Difference between TCP AND Http

The difference between TCP (Transmission Control Protocol) and HTTP (Hypertext Transfer Protocol):

1. **Protocol Type:**
   * **TCP (Transmission Control Protocol):** TCP is a transport-layer protocol used for establishing a reliable connection between two devices in a network. It ensures that data is delivered accurately and in the correct order.
   * **HTTP (Hypertext Transfer Protocol):** HTTP is an application-layer protocol used for transferring data, primarily for web page retrieval and communication between a web client (e.g., a browser) and a web server.
2. **Layer of the OSI Model:**
   * **TCP:** It operates at the transport layer (Layer 4) of the OSI model.
   * **HTTP:** It operates at the application layer (Layer 7) of the OSI model.
3. **Purpose:**
   * **TCP:** TCP is a low-level protocol responsible for reliable data transfer and connection management. It handles things like establishing and terminating connections, flow control, and error detection and correction.
   * **HTTP:** HTTP is a high-level protocol used for requesting and delivering web content. It defines how web browsers and web servers communicate to retrieve and display web pages, images, videos, etc.
4. **Connection:**
   * **TCP:** It establishes a connection-oriented communication. It creates a connection between two devices before data transfer and ensures the data is delivered reliably.
   * **HTTP:** HTTP uses TCP as its underlying transport protocol. When you access a website, your browser establishes a TCP connection to the web server and then sends HTTP requests to retrieve web content.
5. **Data Format:**
   * **TCP:** TCP doesn't specify the format or structure of data being transmitted. It's a generic protocol for reliable data transport.
   * **HTTP:** HTTP defines how requests and responses should be formatted. HTTP requests include methods like GET, POST, PUT, and DELETE, while responses typically include HTML, JSON, or other data formats.
6. **Port Numbers:**
   * **TCP:** It uses port numbers to identify different services or applications. For example, HTTP typically uses port 80 for non-secure connections and port 443 for secure (HTTPS) connections.
   * **HTTP:** HTTP itself doesn't use specific port numbers, but it operates over the TCP ports mentioned above.

In summary, TCP is a lower-level protocol responsible for reliable data transfer, while HTTP is a higher-level application protocol used for web communication. HTTP utilizes TCP as its transport protocol to ensure data is delivered reliably over the internet.

|  |  |  |
| --- | --- | --- |
|  | **TCP** | **HTTP** |
| **Protocol Type** | TCP is a transport-layer protocol used for establishing a reliable connection between two devices in a network. It ensures that data is delivered accurately and in the correct order. | HTTP is an application-layer protocol used for transferring data, primarily for web page retrieval and communication between a web client (e.g., a browser) and a web server. |
| **Layer of the OSI Model** |  |  |

Entity Framework

Get-Help about\_EntityFrameworkCore

* To check wether we have installed the extensions for entityframework.

The following Entity Framework Core commands are available.

Cmdlet Description

-------------------------- ---------------------------------------------------

Add-Migration Adds a new migration.

Bundle-Migration Creates an executable to update the database.

Drop-Database Drops the database.

Get-DbContext Lists and gets information about available DbContext types.

Get-Migration Lists available migrations.

Optimize-DbContext Generates a compiled version of the model used by the DbContext.

Remove-Migration Removes the last migration.

Scaffold-DbContext Scaffolds a DbContext and entity types for a database.

Script-DbContext Generates a SQL script from the DbContext. Bypasses any migrations.

Script-Migration Generates a SQL script from migrations.

Update-Database Updates the database to a specified migration.

**Command for reverse engineering**

* **Unsecure,**
  + **Scaffold-DbContext -provider Microsoft.EntityFrameworkCore.SqlServer -connection “server=** **200411LTP2860\SQLEXPRESS;database=Northwind;integrated security=true;Encrypt=false;” -OutputDir Models**
* **More secure is,**
  + Scaffold-DbContext -provider Microsoft.EntityFrameworkCore.SqlServer -connection name="NWindConnection" -OutputDir Models

**Cascade delete**

* **If parent deleted then all referencing child record will be at one go deleted**
* **Disable it very very dangerous**
* **When to use?**
  + **While testing**

**Cascade update**

* **If primary key value is updated then its updated in all the childs foreign keys**
* **Very useful and required**

**Connection Pooling**

**SQL connection pool**

**Context object is a wrapper to sql connection**

**From(Foreach) o in table where(if ) bla bla select() 0**

**15 on SQL alone**

**And then last weeks on Wednesday delegate event thread task library arrays and collection generics ef link MVC dependency injection**