

Creating ReST API using express

By: Saad Rahman

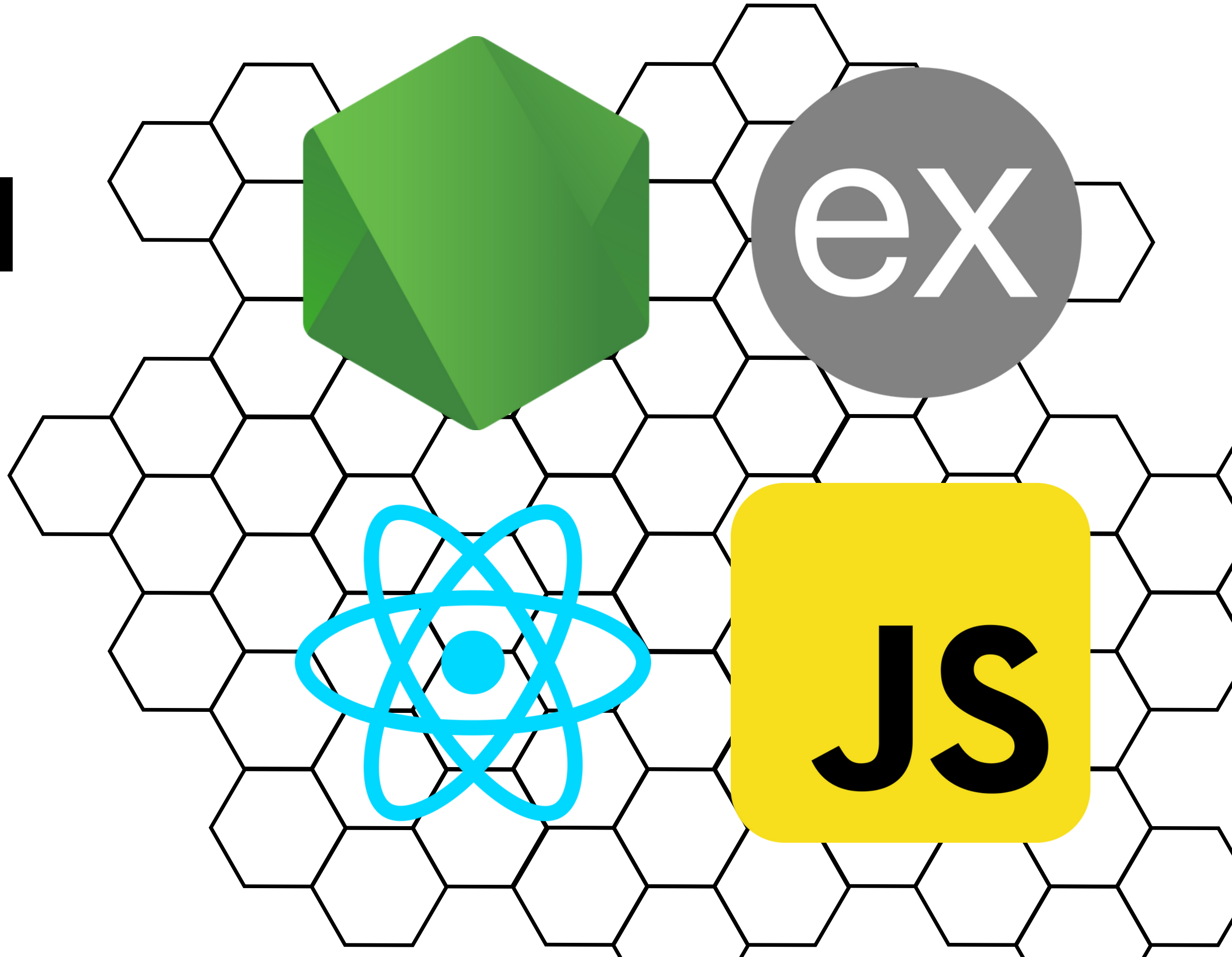
