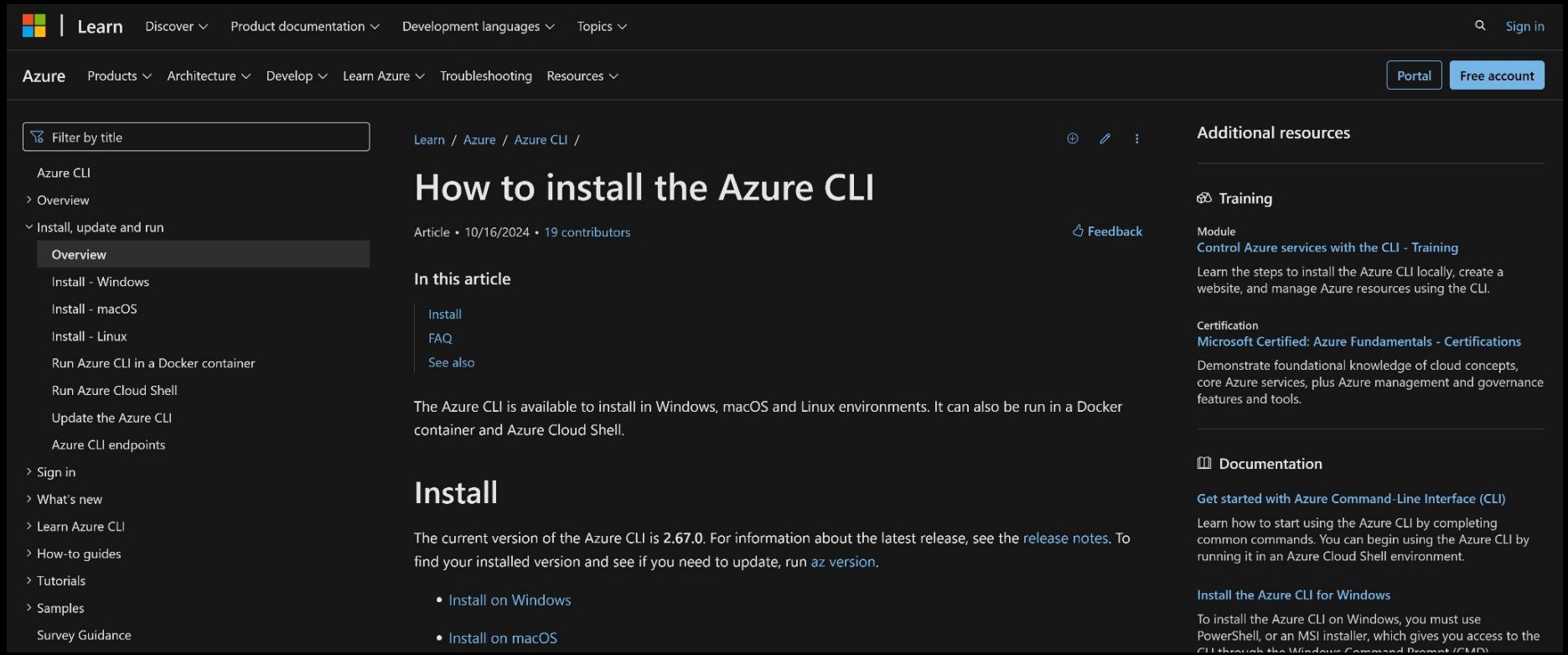
22.8 Using Hosting Provider



22.8 Using Hosting Provider

Install or update

The MSI and ZIP distributable are used for installing or updating the Azure CLI on Windows. You don't need to uninstall current versions before using the MSI installer because the MSI updates any existing version.

> ⓘ Important
>
> After the installation is complete, you will need to **close and reopen any active terminal window to use the Azure CLI**.

Microsoft Installer (MSI) | Microsoft Installer (MSI) with PowerShell | Windows Package Manager | ZIP Package

Latest version

Download and install the latest release of the Azure CLI. When the installer asks if it can make changes to your computer, select the "Yes" box.

Latest MSI of the Azure CLI (32-bit)

Latest MSI of the Azure CLI (64-bit)

If you have previously installed the Azure CLI, running either the 32-bit or 64-bit MSI will overwrite an existing installation.

