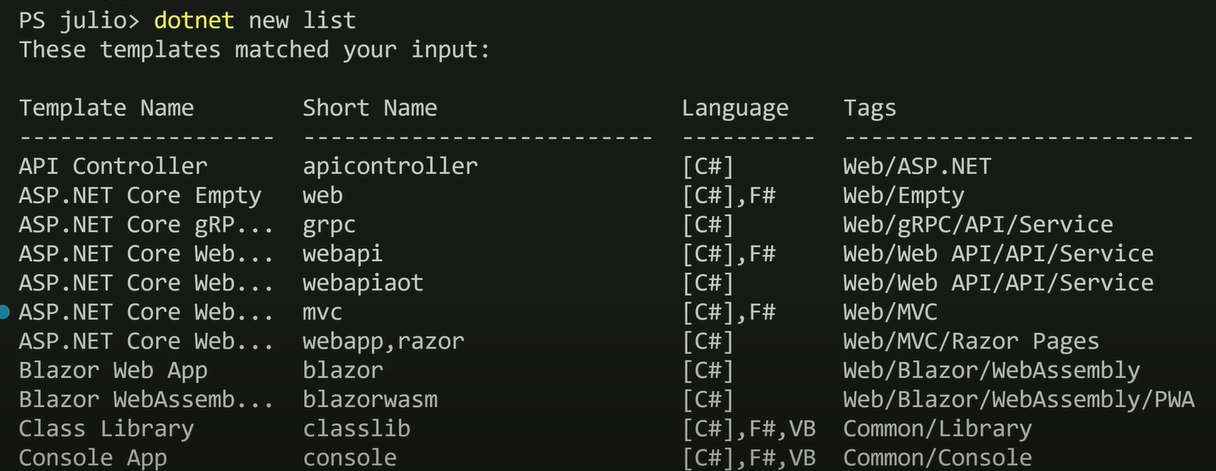
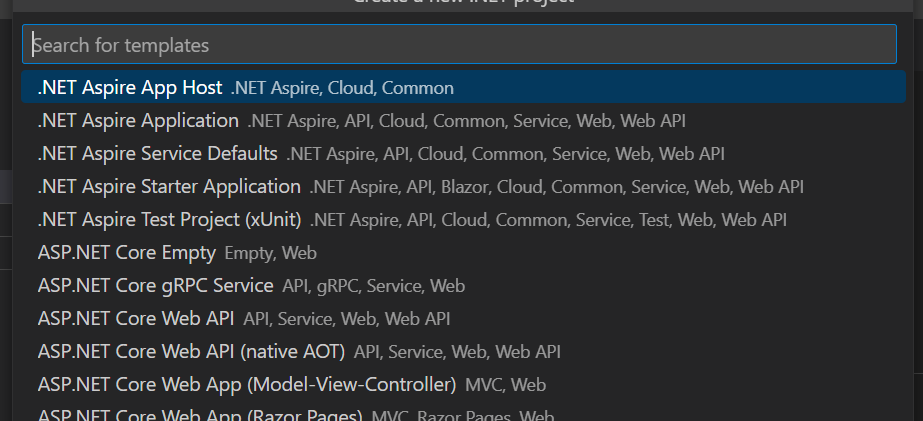
**Create Asp.Net Core apps**

Install the below extensions at VS Code

1. C# Dev Kit
2. REST Client
3. SQLite

Verify the .NET is installed properly. Press **Ctrl + J** to open the terminal at VS Code and type   
dotnet –version. It will show the installed .Net version.

Create New project

1. From terminal:
   1. To show all installed template “dotnet new list”  
      
2. From command plate: (Ctrl + Shift +P)
   1. .NET:New Project  
      

How to build/run a program:

* + - **dotnet build** (build the program)
    - To build press F5 or **dotnet run**

**Short Cut:**

* Command Palate: Ctrl + Shift + P
* Open Terminal: Ctrl +J
* Copilot Inline chat: Ctrl + I
* Copilot full chat: Ctrl + Shift + I

**Understand REST APIs**

Representational State Transfer. A set of guiding principles that impose conditions on how an API should work.

