LEARN SMART
CODING

SQL DATABASE DESIGN FROM SCRATCH

BY KARTHIK KANNAN

SUBSCRIBE LIKE

https://www.youtube.com/@learnsmartcoding

SQL Database design from scratch

LEARN SMART CODING

- *Softwares to install*
- *Basics of how to come up with entities for a application*
- *Full hands-on demo!*

Softwares to install

LEARN SMART CODING

Install SQL Server Database engine - We will use SQL Express

https://www.microsoft.com/en-us/sql-server/sql-server-downloads

Softwares to install

LEARN SMART CODING

Install SQL Server Management Studio (SSMS)

https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

What is SQL (Structured Query Language)

SQL (Structured Query Language) *is a standardized programming language used for managing and manipulating relational databases. It allows users to perform various operations such as querying data, inserting new records, updating existing records, and deleting records. SQL is essential for defining database structures, controlling access permissions, and ensuring data integrity through transactions. It also supports data aggregation and combination from multiple tables, making it a powerful tool for database management and application development.*

You will know all of the following at the end of the course.

- *.Net Core 8 using C# language.*
- *Entity Framework Core 8 and EF Tools, EF CLI*
- *SQL Database Design, from scratch to full.*
- *Async programming. e.g. async/await*
- *N-Tier Layer Architecture with Repository Patterns.*
- *Logging using Seri Log framework.*
- *Azure Application Insights*
- *Exception handling globally*
- *Authentication & Authorization using Azure AD B2C*

Getting Started

LEARN SMART CODING

- *What is a Web API?*
- *Web API Vs Microservice*
- *HTTP Status Codes*

What is a Web API?

LEARN SMART CODING

Definition

A Web API is a set of protocols and tools for building software applications. It allows different software systems to communicate with each other over the web. Web APIs expose endpoints (URLs) that can be accessed via HTTP methods (GET, POST, PUT, DELETE, etc.)

What is a Web API?

LEARN SMART CODING

Purpose

The primary purpose of a Web API is to enable interaction between disparate systems. For instance, a client application (like a web or mobile app) can interact with a server to perform CRUD (Create, Read, Update, Delete) operations on data.

What is a Web API?

LEARN SMART CODING

Components

- *Endpoints: URLs that represent resources (e.g., /api/users, /api/products).*
- *HTTP Methods: Operations that can be performed on these resources (e.g., GET to retrieve data, POST to create data).*
- *Request and Response: Data sent to and received from the API, typically in JSON format.*

What is a Web API?

LEARN SMART CODING

Example

An e-commerce application might have a Web API with endpoints like:

- *GET /api/products - Retrieve a list of products.*
- *POST /api/orders - Create a new order.*
- *PUT /api/users/1 - Update the user with ID 1*

What is a Microservice?

LEARN SMART CODING

Definition

A microservice is a small, independent service that does one thing well. It is a part of a larger architecture where multiple microservices work together to form a complete system

What is a Microservice?

LEARN SMART CODING

Scope

Microservices break down an application into smaller, loosely coupled services. Each microservice runs in its own process and communicates with other microservices through APIs (often Web APIs).

Example Scenario

- *Web API: A large monolithic e-commerce application has a Web API to handle user requests. The Web API exposes endpoints for products, orders, and users.*
- *Microservices: The same e-commerce application is broken down into multiple microservices:*
- *Product Service: Manages product-related operations.*
- *Order Service: Handles order processing.*
- *User Service: Manages user accounts. Each of these microservices might expose its own Web API for communication with other services and clients.*

LEARN SMART CODING

Summary

A Web API is a means of communication over the web, while a microservice is an architectural style that involves building small, independent services that often communicate via APIs.

What are HTTP Status Codes?

HTTP status codes are standardized codes returned by web servers to indicate the result of a client's request. They are grouped into five categories:

What are HTTP Status Codes?

LEARN SMART CODING

- 1xx (Informational): Request received, continuing process.
- 2xx (Successful): The request was successfully received, understood, and accepted.
- 3xx (Redirection): Further action needs to be taken in order to complete the request.
- 4xx (Client Error): The request contains bad syntax or cannot be fulfilled.
- 5xx (Server Error): The server failed to fulfill an apparently valid request.

What are HTTP Status Codes?

LEARN SMART CODING

Status Code	Category	Description
1xx (Informational)		
100	Continue	The server has received the request headers and the client should proceed to send the request body.
101	Switching Protocols	The requester has asked the server to switch protocols and the server is acknowledging that it will do so.
2xx (Successful)		
200	OK	The request has succeeded.
201	Created	The request has been fulfilled and resulted in a new resource being created.
202	Accepted	The request has been accepted for processing, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.
204	No Content	The server successfully processed the request, but is not returning any content.
205	Reset Content	The server successfully processed the request, but is not returning any content and requires the requester to reset the document view.
206	Partial Content	The server is delivering only part of the resource due to a range header sent by the client.

What are HTTP Status Codes?

3xx (Redirection)		
300	Multiple Choices	The request has more than one possible response.
301	Moved Permanently	The URL of the requested resource has been changed permanently.
302	Found	The URL of the requested resource has been changed temporarily.
303	See Other	The server is redirecting to a different URL using the GET method.
304	Not Modified	The resource has not been modified since the last request.
307	Temporary Redirect	The requested resource resides temporarily under a different URL.
308	Permanent Redirect	The requested resource has been moved to a new permanent URL.
4xx (Client Error)		
400	Bad Request	The server cannot or will not process the request due to a client error.
401	Unauthorized	Authentication is required and has failed or has not yet been provided.
403	Forbidden	The server understands the request but refuses to authorize it.
404	Not Found	The requested resource could not be found.
405	Method Not Allowed	The request method is not supported for the requested resource.

