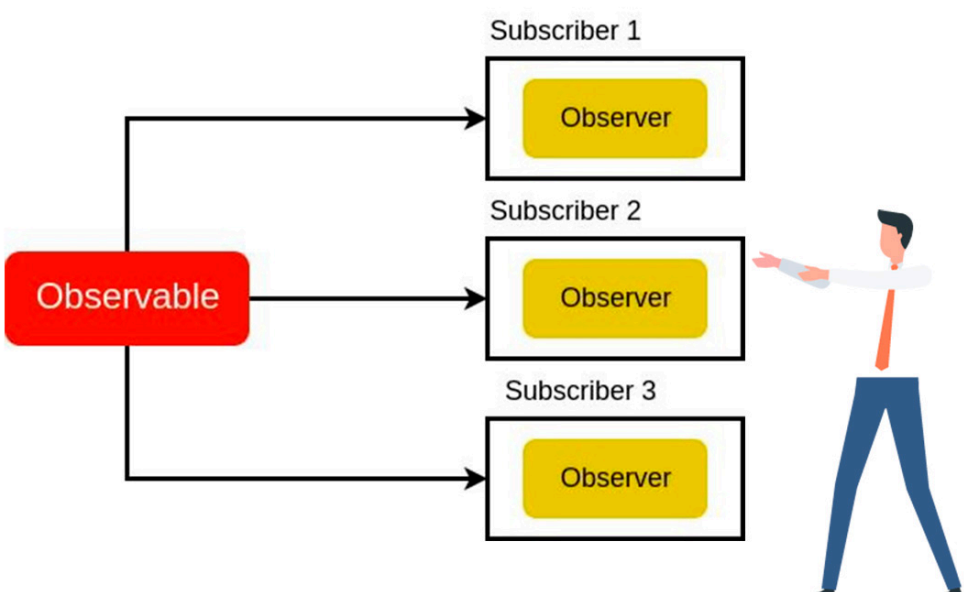

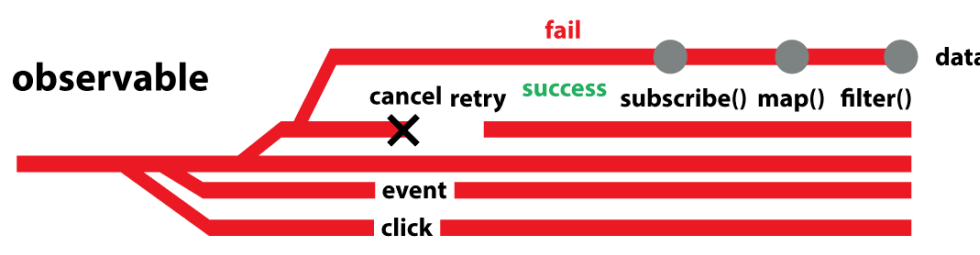


<p>Day 22</p>	<p>Advanced Angular Directives and Pipes Component Styling and Communication</p>	<p>Cont.. HTTP & Observables</p> <p>HTTP in Angular : the <code>HttpClient</code> module allows us to perform HTTP request (GET, POST, PUT, DELETE) to communicate with back ends APIs</p> <p>Observables : A data stream that emits multiple values over a time, used to handle asynchronous data in Angular .</p>  <p>The diagram illustrates the Observable pattern. A red box labeled 'Observable' has three arrows pointing to three separate boxes, each labeled 'Subscriber' (Subscriber 1, Subscriber 2, Subscriber 3). Inside each subscriber box is a yellow box labeled 'Observer'. A person is standing next to Subscriber 2, pointing towards the Observer box.</p> <hr/> <div data-bbox="430 1344 1510 1816"> <p>promise</p>  <p>observable</p>  <p>The observable diagram shows a sequence of operations: 'cancel' (with a red X), 'retry', 'subscribe()', 'map()', and 'filter()', leading to 'data'. There are also 'event' and 'click' nodes branching off the main line. A 'fail' path is also shown branching off the main line.</p> </div>
---------------	--	---

		<p>Pros and cons :</p> <p>Works well for asynchronous and event based programming.</p> <p>Support real time streaming</p> <p>Easy to subscribe to prevent memory leaks.</p> <p>It can be complex when chaining multiple observables.</p> <p>Demo : HTTP based observables in Angular</p> <p>DataService — central place for HTTP calls (GET, POST) returning Observables</p> <p>AuthInterceptor — adds a demo Authorization header to every request</p> <p>PostsComponent — displays posts, demonstrates both manual subscribe (with proper cleanup) and async-pipe usage</p> <p>Step 1: Creating an Angular Project</p> <p>Step 2: Adding HTTPClientModule in app.module.ts</p> <p>Step 3: Create a service for HTTP calls (ng generate service data)</p> <p>Step 4: Inject the service into the component</p> <p>Step 5: Update the template (HTML)</p> <p>Step 6: Run the app</p> <p>HTTP interceptor : a service that can inspect and transform HP request.</p> <p>- Routing in an Angular application</p>
--	--	---

Demo based on different types of Angular Pipes

Step 1: Create an Angular application

Step 2: Creating a demo component with pipes implementation

Day 23	Angular Advanced	Fetch Data Using HTTP
		Error Handling using HTTP
Day 24	Angular Advanced & Angular Forms	Template Driven Forms
		Reactive Forms

16th of Aug : Milestone 2

18th of Aug .NET Core (Backend creation using MVC, DB connectivity(MS SQL) [ADO.NET/EF](#))

Milestone 3

Power platform & power Apps

Azure Cloud Introduction _ No Cloud credentials

Capstone project : 03 Sept -06th Sep

Final milestone: 8th Sept

Microsoft Tech Stack (

C# - Basic programming

ASP.NET

MS SQL Server

.NET Core

Testing using Microsoft frameworks (Nunit, Xunit, MS test)

Power Platform for working with virtual agent

Azure Cloud

Security & pentesting

Day 1 : Discussing problem statement & Evaluation rubrics

Day 2: Show some partial progress Database diagram, ER Diagram , any Blueprint diagram

Front End : Angular + HTML CSS, Js + typescript

Database: MS SQL Server

Back End : APi creation using .NET Core + Node.js,

Testing :

Login & authentication:

Reporting

Pro & Cons of Capstone Project :

Best Practices for your first capstone project:

If you are a .NET Developer working on HMS based on

Front End : Angular + HTML CSS, Js + typescript

Database: MS SQL Server

Back End : APi creation using .NET Core + Node.js,

Testing :

Login & authentication:

Reporting

How will you divide your work ?

What all diagrams can you create to get started ?

What will be the deliverables of the above project ?

How will you divide above work in a group of 5 developers ?