# **Build Serverless APIs with Azure Functions**

**Learning objectives**

* Establish a solid understanding of Application Programming Interfaces (API).
* Create an API with Azure Functions v4 programming model for Node.js.
* Connect to Azure Cosmos DB from an Azure Function.
* Discern the benefits of REST and how to implement a RESTful API.
* Distinguish the different types of security for Azure Functions and web-based APIs.
* Deploy an Azure Functions application.

Scenario:  
Tailwind Traders is a global corporation with a massive online retail operation. You also have brick-and-mortar stores in just about every major city in the world. One of your most critical tasks is keeping up with the inventory of products that you carry.

Logistics specialists need to manage the inventory from all over the globe. You never know when the buying mood will strike, so customer demand can dramatically increase and decrease without warning. When that happens, the inventory management tool's usage spikes. Thousands of inventory managers also use the tool to keep up with demand. Your mission is to create an HTTP API that a lightweight web front end can consume. The whole app needs to be fast and global. It also needs to scale up automatically when traffic unexpectedly spikes without costing a fortune when traffic is low.

## What is an API?

* Connects front end and backend
  + In this case
* Extra leyer for security
  + Ex can’t access connection strings via the front end
* API layers make it easier to create authentication layer.
* Generally easier to manage access to the api and by extension database

## Store connection strings as app setting

* In local.settings.json
* Or .env for js
* App settings are specified in one place and referenced throughout the app.
* You can easily change them at any point in the future without having to change the code.
* App settings are never checked in to source control.

## Migration to RESTful APIs

* Action/Resource pattern
  + Ie api/GetProducts
  + Quickly gets cluttered with long api names
* REST
  + Handler func is separated from definition of the HTTP trigger
  + Uses route parameter to collect endpoints