# MVC

## Version

## Filters

* **Authorization** filters – Implements the **IAuthorizationFilter** attribute.
* **Action** filters – Implements the **IActionFilter** attribute.
* **Result** filters – Implements the **IResultFilter** attribute.
* **Exception** filters – Implements the **IExceptionFilter** attribute.

# WEB API

## Application structure of asp net core web api

appsettings.json,appsettings.Development.json,appsettings.Production.json. Development and production specific files overrides appsettings.json property

IConfiguration can be used to access all data from json file

Sample file,

{

"**ConnectionStrings**": {

"DefaultConnection": "Server=localhost;Database=MyDb;User Id=myUser;Password=myPassword;"

},

"**Logging**": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning"

}

},

"**AllowedHosts**": "\*"

}

Sample, accessing **connectionstring** in controller

public class MyController : ControllerBase

{

private readonly **IConfiguration** \_configuration;

public MyController(IConfiguration configuration)

{

\_configuration = configuration;

}

public IActionResult Get()

{

var connectionString = \_configuration.GetConnectionString("DefaultConnection");

// Use the connection string for database operations

return Ok("Connection string retrieved.");

}

}

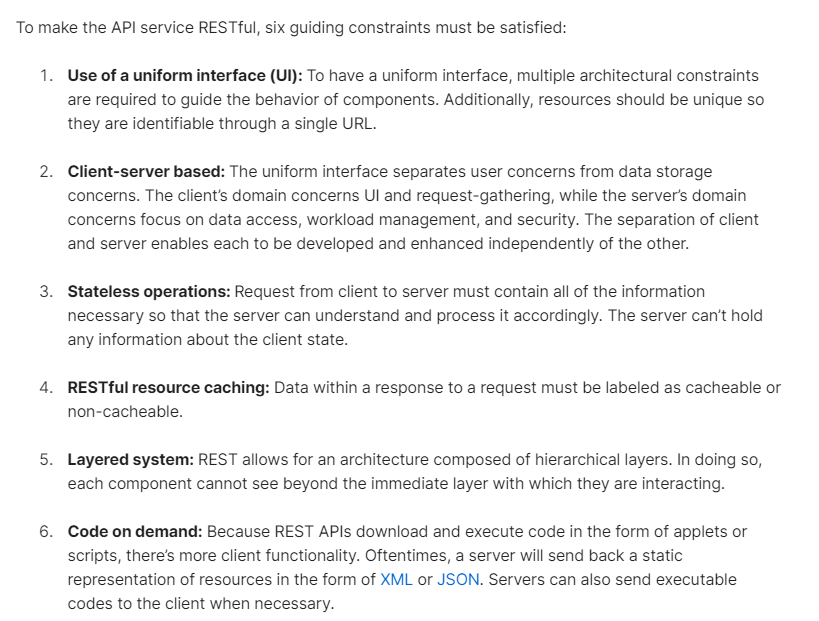
## References

Nick - Dometrain

## Intro to RESTful APIs

**REpresentational State Transfer**

### Six constraints to make API services RESTful



Source: [Link](https://blog.postman.com/rest-api-examples/)

### Resource naming and routing

Usage of plural and singular

-GET /movies

-GET /movies/1

-GET /movies/1/ratings : To find the ratings of movie

-GET /ratings/me : To find ratings given by me

- PUT /movies/1/ratings

- DELETE /movies/1/ratings

URL should not define the action. i.e For GET/PUT and DELETE URL remains same ***“/movies/1/ratings”.*** But the http verbs will define the action.

### Http verbs are meaningful

POST - Create

GET - Retrieve, safe (because it will not change the state of the resource)

PUT - Complete update

PATCH - Partial update

DELETE - Delete

### Using response codes to indicate status

POST

- Single resource (/items/1): NA

- Collection resource (/items): 201 (Location header) (Provides link to the newly added item), 202(accepted)

GET

- Single resource (/items/1): 200(Ok), 404(Not found)

- Collection resource (/items): 200 (always ok irrespective of empty collection or multiple items)

PUT

- Single resource (/items/1): 200, 204(no content),404(not found)

- Collection resource (/items): 405 (method not allowed)

DELETE

- Single resource (/items/1): 200, 404

- Collection resource (/items): 405. This can be done, but not recommended.

**TODO :: Write APIs for each use case**

### Flexible response body options

Can take Xml

Accept: application/xml

Mostly json type will be used

### Understanding idempotency

Post is not idempotent, all other verbs are idempotent.

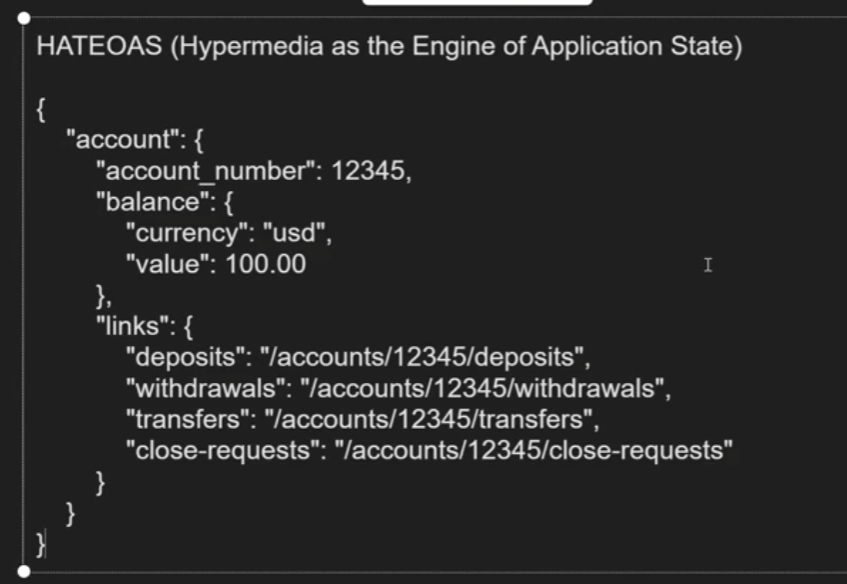
TODO::

What happens if user tries to send same Post request again?

