



Introduction to NodeJS & MongoDB

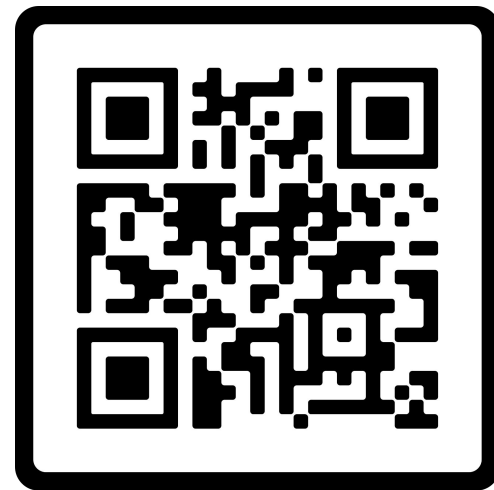
A Student-Run Short Course (SRC) conducted by the Student Academic Council.

Instructors:

- Reuben Devanesan
- Ananthu JP
- Kanishk Singhal

Logistics

- ❑ 3rd to 12th Apr, 10 PM to 11 PM. AB 7/209.
- ❑ All important updates will be communicated through Google Classroom.
- ❑ Earlier Parts :
 - ❑ <https://github.com/Reuben27/ReactJS-SRC>
 - ❑ <https://github.com/Reuben27/Web-Development-SRC>
- ❑ P/F course. Attendance compulsory.



Frontend and Backend Development



Focuses on User Interface and Experience.	Focuses on building logic and functionality behind the scenes.
Involves technologies like HTML, CSS, and JS.	Involves technologies like Node.js, Python, Ruby on Rails, etc.
Responsible for the client-side of applications.	Responsible for the server-side of applications.
It ensures that the application is visually appealing, interactive, and easy to use.	Handle tasks such as data storage, authentication, and server communication.
Tools and frameworks: React.js, Angular, Vue.js, etc	Tools and frameworks: Express.js, Django, Flask, etc.



NodeJS and ExpressJS

- Node.js is an open-source, server-side JavaScript runtime environment.
- It allows developers to run JavaScript code outside of a web browser, on the server side.
- Express.js is a minimalistic web application framework for Node.js. It simplifies the process of building web applications and APIs.
- Provides robust routing, middleware, and templating features with Highly extensible range of plugins available.



NodeJS vs Browsers

- Both the browser and Node.js use JavaScript as their programming language. Building apps that run in the browser is a completely different thing than building a Node.js application.
- **What changes is the ecosystem.** In the browser, most of the time what you are doing is interacting with the DOM, or other Web Platform APIs like Cookies. Those do not exist in Node.js, of course. You don't have the document, window and all the other objects that are provided by the browser.
- And in the browser, we don't have all the nice APIs that Node.js provides through its modules, like the filesystem access functionality.



NodeJS vs Browsers

- Another big difference is that in Node.js you control the environment. Compared to the browser environment, where you don't get the luxury to choose what browser your visitors will use, this is very convenient.
- Another difference is that Node.js supports both the CommonJS and ES module systems (since Node.js v12), while in the browser we are starting to see the ES Modules standard being implemented.
- In practice, this means that you can use both `require()` and `import` in Node.js, while you are limited to `import` in the browser.



API (Application Programming Interface)

- It's a set of protocols that enable different software components to communicate and transfer data. Developers use APIs to bridge the gaps between small, discrete chunks of code to create applications that are powerful, resilient, secure, and able to meet user needs.
- **In layman's terms, API is a messenger.**
 - Say, you visit a restaurant, the waiter presents you with the menu card. You decide what you want to eat. The waiter takes down your order and then shares it with the chef. Once your food is ready, he delivers the meal right to you.
 - This is a very simple process. You do not have to worry about how the food will be prepared or any other thing that takes place inside the kitchen for that matter.
 - Here, the waiter acts as an API. He is the link that establishes connectivity between you and the kitchen.



REST APIs

- Roy Fielding defined **REST (REpresentational State Transfer)** in 2000 as an architectural style and methodology commonly used in the development of internet services, such as distributed hypermedia systems.
- When a request is made via a REST API, it sends a representation of the resource's current state to the requester or endpoint.
- This state representation can take the form of JSON (JavaScript Object Notation), XML, or HTML. RESTful APIs allow you to perform CRUD operations using the POST, GET, PUT, and DELETE HTTP methods
- Other API protocols are SOAP (Simple Object Access Protocol), XML-RPC (XML-Remote Procedure Call), JSON-RPC, etc.