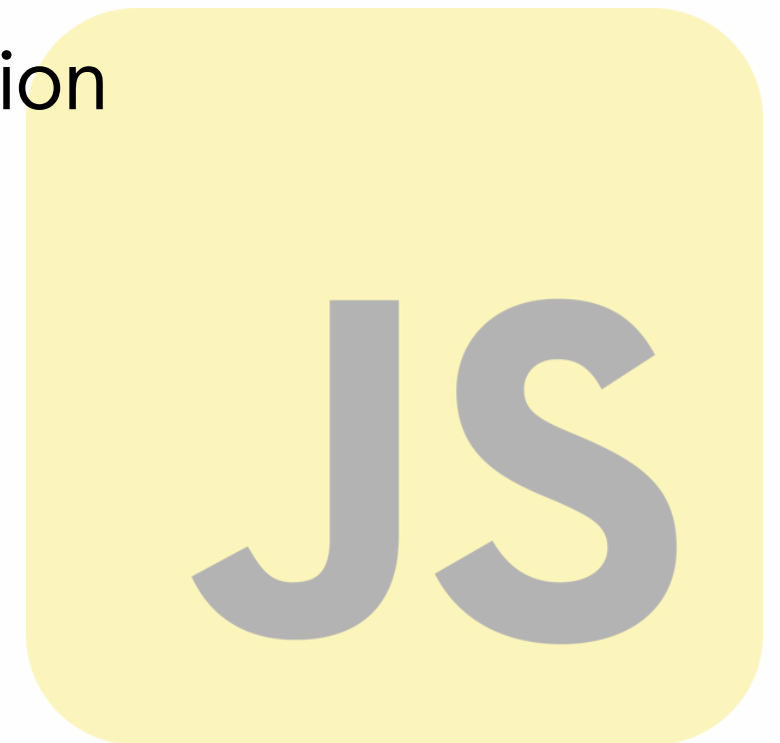
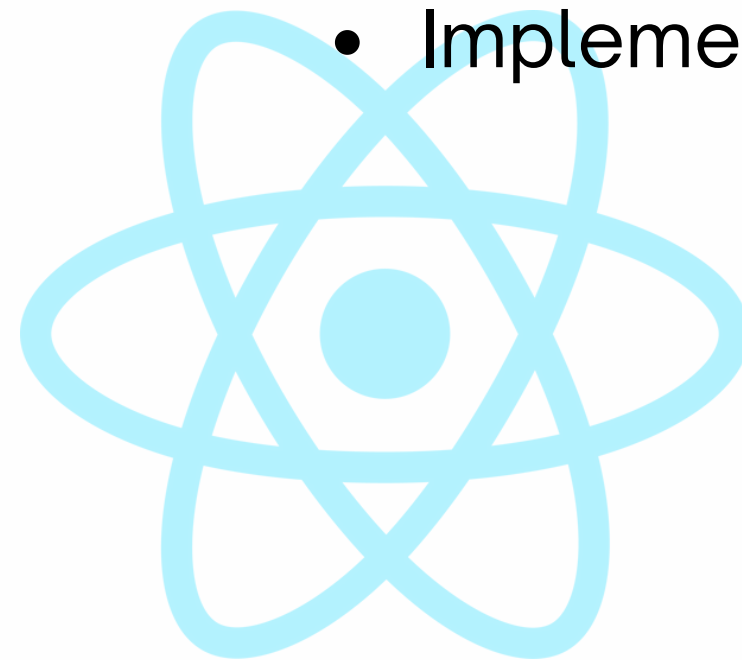
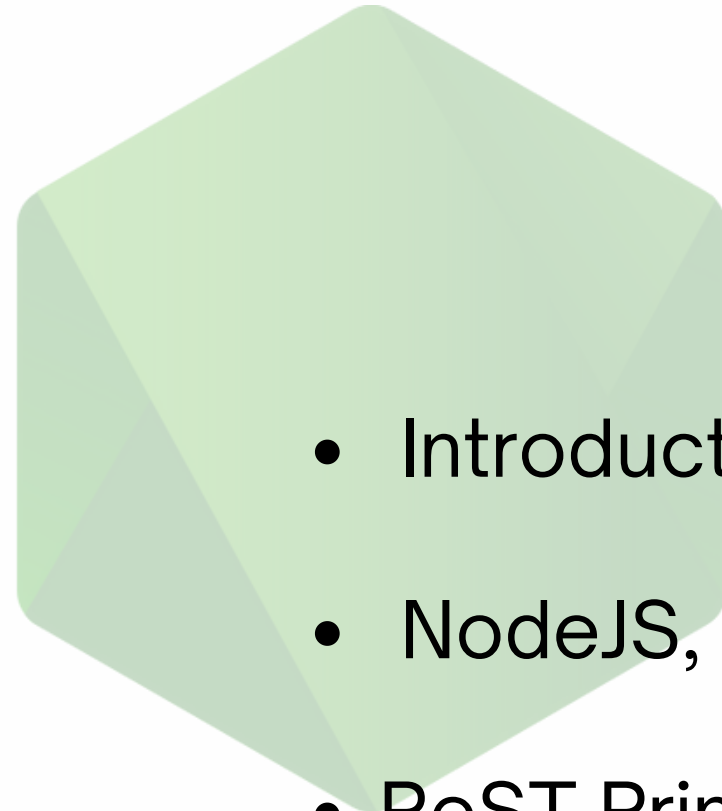


