**API GATEWAY**

* Example – build a serverless API



* CRUD: Creating, Reading, Updating, Deleting
* So, in this example, we have server less technologies so we're using Lambda, and we're Reading, Creating, Updating and Deleting data from DynamoDB, which are both server less technologies.
* We want external clients to be able to access our Lambda function. But a Lambda function is not exposed as an API right away. For this we need to expose it through an **API Gateway** which is going to provide the client with the rest HTTP API to connect directly to your website. And so, as we can see the client will talk to the API Gateway.
* The API gateway will proxy the request to your lambda functions which will execute the transformations on your data. And so, API Gateway is used as a fully managed service that is going to allow developers to easily create, publish, maintain, monitor, and secure APIs in the cloud.
* API Gateway is fully managed services for developers to easily create, publish, maintain, monitor and secure APIs
* This is serverless is scalable
* Supports RESTful APIs and WebSocket APIs
* Support for user authentication, API throttling, API keys, monitoring etc.