### Using hypermedia for connectedness

HATEOAS - Hypermedia as the Engine of application state

Links to some of the endpoints will be past of response. Not recommended to use.



### Types of errors

Error - Client issue (400)

Fault - Server issue (500)

## Getting started with REST API

Repo link: https://github.com/BalajiBaskaran24/DotnetDev/tree/main/WebAPI

APIs implementation

## Beyond basic CRUD

## Authentication and Authorization

Authentication: Who is the user

Authorize: What user can do

Web APIs are stateless: So token will be passed as part of every request

Services to create or validate tokens: Azure, Aws, Auth0

Standard for token: Json web token

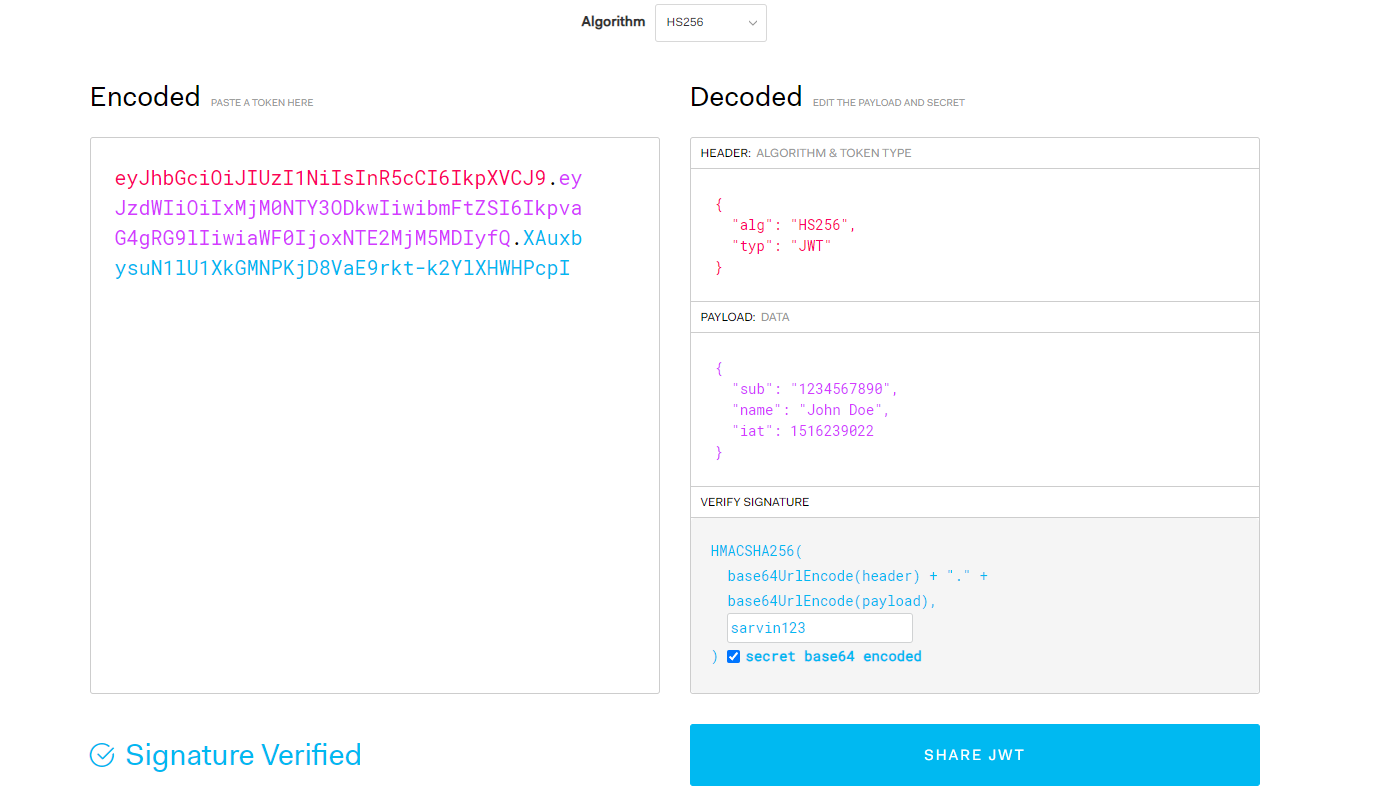
To keep tokens: azure key vault or aws secrets manager

Attributes: [Authorize] [AllowAnonymous]

Postman: Add the token in parent collection so that we don’t need to provide token for every API call

### JWT

Token contains Header, payload and signature.



### Links

<https://jwt.io/>

https://www.postman.com/api-platform/api-authentication/

## User specific action

## Implementing Advanced features

## Building an SDK for our API

## Migrating to Minimal API

## Misc and Questions

### Interview Questions

### Redis Output caching

Reference youtube video: <https://www.youtube.com/watch?v=_bg5dGnudPs>