# WEB API

[WEB API 1](#_Toc25250)

[References 1](#_Toc30801)

[Concepts 2](#_Toc10223)

[API Questions 2](#_Toc26437)

[Getting started 2](#_Toc7711)

[Launchsettings.json 2](#_Toc15532)

[Appsettings.json 2](#_Toc29351)

[Secrets.json 3](#_Toc19200)

[Program.cs 3](#_Toc1415)

[Controllers 3](#_Toc13290)

[Http Verbs 3](#_Toc21704)

[FromBody attribute 4](#_Toc23630)

[Testing 4](#_Toc31951)

[API security 4](#_Toc21952)

[Versoning 4](#_Toc15489)

[Monitoring 4](#_Toc4269)

[Protecting 4](#_Toc9036)

[Building 4](#_Toc1209)

[Consuming 4](#_Toc131)

[Open API 4](#_Toc10531)

[Minimal API 4](#_Toc9917)

[Best practices 4](#_Toc9515)

### References

Tim videos: Tim\Web API

Code: DotnetDev\WebAPI\WebAPI

## Concepts

## API Questions

## Getting started

Project creation “ASP net core web API”

### Launchsettings.json

Helps us to launch the app with various profiles.

Profile with app name runs in kestrel(used for internal testing) server

Development or Production settings. Production will be selected by default.

Authentication settings

### Appsettings.json

Configuration settings. Used to avoid hard coding in c#.

Example: Logging information - loglevel, etc

Separate files will be present for production and development mode mentioned in appsettings.json file

Need of this file?

Text file, so we can change the information during production for debug. Connection string based on the stage of application release we can select the appropriate server.

### Secrets.json

Update app settings that are specific to particular machine. This will not be in source code location. So this data will not be committed to repo.

### Program.cs

Dependency injection of service. Scoped(one instance per user of API), transient(new instance every time) or singleton(one instance for entire instance of API) service

WebApplication supports many features(middleware) like UseAuthorization, UseHttpsRedirection, UseSwagger, Run(Actually starts the API service).

### Controllers

Define various endpoints. Various decorator like [ApiController] , Route

### Http Verbs

**Http Verbs: Put, Post, Get, Patch and Delete**

### FromBody attribute

## Testing

## API security

### Secrets

### Tokens

### Authentication and authorization

### Claims

### Advanced Authorization

## Versoning

## Monitoring

## Protecting

## Building

## Consuming

## Open API

## Minimal API

## Best practices

CORS??????