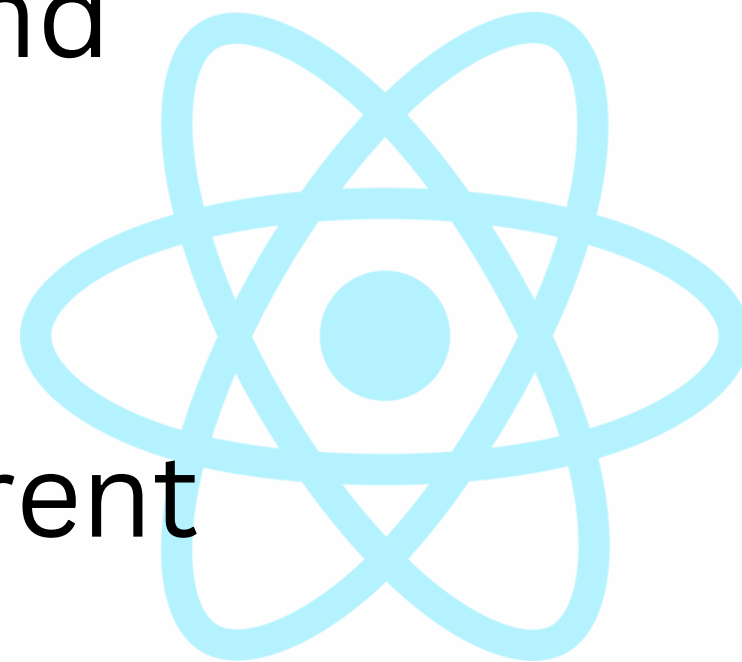
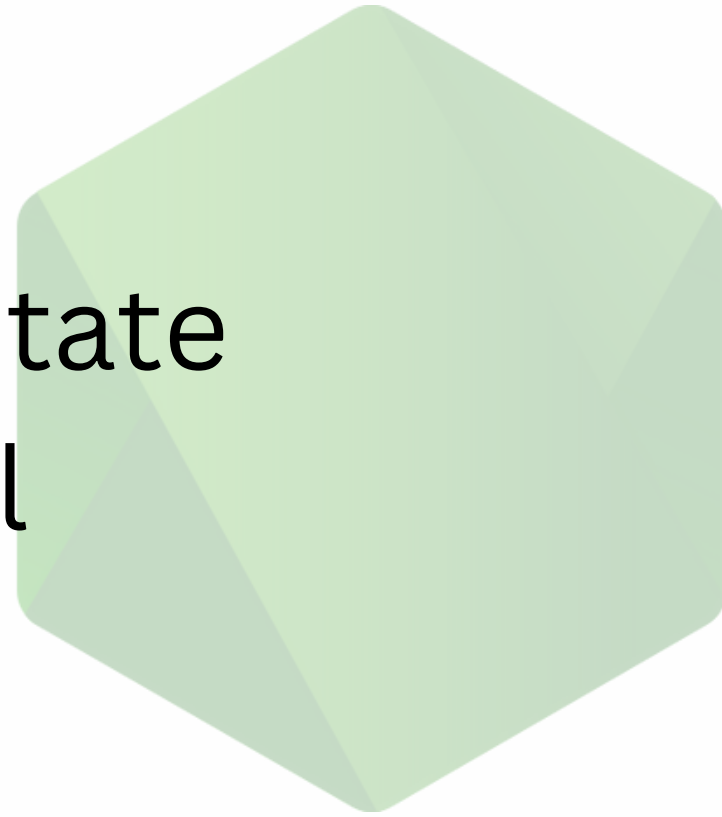
Table of Content