Characteristics

1. Stateless
2. Client-Server Model
3. Uniform interface
4. Layered system (A REST API can use multiple layers (e.g., load balancers, security layers) without impacting client-server communication.)
5. Cacheable (Responses must define themselves as cacheable or non-cacheable to improve efficiency)
6. Code on demand (Servers can send executable code (e.g., JavaScript) to the client to enhance functionality)

HTTP Methods:

1. POST: Creates a new resource (**C**reate)
2. GET: Retrieves the resources (**R**ead)
3. PUT: Update an existing resource (**U**pdate)
4. DELETE: Delete a resource (**D**elete)
5. PATCH: Partially update a resource

HTTP Status:

1. 200 – OK (Get)
2. 201 – Created (Post)
3. 204 – No Content (PUT)
4. 404 - Not Found
5. 400 – Bad Request
6. 500 – Internal Server Error

**Data Transfer Objects (DTOs)**

* **DTOs** focus on data transfer and keep it simple. The **Controller Layer** returns DTOs to the client.
* **Data Models** focus on representing domain objects and can include behaviour logic. The **Service Layer** processes data using data models.

- Implement CRUD Endpoints

- Extension Methods

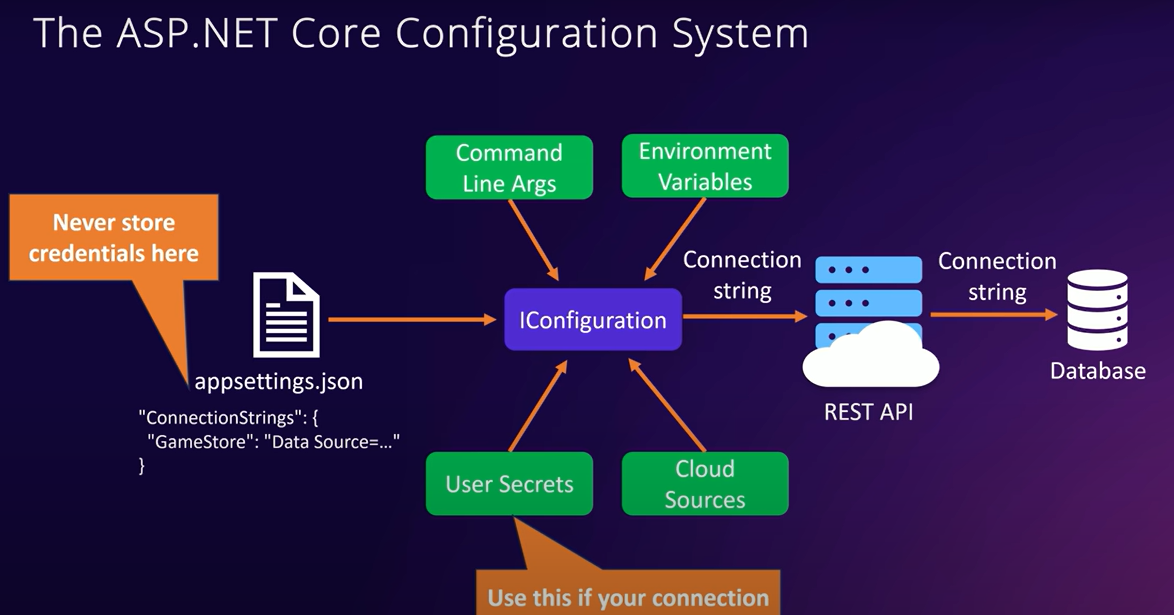
- Route Groups

- Handle Invalid Inputs

* Data Annotation at DTOs
* Endpoint filter (MinimulApis.Extensions (Nuget package))
* dotnet add package MinimalApis.Extensions --version 0.11.0
* Install Nuget Package manager and Nuget Gallary Extension for VS Code

- Entity Framework Core

* dotnet add package **Microsoft.EntityFrameworkCore.Sqlite** --version 9.0.2
* dotnet tool install **--global dotnet-ef** --version 9.0.2
* dotnet add package **Microsoft.EntityFrameworkCore.Design** --version 9.0.2
* List installed package: dotnet list package
* Use IConfiguration object to store all the configuration
* SqLite don’t need any credentials
* Store all credentials at User Secrets. Never store credentials at appsettings.json file

**- Configuration System**

- Dependency Injection

- Service Lifetime

- Mapping Entities to DTOs

- Asynchronous Programming

- Frontend Integration