22.8 Using Hosting Provider

22.8 Using Hosting Provider

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Prashant> az --version
azure-cli                         2.67.0

core                              2.67.0
telemetry                          1.1.0

Dependencies:
msal                              1.31.0
azure-mgmt-resource               23.1.1

Python location 'C:\Program Files\Microsoft SDKs\Azure\CLI2\python.exe'
Extensions directory 'C:\Users\Prashant\.azure\cliextensions'

Python (Windows) 3.12.7 (tags/v3.12.7:0b05ead, Oct  1 2024, 03:06:41) [MSC v.1941 64 bit (AMD64)]

Legal docs and information: aka.ms/AzureCliLegal


Your CLI is up-to-date.
```

22.8 Using Hosting Provider

```
PS C:\Users\Prashant> az login
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No     Subscription name    Subscription ID                       Tenant
-----  -------------------  ------------------------------------  -----------------
[1] *  Free Trial           166737a1-e433-403b-aacb-e39a9d24b588  Default Directory

The default is marked with an *; the default tenant is 'Default Directory' and subscription is 'Free Trial' (166737a1-e433-403b-aacb-e39a9d24b588).

Select a subscription and tenant (Type a number or Enter for no changes): 1

Tenant: Default Directory
Subscription: Free Trial (166737a1-e433-403b-aacb-e39a9d24b588)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.
```

22.8 Using Hosting Provider

Deploy to Azure

Before you continue, ensure that you have all the prerequisites installed and configured.

> **Note**
>
> For your Node.js application to run in Azure, it needs to listen on the port provided by the `PORT` environment variable. In your generated Express app, this environment variable is already used in the startup script *bin/www*. (Search for `process.env.PORT`.)

In the terminal, ensure you're in the *myExpressApp* directory, and deploy the code in your local folder (*myExpressApp*) using the az webapp up command:

Deploy to Linux | Deploy to Windows

Azure CLI — Copy

```
az webapp up --sku F1 --name <app-name>
```

22.8 Using Hosting Provider

```
PS C:\Users\Prashant\Documents\workspaces\MERNLiveProject> az webapp up --sku F1 --name mern-live-backend
The webapp 'mern-live-backend' doesn't exist
Creating Resource group 'omprashantjain_rg_7471' ...
Resource group creation complete
Creating AppServicePlan 'omprashantjain_asp_8475' or Updating if already exists
Readonly attribute name will be ignored in class <class 'azure.mgmt.web.v2023_01_01.models._models_py3.AppServicePlan'>
Creating webapp 'mern-live-backend' ...
Configuring default logging for the app, if not already enabled
Creating zip with contents of dir C:\Users\Prashant\Documents\workspaces\MERNLiveProject ...
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
Polling the status of async deployment. Start Time: 2024-12-01 07:01:15.254002+00:00 UTC
Status: Building the app... Time: 1(s)
Status: Building the app... Time: 18(s)
Status: Building the app... Time: 35(s)
Status: Build successful. Time: 51(s)
Status: Starting the site... Time: 67(s)
Status: Starting the site... Time: 83(s)
Status: Starting the site... Time: 99(s)
Status: Starting the site... Time: 115(s)
Status: Starting the site... Time: 672(s)
Status: Site failed to start. Time: 689(s)
Deployment failed because the site failed to start within 10 mins.
InprogressInstances: 0, SuccessfulInstances: 0, FailedInstances: 1
Error: Deployment for site 'mern-live-backend' with DeploymentId '740108ab-de7d-4235-a240-430f0d16013c' failed because the worker proccess
 start within the allotted time.
```

22.8 Using Hosting Provider

Update appName, .gitignore, and create env files

{} package.json | .gitignore ×

.gitignore

```
1  @output.css
```

{} package.json > {} devDependencies

```
1  {
2    "name": "mern-live-backend",
3    "version": "1.0.0",
4    "main": "app.js",
```

$ .env.development

$ .env.example

$ .env.production

.gitignore

$ .env.production