- Introduction
- NodeJS, ExpressJS, ReactJS, JS
- ReST Principles
- Implementation



Introduction

A REST API (Representational State Transfer API) is an architectural style for designing networked applications that utilize standardized HTTP methods and data formats to provide interoperability and communication between different systems on the web.

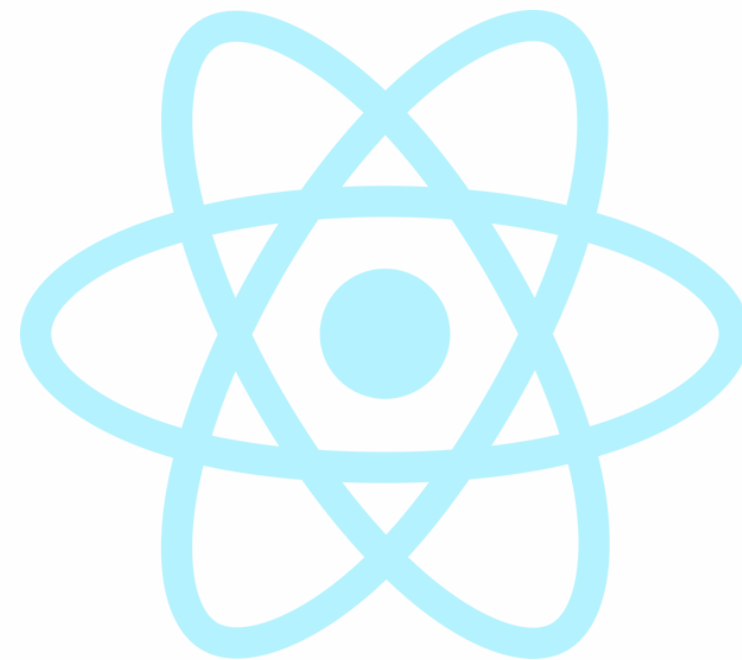


- **Introduction**
- NodeJS, ExpressJS, ReactJS, JS
- ReST Principles
- Implementation



NodeJS

Node.js is a runtime environment that allows developers to run JavaScript code on the server-side

