

Yesterday:-

1. Single Responsibility Principle
2. Dependency Injection Pattern
3. Factory Pattern.
4. Singleton Pattern
5. DRY - Do not Repeat Yourself Pattern
6. Model - View - Controller Pattern. [MVC]
7. Model - ViewModel - View - Controller Pattern  
[MVVM]
8. Repository Pattern

1. Strongly - typed Views
2. @model - directive } in Views. cshtml.
3. @using - directive }
4. Model property
5. diff. model class & View Model class. [mvvm]
6. bootstrap installation.
7. Layout Pages, Layout Property

Client / Home / Index

Port 80: HTTP / Neutral

## HTTP Pipeline

RouteData Dictionary  
ref. URL = "Home / Index"  
URL split() = {"Home", "Index"}

"controller"	"Home"
"action"	"Index"

→ Reflection Use

ControllerFactory obj = new CF();

ctrlName = "Home" + "controller";

Controller controller = obj.getController(ctrlName);

actionName = "Index"; "Edit"

[ FormCollection + Model Entity = initialized  
(class ModelBinder - Bind()) → Model along with values ]

Start Action Filter ✓

controller.InvokeMember("action",  
new object[] { ↓ 3 } )

// call to viewEngine() - RazorViewEng.

Render()

End Action Filter

Response Packet.

ViewEngine  
obj = new VE  
("Razor").

obj.Render()  
ctrl + HTML → op

Start Result Filter → 1) - ViewStart.razor  
2) View -

End Result Filter ← 3) op - HTML Tags

- HTTP Pipeline
- Routing
  - Controller Factory
  - Action Filter
  - Model Binder
  - InvokeMember
  - Load RazorViewEngine
  - Result Filter
    - ViewStart, cshtml
    - Result Filter
    - Action Filter
    - Response  $\rightarrow$  o/p

Services:-

1) AddSingleton: Creates a single instance for the entire application.  
e.g:- logger

2) AddScoped: Creates an instance per HTTP Request.  
e.g:- DbContext, filters

3) AddTransient: Creates a new instance each time it is required  
e.g:- email service

---

**DI Container** - A tool that manages the creation, configuration, and lifetimes of objects and their dependencies

Client Name/Inte 80 IS / Restra  
- Routing.

### DI Container

Singleton,	Scoped,	Transient Services
class Logger	dbContent filter	myEmailService

Your Application object  
along with dependencies & Config

Middlewares

app. Run

- Controller Factory.  
↳ o/p.