RESTful API Methods



GET

Retrieve a
resource



POST

Create a
resource



PUT

Replace a
resource



PATCH

Update a
resource



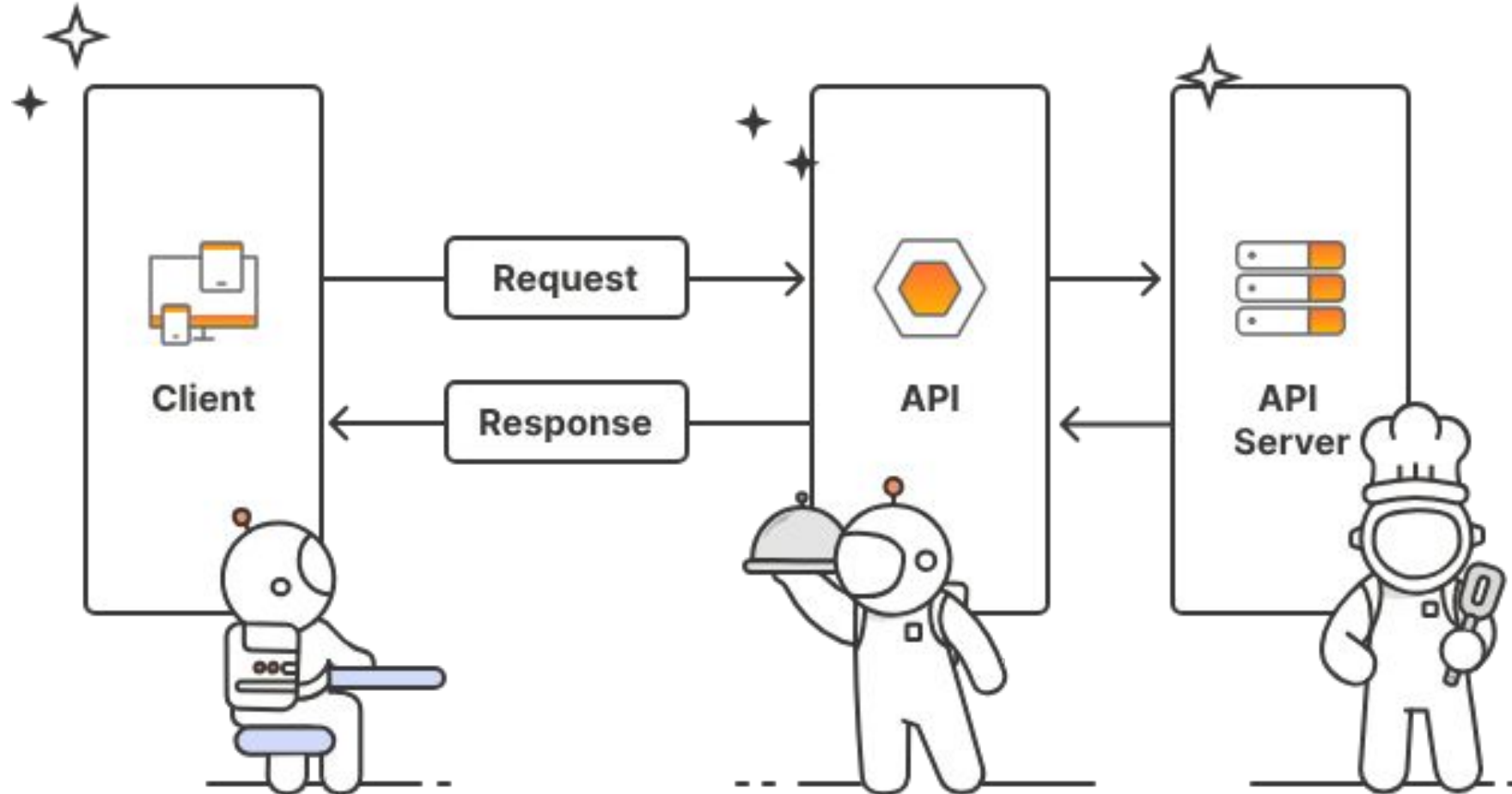
DELETE

Delete a
resource



How do APIs work?

- APIs work by sharing data between applications, systems, and devices. This happens through a request and response cycle. The request is sent to the API, which retrieves the data and returns it to the user.
- **API Client**
 - The API client is responsible for starting the conversation by sending the request to the API server.
 - The request can be triggered in many ways.
 - For instance, a user might initiate an API request by entering a search term or clicking a button.
 - API requests may also be triggered by external events, such as a notification from another application.





How do APIs work?

- API Request

An API request will look and behave differently depending on the type of API, but it will typically include the following components:

- **Endpoint:** An API endpoint is a dedicated URL that provides access to a specific resource. For instance, the /articles endpoint in a blogging app would include the logic for processing all requests that are related to articles.
- **Method:** The request's method indicates the type of operation the client would like to perform on a given resource. REST APIs are accessible through standard HTTP methods, which perform common actions like retrieving, creating, updating, and deleting data.



How do APIs work?