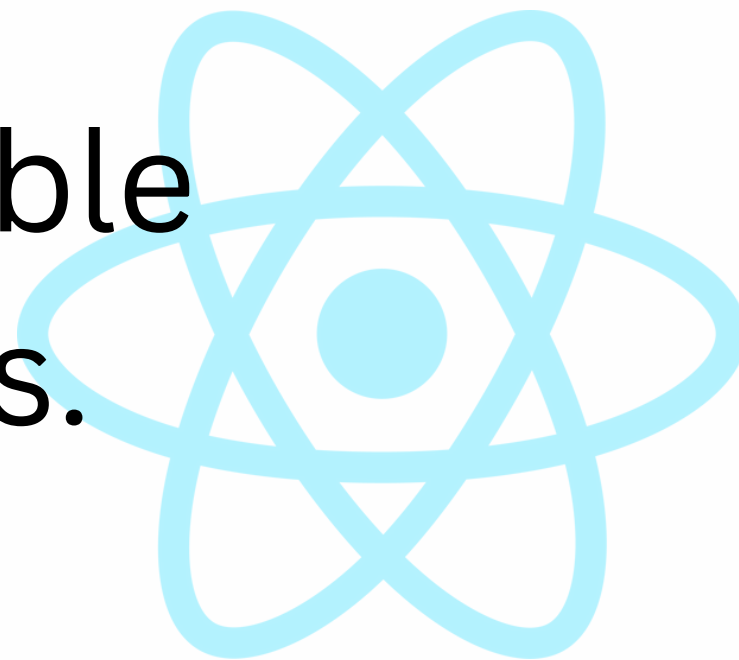
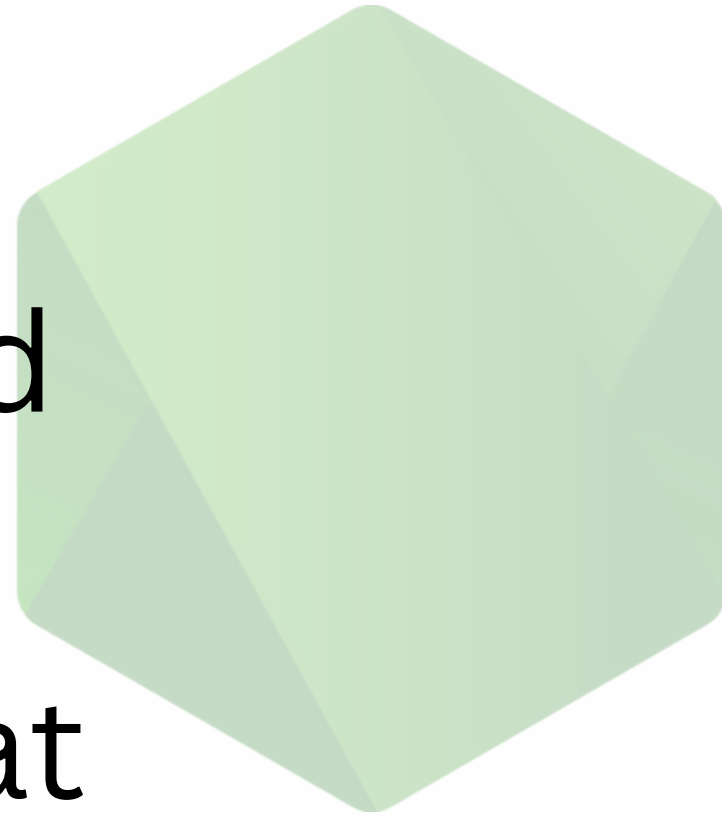


- Introduction
- **NodeJS**, ExpressJS, ReactJS, JS
- ReST Principles
- Implementation

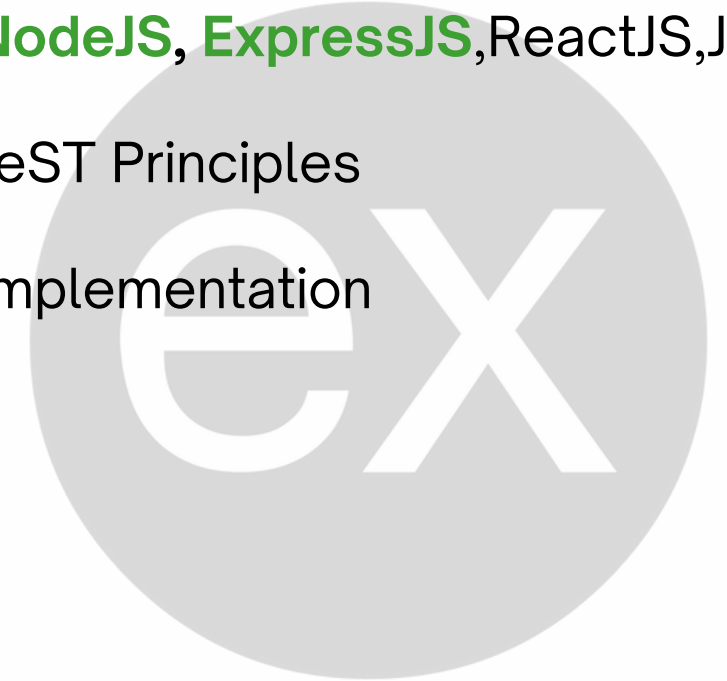


ExpressJS

Express.js is a minimal and flexible web application framework for Node.js that simplifies the process of building robust and scalable web applications and APIs.



