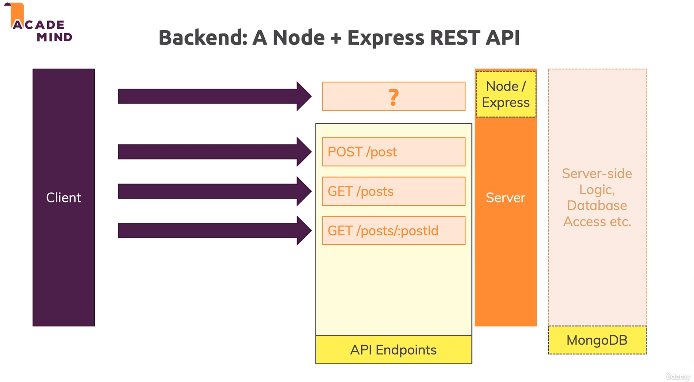
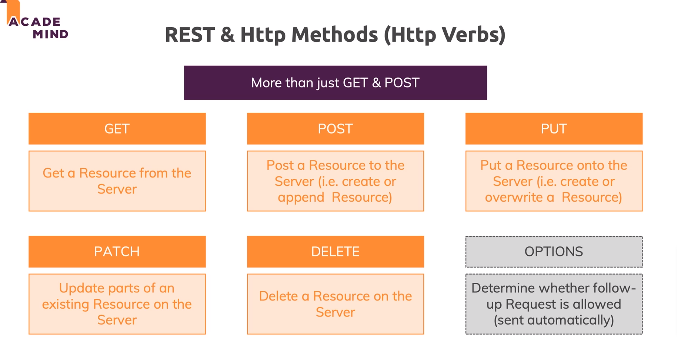
**REST Vs GraphQL API**

**REST API:-**

****

We’ve Client Server combination and the server uses Node Express and then we talk to a MongoDB database from that Node Express Server, the client of course uses React.

Now how does such a request look like which we send to the Node Express server if that server is built as a REST API?



**GET:** Get a Data/Resource from the server.

**POST:** Post a Data/Resource to the server. So to send data there and then create new data, new object, new resources on the server.

**PUT**: PUT which we typically use to create or override a data/resource.

**PATCH**: PATCH which we use to update the data/resource.

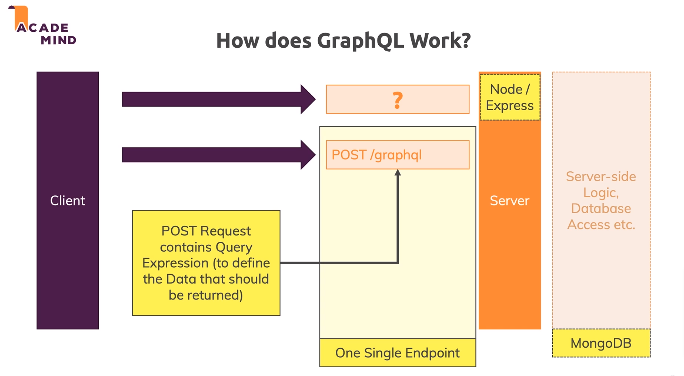
**DELETE**: To delete a data/resource.

OPTIONS: Which we won’t send on ourselves but which the browser automatically attaches for certain requests which browser automatically attaches for certain requests which the browser then uses to find out if the request we actually want to send is supported, I’ll come back to that later.

**Note:**

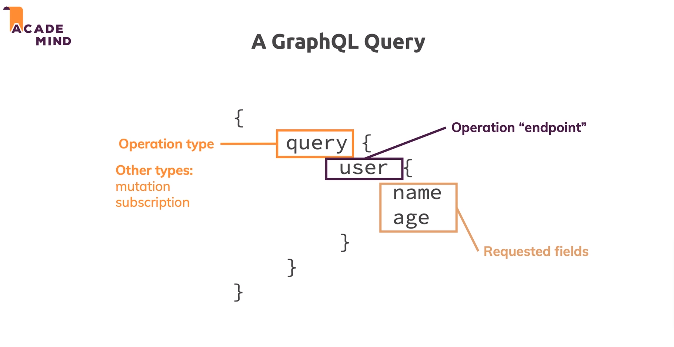
Just because you send a POST request, you don’t actually store something in that database automatically. As I mentioned, it’s up to you, the developer of the API, to decide what you want to do, which code you want to execute, for which verb path combination and you typically want to map the verbs in a way that’s logical, so you wanna create data when a POST request is received, you wanna delete data when a DELETE request is received but theoretically, you could also return data, so you could get data upon a DELETE request or return data upon a PUT request or return data upon a POST request, that would just not be logical and therefore you want to avoid it but it’s really important to understand that the verb path combination alone does not dictate what happens on the server, that is up to the logic you wrote there for the different verb path combinations.

**GraphQL API:-**

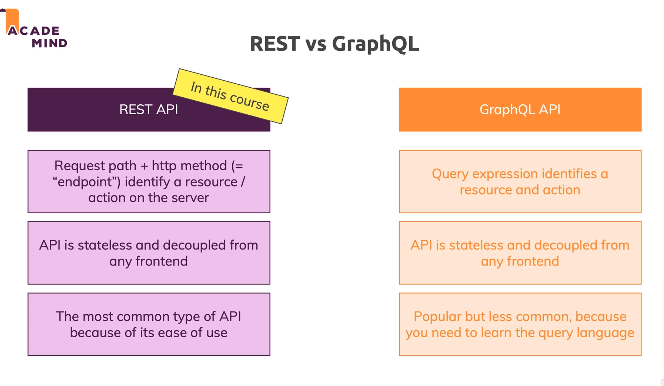
****

In GraphQL, there aren’t multiple end point. There is one single end point i.e. typically a POST request on some path of your choice. Often the path is /graphql but that not must use. Now the trick here is, this request since this is POST request has request body. And that body contains Query Expression(written in standardized graphql query language) which is then parsed by your Node Express server. And then you wrote code to support that specific expression which was send, you can then well use the data which is part of that expression on to store data or to get data and so on.

A typically **Graphql Query** look like this:-



**REST API VS GraphQL API:-**



API is stateless and decoupled from any frontend means?

An API you built cannot just be used with your react single page application. Any client could talk to it. So if you later built a mobile application with iOS or Android , you could talk to the same API because the API is totally seprated from the front end which is the cool thing about this API approach, you can reuse the API and just attach different front end. That’s one of the reason why such API are so popular these days.