**ReST API – Representational State Transfer**

It is a conceptual way of building a app.js basically which is industry acceptable.

1. We have our client webpage, and we make the client communicate to the Cloud Server via ReST API.
2. It is all about communication between client and server. Any service which uses ReST API, is called a ReSTful API.
3. They are scalable and stateless. As the service grows, we can update the same API, and the data current state does not matter.
4. They support cacheing, so the performance is also good.

**EG:**

1. <http://icecream.com/api/flavour>, here flavour is the resource we might want to work with in an icecream center website.
2. ReST api, is usually used to do CRUD operations with HTTP operations.
   1. Create – POST
   2. Read – GET
   3. Update/Replace – Put or Patch
   4. Delete – DELETE
3. What does a client request consist of? -> Hader, operation, endpoint, parameters.
4. What does a server response consist of? -> JSON object.
5. API GATEWAY – client request traffic controller. Adv – performance, security, protocol translation to a standard, monitoring.



**WDS Simplified – Build an API.**

Nodejs, express and mongodb.

STEPS:

1. – initialise and install all dependencies.
   1. [Use of –-save-dev](https://stackoverflow.com/questions/22891211/what-is-the-difference-between-save-and-save-dev#:~:text=By%20default%2C%20NPM,be%20referred%20here.)
   2. We .gitignore files which are large in size and can be resourced from some other same source, and the sensitive information.
   3. Install dotenv and rest client for vscode if not already.
2. Create a server.js with basic boilerplate
3. Create a mongoose model in a separate folder
4. Create the requests
5. Create middleware