What are HTTP Status Codes?

406	Not Acceptable	The requested resource is capable of generating only content not acceptable according to the Accept headers sent in the request.
407	Proxy Authentication Required	The client must first authenticate itself with the proxy.
408	Request Timeout	The server timed out waiting for the request.
409	Conflict	The request could not be processed because of conflict in the request.
410	Gone	The resource requested is no longer available and will not be available again.
411	Length Required	The request did not specify the length of its content, which is required by the requested resource.
412	Precondition Failed	The server does not meet one of the preconditions that the requester put on the request.
413	Payload Too Large	The request is larger than the server is willing or able to process.
414	URI Too Long	The URI provided was too long for the server to process.
415	Unsupported Media Type	The request entity has a media type which the server or resource does not support.
416	Range Not Satisfiable	The client has asked for a portion of the file (byte serving), but the server cannot supply that portion.
417	Expectation Failed	The server cannot meet the requirements of the Expect request-header field.
418	I'm a teapot	Defined in 1998 as an April Fools' joke in RFC 2324, Hyper Text Coffee Pot Control Protocol.
422	Unprocessable Entity	The request was well-formed but was unable to be followed due to s ntic errors.

What are HTTP Status Codes?

LEARN SMART CODING

426	Upgrade Required	The client should switch to a different protocol.
5xx (Server Error)		
500	Internal Server Error	A generic error message, given when an unexpected condition was encountered.
501	Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server.
503	Service Unavailable	The server is currently unavailable (because it is overloaded or down for maintenance).
504	Gateway Timeout	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server.
505	HTTP Version Not Supported	The server does not support the HTTP protocol version used in the request.

Asynchronous programming in .Net Core

LEARN SMART CODING

What is Async/Await in C# ?

Definition: *async and await are keywords in C# that facilitate asynchronous programming. They allow you to write asynchronous code that appears synchronous, making it easier to read and maintain.*

Asynchronous programming in .Net Core

LEARN SMART CODING

What is Async/Await in C# ?

Purpose: *Asynchronous programming is used to improve the responsiveness of applications by performing time-consuming operations without blocking the main thread. This is particularly useful for I/O-bound operations like web requests, file access, and database queries.*

Asynchronous programming in .Net Core

LEARN SMART CODING

How Async/Await Works?

Async Method:

- *Mark a method with the async keyword to indicate it contains asynchronous operations.*
- *An async method usually returns a Task, Task<T>, or void (for event handlers).*

Asynchronous programming in .Net Core

LEARN SMART CODING

How Async/Await Works?

Await Keyword:

- *Inside an async method, use the await keyword to call an asynchronous method.*
- *await pauses the execution of the method until the awaited task completes, without blocking the main thread.*

Asynchronous programming in .Net Core

LEARN SMART CODING

Example Usage

Here's a simple example of using `async` and `await` to make an asynchronous web request:

csharp

Copy code

```csharp
public async Task<string> FetchDataFromApiAsync(string url)
{
    using (HttpClient client = new HttpClient())
    {
        HttpResponseMessage response = await client.GetAsync(url);
        response.EnsureSuccessStatusCode();
        string responseData = await response.Content.ReadAsStringAsync();
        return responseData;
    }
}
```

Asynchronous programming in .Net Core

LEARN SMART CODING

Proper usage of async/await

1. *Avoid Blocking Calls: Don't use .Result or .Wait() on a task in an async method, as it can lead to deadlocks.*
2. *Return Types:*

- *For non-generic tasks, return Task.*
- *For methods returning a result, return Task<T>.*
- *Use void only for event handlers.*

Asynchronous programming in .Net Core

LEARN SMART CODING

Proper usage of async/await

1. *Error Handling: Use try-catch blocks to handle exceptions in async methods. Awaited calls should be wrapped in try-catch to catch exceptions thrown by the task.*
2. *Async All the Way: Ensure that all methods in the call chain are async. Mixing synchronous and asynchronous code can cause performance issues and deadlocks.*

Asynchronous programming in .Net Core

LEARN SMART CODING

Example with Proper Usage

csharp

Copy code

```csharp
public async Task ProcessDataAsync(string url)
{
    try
    {
        string data = await FetchDataFromApiAsync(url);
        // Process the data
    }
    catch (HttpRequestException ex)
    {
        // Handle specific exception
    }
    catch (Exception ex)
    {
        // Handle general exceptions
    }
}
```

Asynchronous programming in .Net Core

LEARN SMART CODING

Benefits of Async/Await

1. ***Improved Responsiveness:*** *Async methods keep the UI responsive by offloading time-consuming operations.*
2. ***Scalability:*** *Asynchronous code allows better resource utilization, which is crucial for scalable applications.*
3. ***Readability:*** *Code using async and await is more readable and maintainable compared to traditional asynchronous programming models.*

Asynchronous programming in .Net Core

Customers

Shop Owner

Worker

Asynchronous programming in .Net Core

Tips

Put your hands on deck!

Practice, practice, and practice to succeed!

HANDS ON PRACTICE

LEARN SMART CODING

VISIT HTTPS://GITHUB.COM/LEARNSMARTCODING

COMPLETE CODE FOR .NET CORE 8 WEB API CRASH COURSE | EXPENSE TRACKER IS AVAILABLE IN THE BELOW LINK

HTTPS://GITHUB.COM/LEARNSMARTCODING/EXPENSE-TRACKER-WEB-API

BY KARTHIK KANNAN

https://www.youtube.com/@learnsmartcoding

THANKS FOR WATCHING

BY KARTHIK KANNAN

SUBSCRIBE LIKE

https://www.youtube.com/@learnsmartcoding