```
1  MONGO_DB_USERNAME=root
2  MONGO_DB_PASSWORD=root
3  MONGO_DB_DATABASE=airbnb
4
5  FROM_EMAIL=contact@completecoding.in
6  SENDGRID_API_KEY=SG.gcT1EgxbSFWI8Lu_jHo8rQ.JkyF
```

22.8 Using Hosting Provider

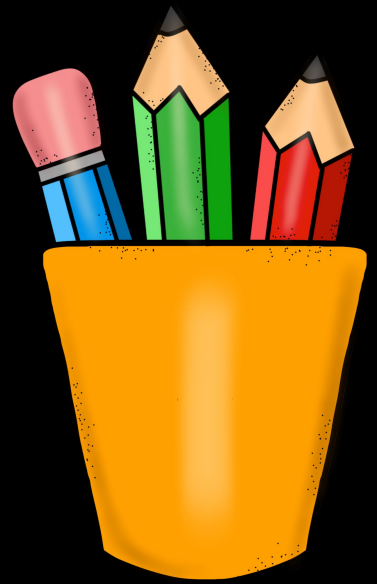
> ⓘ Note
>
> The `az webapp up` command does the following actions:
>
> - Create a default resource group.
> - Create a default App Service plan.
> - Create an app with the specified name.
> - Zip deploy all files from the current working directory, with build automation enabled.
> - Cache the parameters locally in the *.azure/config* file so that you don't need to specify them again when deploying later with `az webapp up` or other `az webapp` commands from the project folder. The cached values are used automatically by default.

22.8 Using Hosting Provider

