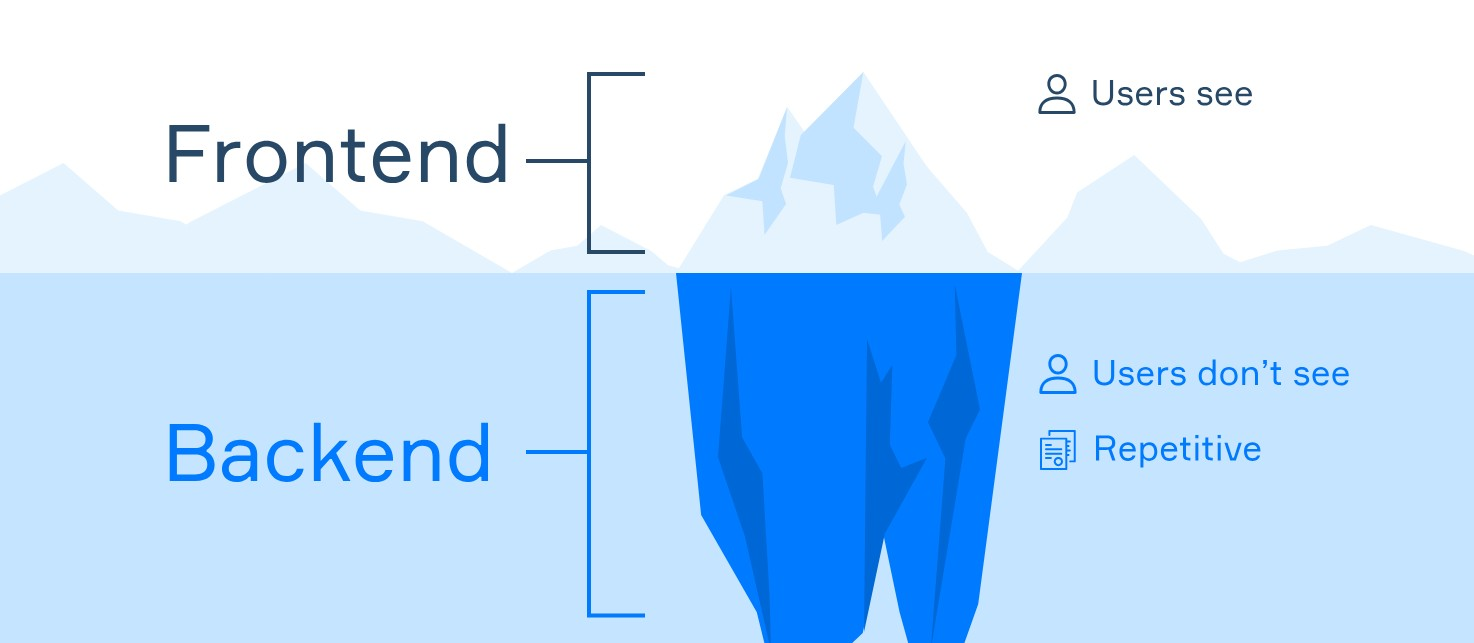
Web development is the process of creating web applications and websites that power the digital world. It enables businesses, communities, and individuals to share information, conduct transactions, and interact online. Developers often separate a web application's logic into server-side and client-side components, each requiring specialized skills. Understanding both frontend and backend development is essential for building robust, user-friendly, and scalable web applications. Let's delve deeper into these components and explore their roles in the web development ecosystem.

**Frontend and backend development**

It is customary to divide web development into two parts: frontend and backend.

**Frontend development,** also known as **client-side** programming, involves creating the user interface and writing the interaction logic with users. In essence, frontend development handles everything users see and interact with on a web page.

**Backend development,** also called **server-side** programming, focuses on developing the internal server side of the web application and implementing its logic. Backend development includes everything users do not see, such as server management, database interactions, and application logic.



**Developers**

In web development, specialists focus on different aspects of creating web applications. These specialists include frontend developers, backend developers, and full-stack developers, each playing a key role in bringing a project to life.

* **Frontend developers** are responsible for interpreting designs into functional web pages. They ensure websites look great and function correctly across various browsers and devices. Typical tasks for frontend developers include creating responsive layouts, building interactive elements like sliders and forms, and optimizing user interfaces for performance and accessibility.
* **Backend developers** focus on the server-side logic and infrastructure of web applications. They write core business logic, process user data, manage databases, and ensure data security and application scalability. Tasks handled by backend developers include developing APIs, integrating third-party services, and implementing authentication and authorization systems.
* **Full-stack developers** handle both frontend and backend development. They work on all phases of web application development, from creating the user interface to implementing server-side functionality. Full-stack developers are often involved in building complete applications, managing end-to-end development processes, and integrating various technologies.

**Separation of client and server parts**

In web development, "client-side" and "server-side" describe where the application code is executed. But why do we separate the client and server components? Why do we need the user interface representation and the server logic separated from each other? There are several reasons for this:

* You can independently update the frontend and backend logic, thus reducing errors. For instance, you can alter the site's appearance without modifying the common processes.
* You need a server to store and structure data.
* Data on the client's side may become outdated, but the server's responses are usually up to date.
* It's easier to manage data flows and orchestrate processes on a few servers than on millions of clients.
* Some tasks run faster locally without needing a server update.

**How do frontend and backend communicate**

To ensure a web service functions properly, the client and server parts must interact. This interaction occurs through HTTP requests. At its core, the communication between frontend and backend happens as follows:

1. The client sends a request to the server to retrieve or modify data using the HTTP protocol.
2. The backend processes this request and returns a response via HTTP.
3. The frontend receives the response, processes it, and displays the result to the user.

JSON stands as the most popular format for exchanging data between client and server, though other formats exist. Markup languages, like HTML and XML, help display server responses in a user-friendly manner.

**Conclusion**

In this topic, you've explored the organization of the web development process, the reasons for dividing it into server and client parts, and how these parts interact with each other. Typically, backend and frontend developers handle the server and client parts, respectively. However, full-stack developers work with both frontend and backend development.

 **Frontend developer** → *Make the site display correctly on devices with different screen sizes*

 **Frontend developer** → *Write the logic of interaction with users*

 **Backend developer** → *Write the basic business logic of the application*

 **Backend developer** → *Organize information storage in databases*

What is the most popular format for exchanging data between a client and a server?

Hint Earlier, XML was the dominant format for data exchange, but considering readability and agility, it was replaced by a Javascript based notation

 **JSON (JavaScript Object Notation)** is the most popular data exchange format today due to its **readability, simplicity, and compatibility** with JavaScript and many other languages.

 **XML** was widely used before but is more verbose and complex.

 **HTML** is used for displaying content, not exchanging data.

 **HTTP** is a protocol, not a data format.

Client sends a request to the server to retrieve or modify data using the HTTP protocol.

Backend processes this request and returns the response via HTTP.

Frontend receives the response and processes it, and then displays the result to the user.

These are markup languages:

HTML

XML

Front end backend interact through http request