- Introduction
- **NodeJS**, **ExpressJS**, ReactJS, JS
- ReST Principles
- Implementation

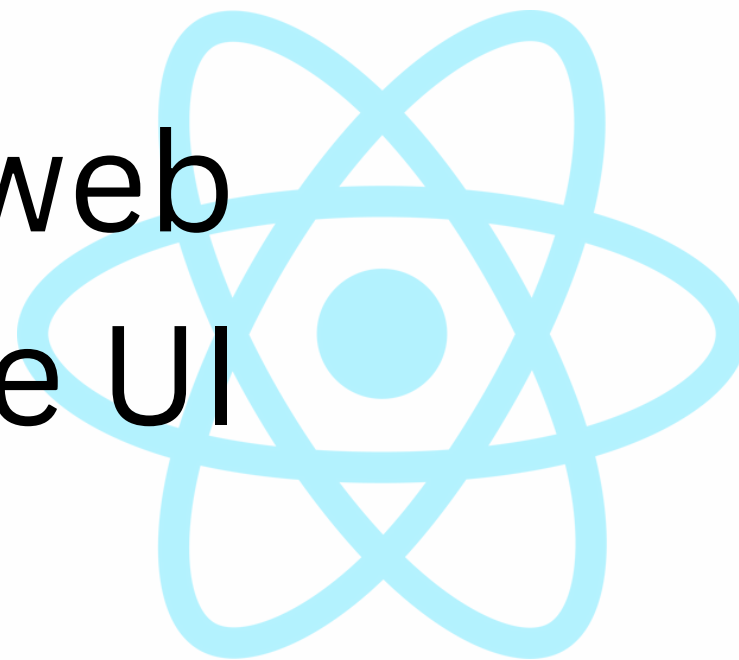
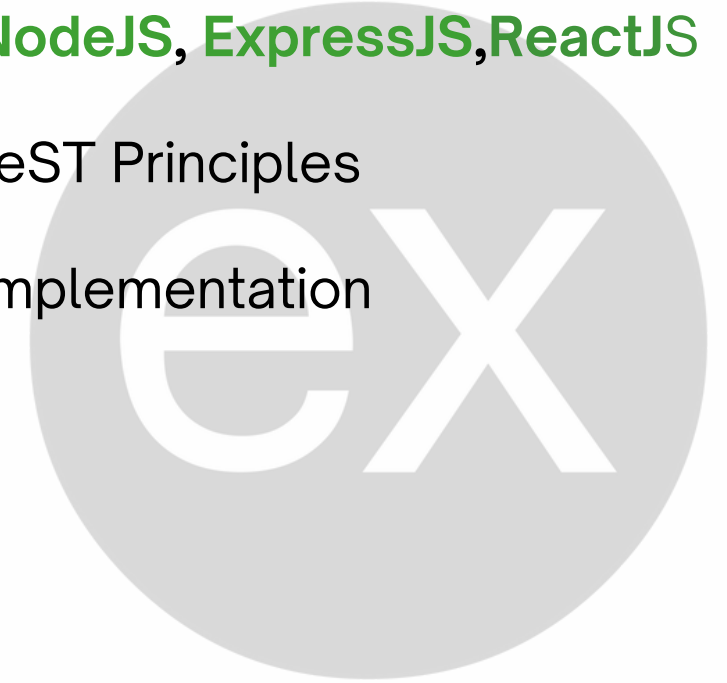


ReactJS

React.js is a JavaScript library for building user interfaces that allows developers to create interactive and dynamic web applications with reusable UI components.