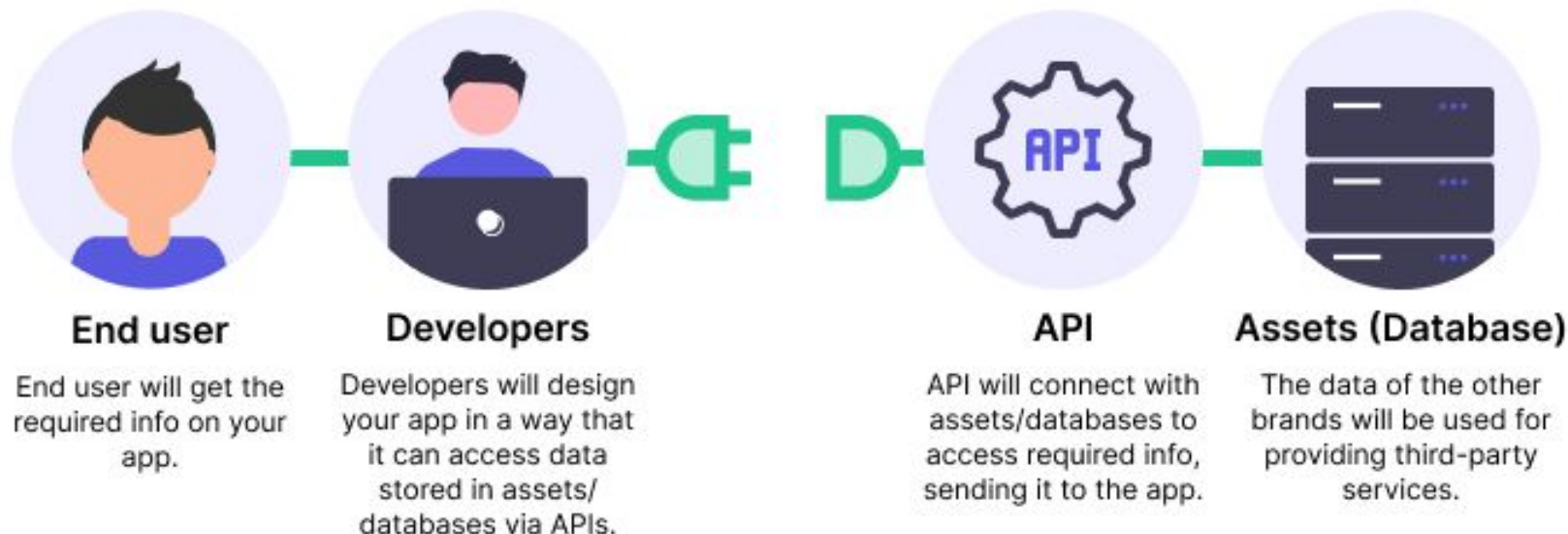
- **API Request**
 - **Parameters:** Parameters are the variables that are passed to an API endpoint to provide specific instructions for the API to process. These parameters can be included in the API request as part of the URL, in the query string, or in the request body. For example, the /articles endpoint of a blogging API might accept a “topic” parameter, which it would use to access and return articles on a specific topic.
 - **Request headers:** Request headers are key-value pairs that provide extra details about the request, such as its content type or authentication credentials.
 - **Request body:** The body is the main part of the request, and it includes the actual data that is required to create, update, or delete a resource. For instance, if you were creating a new article in a blogging app, the request body would likely include the article's content, title, and author.



How do APIs work?

- **API Server**
 - The API client sends the request to the API server, which is responsible for handling authentication, validating input data, and retrieving or manipulating data.
- **API Response**
 - **Status code:** HTTP status codes are three-digit codes that indicate the outcome of an API request. Some of the most common status codes include 200 OK, 201 Created, and 404 Not Found.
 - **Response headers:** HTTP response headers are very similar to request headers, except they are used to provide additional information about the server's response.
 - **Response body:** The response body includes the actual data or content the client asked for—or an error message if something went wrong.

How does an API work?





Benefits of API

- APIs connect various software systems, applications, and devices by allowing them to communicate with one another.
- This unlocks many benefits, ranging from enhanced user experiences to increased business efficiency. The most common advantages of APIs include:
 - **Automation**
 - **Cost Efficiency**
 - **Innovation:** Public APIs used by external teams spurs innovation
 - **Security:** APIs separate the requesting application from the infrastructure of the responding service, and offer layers of security between the two as they communicate.

Types of APIs



Private APIs

- ✦ Used to connect different software components within a single organization
- ✦ Not available for third-party use
- ✦ Some applications may include dozens or even hundreds of private APIs



Public APIs

- ✦ Provide public access to an organization's data, functionality, or services
- ✦ Can be integrated into third-party applications
- ✦ Some public APIs are available for free, while others are offered as billable products



Partner APIs

- ✦ Enable two or more companies to share data or functionality in order to collaborate
- ✦ Not available to the general public
- ✦ Leverage authentication mechanisms to restrict access



Postman

- Postman is an application that allows the testing of web APIs.
- The software was created in 2012 by Abhinav Asthana, Ankit Sobti and Abhijit Kane in Bangalore, India in order to solve the API tests sharing problem.
- Postman provides tools for viewing or visualizing API response data and for managing cookies.
- [30 Days of Postman for Developers](#)
- Postman Collections are a group of saved requests.



Thank you!