```
PS C:\Users\Prashant\Documents\workspaces\MERNLiveProject> az webapp up --name mern-live-backend
Webapp 'mern-live-backend' already exists. The command will deploy contents to the existing app.
Creating AppServicePlan 'omprashantjain_asp_8475' or Updating if already exists
Readonly attribute name will be ignored in class <class 'azure.mgmt.web.v2023_01_01.models._models_py3.AppServicePlan'>
Creating zip with contents of dir C:\Users\Prashant\Documents\workspaces\MERNLiveProject ...
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
Status: Building the app... Time: 1(s)
Status: Building the app... Time: 21(s)
Status: Build successful. Time: 37(s)
Status: Starting the site... Time: 53(s)
Status: Starting the site... Time: 69(s)
Status: Starting the site... Time: 86(s)
Status: Site started successfully. Time: 102(s)
You can launch the app at http://mern-live-backend.azurewebsites.net
Setting 'az webapp up' default arguments for current directory. Manage defaults with 'az configure --scope local'
--resource-group/-g default: omprashantjain_rg_7471
--sku default: F1
--plan/-p default: omprashantjain_asp_8475
--location/-l default: canadacentral
--name/-n default: mern-live-backend
{
  "URL": "http://mern-live-backend.azurewebsites.net",
  "appserviceplan": "omprashantjain_asp_8475",
  "location": "canadacentral",
  "name": "mern-live-backend",
  "os": "Linux",
  "resourcegroup": "omprashantjain_rg_7471",
  "runtime_version": "node|16-LTS",
  "runtime_version_detected": "0.0",
  "sku": "FREE",
  "src_path": "C:\\Users\\Prashant\\Documents\\workspaces\\MERNLiveProject"
}
```

Revision

1. What is Deployment?
2. Cloud vs Local Deployments
3. Using Environment Variables
4. Secure Response Headers
5. Use Compressed Assets
6. Request-Response Logging
7. Using SSL/TLS Encryption
8. Using Hosting Provider



Practise Milestone

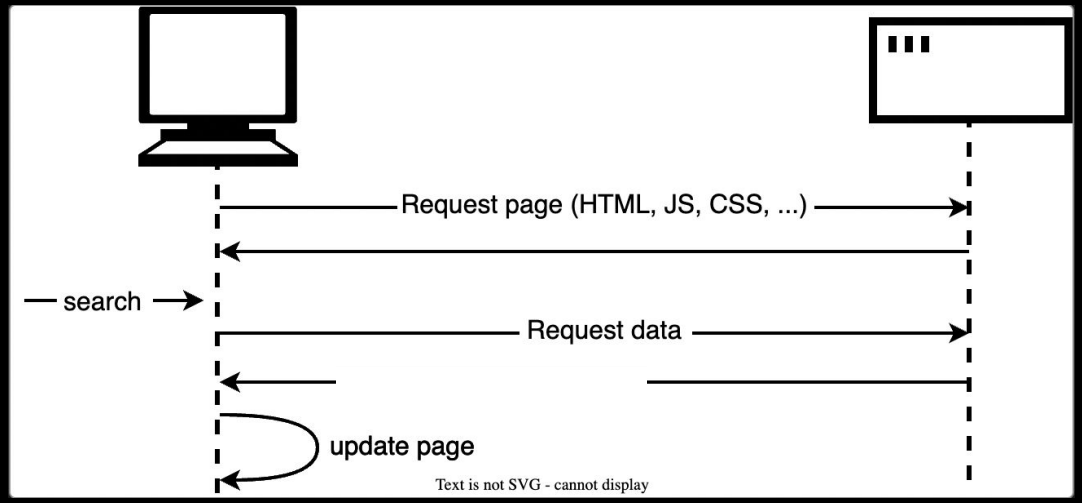
Take your airbnb forward:

1. Check the UI of Azure Cloud Services and familiarize yourself with it.

23. REST API / JSON Requests

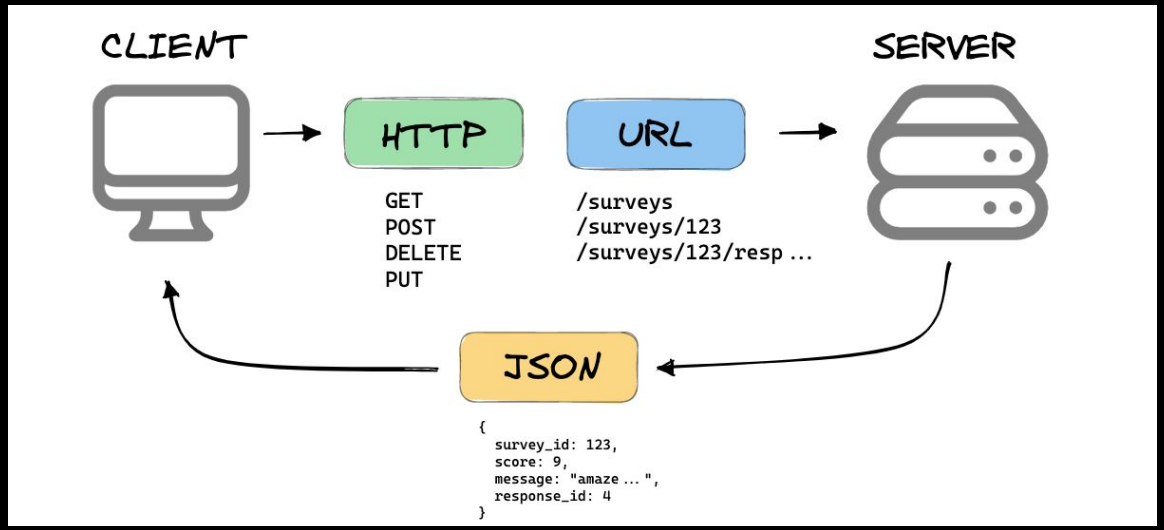
1. What are Async Requests
2. What are REST APIs
3. Decoupling Frontend & Backend
4. Routes & HTTP Methods
5. REST Core Concepts
6. First API Todo App
7. API for Fetch Items
8. API for Deleting Items
9. Adding Complete Item functionality

23.1 What are Async Requests



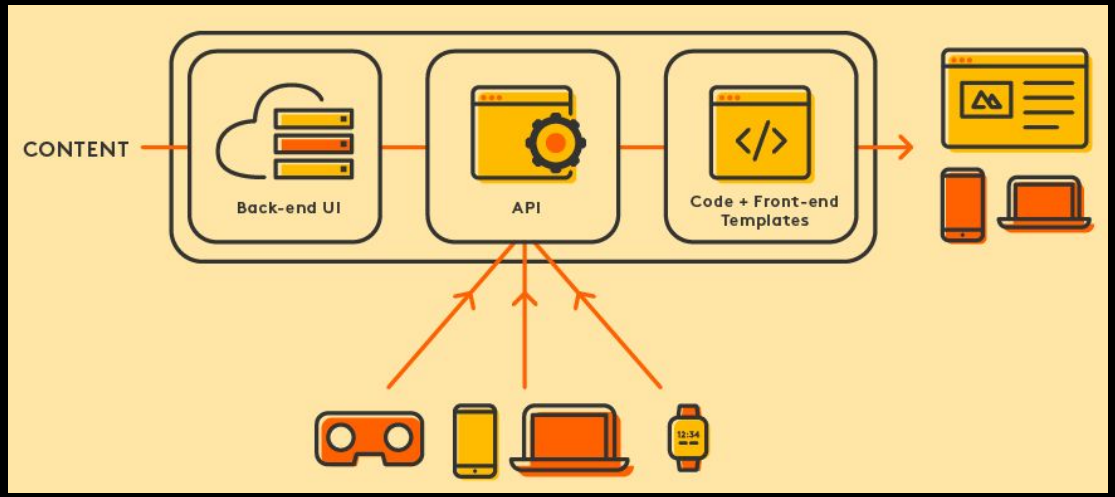
- Async network requests enable web pages to communicate with servers without reloading.
- The client sends JSON requests to the server asynchronously in single-page apps.