- Introduction
- **NodeJS, ExpressJS, ReactJS**
- ReST Principles
- Implementation



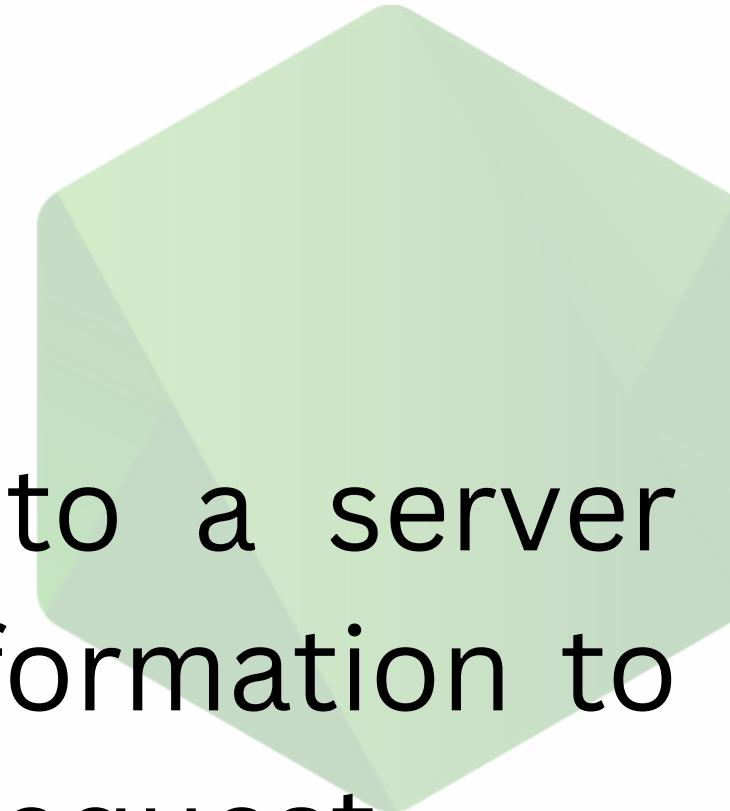
ReST Principles

- **Stateless:**

Each request from a client to a server contains all the necessary information to understand and process that request.

The server does not store any client state between requests.

- Introduction
- NodeJS, ExpressJS, ReactJS
- **ReST Principles**
- Implementation

