ASP.Net Web API Tutorial Video Playlist:

https://www.youtube.com/watch?v=\_8nLSsK5NDo&list=PL82C6-O4XrHdiS10BLh23x71ve9mQCln0

<https://www.youtube.com/watch?v=0pcM6teVdKk&list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b>

ASP.Net Web API – HTTP-Based Services on top of ASP.Net Framework

REST Constraints:

1. Request between Client and Server
2. Communication between client and server must be stateless, means we should not store anything on server related to client.
3. Cacheable: let the client know how long the information can be used.
4. Uniform interface: GET, PUT, POST, DELETE
5. Layered System
6. Code on Demand

Each API tells what to do with resource

**REST API Documentation (CRUD)**

Resource shown as **Product** model for convenience.

**Base URL**

https://{host}/api

**Authentication**

None by default. (Add Bearer <token> if you enable auth.)

**Media Types**

* Request: application/json
* Response: application/json; charset=utf-8

**Endpoints**

**List Products**

**GET** /products

Returns a paginated list of products.

**Query parameters**

* page (int, default 1) – 1-based page index
* pageSize (int, default 20, max 100)
* search (string, optional) – case-insensitive match on name
* sort (string, optional) – e.g., name, -price (prefix - for desc)
* minPrice, maxPrice (decimal, optional) – filter range

**Responses**

* 200 OK

{

"items": [

{ "id": 1, "name": "Keyboard", "price": 49.99 }

],

"page": 1,

"pageSize": 20,

"totalItems": 32,

"totalPages": 2

}

**Get Product by Id**

**GET** /products/{id}

**Path parameters**

* id (int) – Product identifier

**Responses**

* 200 OK

{ "id": 1, "name": "Keyboard", "price": 49.99 }

* 404 Not Found

{ "error": "NotFound", "message": "Product 99 was not found." }

**Create Product**

**POST** /products

**Request body**

{

"name": "Headset",

"price": 99.50

}

**Validation**

* name: required, length 1–100
* price: required, >= 0

**Responses**

* 201 Created + Location: /api/products/{id}

{ "id": 3, "name": "Headset", "price": 99.5 }

* 400 Bad Request (validation)

{

"error": "ValidationFailed",

"details": {

"name": ["Name is required."]

}

}

* 409 Conflict (duplicate unique field, if applicable)

**Update Product (full)**

**PUT** /products/{id}

**Path parameters**

* id (int)

**Request body**

{

"name": "Gaming Mouse",

"price": 39.99

}

**Notes**

* Full replace of updatable fields.
* Use If-Match with an ETag (if you enable optimistic concurrency).

**Responses**

* 204 No Content
* 400 Bad Request (validation)
* 404 Not Found
* 412 Precondition Failed (ETag mismatch, if using concurrency)

**Delete Product**

**DELETE** /products/{id}

**Path parameters**

* id (int)

**Responses**

* 204 No Content
* 404 Not Found

**Data Models**

**Product (response)**

{

"id": 1,

"name": "string",

"price": 0.0

}

**CreateProduct (request)**

{

"name": "string (1-100)",

"price": 0.0

}

**UpdateProduct (request)**

{

"name": "string (1-100)",

"price": 0.0

}

**Error (response)**

{

"error": "string-code",

"message": "Human-readable message.",

"details": { "field": ["issue"] }

}