- The server processes the request and returns a JSON response.
- The page updates itself dynamically using the received data.

23.2 What are REST APIs



- REST APIs enable communication between clients and servers using HTTP.
- They are mainly identified by a URI.
- They use standard HTTP methods like GET, POST, PUT, and DELETE.
- Data is exchanged in formats like JSON or XML.
- REST APIs are stateless.
- REST APIs allow clients to access and manipulate web resources.

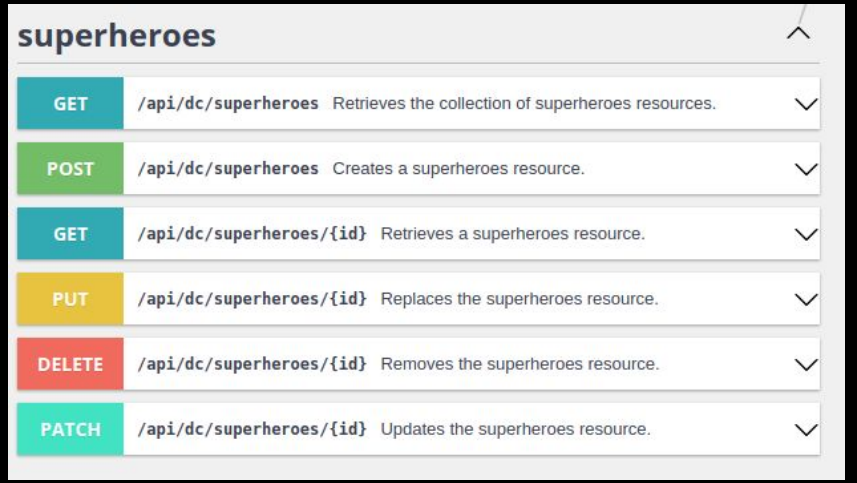
23.3 Decoupling Frontend & Backend



- Separating front-end and back-end allows independent development and scaling.
- REST APIs serve as a communication layer between them.
- Front-end interacts with back-end through standardized RESTful calls.
- Decoupling enhances flexibility and simplifies maintenance.
- REST APIs enable front-end updates without altering back-end code.

23.4 Routes & HTTP Methods

- REST API routes define the endpoints (URLs) where resources can be accessed by clients.
- GET: Retrieves data from the server at the specified route.
- POST: Sends new data to the server to create a resource.
- PUT: Updates or replaces an existing resource at a given route.
- DELETE: Removes a resource from the server at the specified route.
- PATCH: Partially updates an existing resource with new data.



23.5 REST Core Concepts

- Statelessness: Each request contains all necessary information; the server maintains no client session.
- Uniform Interface: Standardized communication using HTTP methods like GET, POST, PUT, DELETE.
- Client-Server Separation: Independent development of front-end and back-end components.
- Cacheability: Responses indicate if they can be cached to improve performance.
- Layered System: Architecture allows for multiple layers between client and server.
- Code on Demand (Optional): Servers can extend client functionality by sending executable code.

23.6 First API Todo App

1. Copy the following from the previous app:
    a. Env files
    b. Package dependencies
    c. .gitignore
    d. app.js
2. Changes in env files
    a. Remove the email data.
    b. Change DB name to todo-app
3. Create empty controllers and routes folders.
4. Remove excess code from app.js.
5. Have the error controller give out a 404 error message and status.
6. Create a Mongoose model for TodoItem.
7. Create itemsRouter and itemsController for POST /todos request from frontend.
8. Install and setup CORS package in backend.
9. Add express.json() to app.