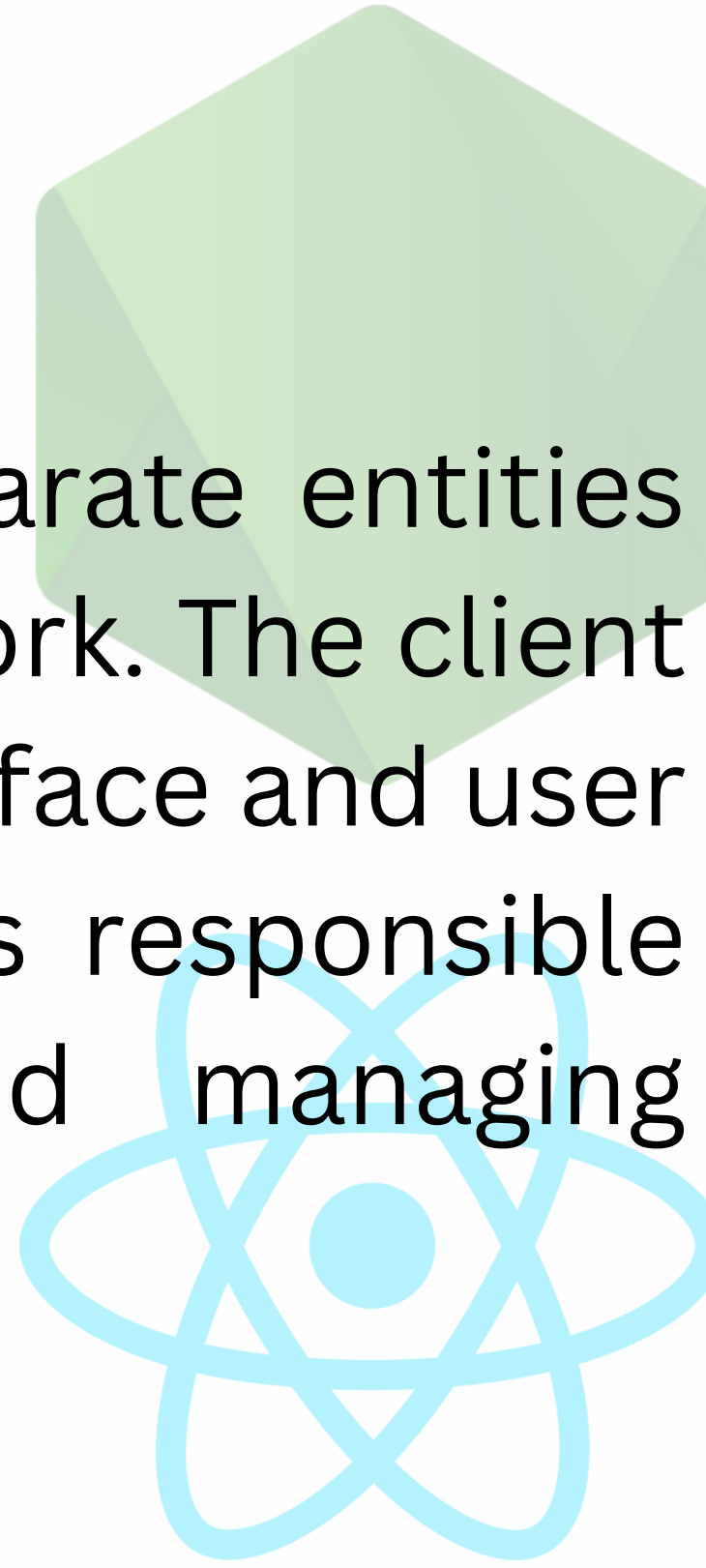


ReST Principles

- **Client-server:**

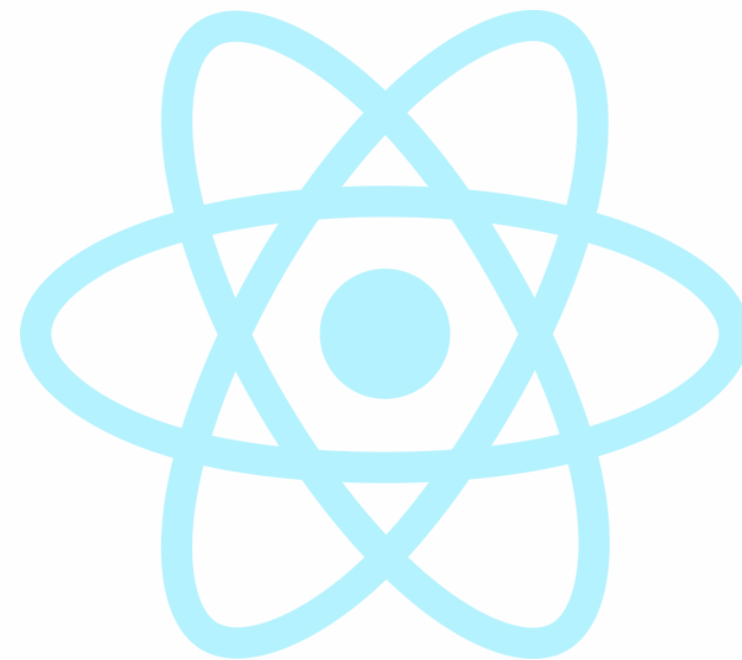
The client and server are separate entities that communicate over a network. The client is responsible for the user interface and user experience, while the server is responsible for processing requests and managing resources.

- Introduction
- NodeJS, ExpressJS, ReactJS
- **ReST Principles**
- Implementation



In Distributed Computing

- Service Abstraction
- Interoperability
- Remote Procedure Calls (RPC's)
- Message Passing





Implementation

