# Session 9

# Working with Forms, HTTP, REST, and Animation APIs

# Session Objectives

- ✓ Describe an overview of forms in Angular 9
- ✓ Explain how to work with Angular 9 forms
- ✓ Explain how to create and consume a REST API with HTTP in Angular application
- ✓ Outline steps to create an Animation based application in Angular 9

# Forms in Angular 9

**Angular Forms**

Reactive Forms

Template-driven Forms
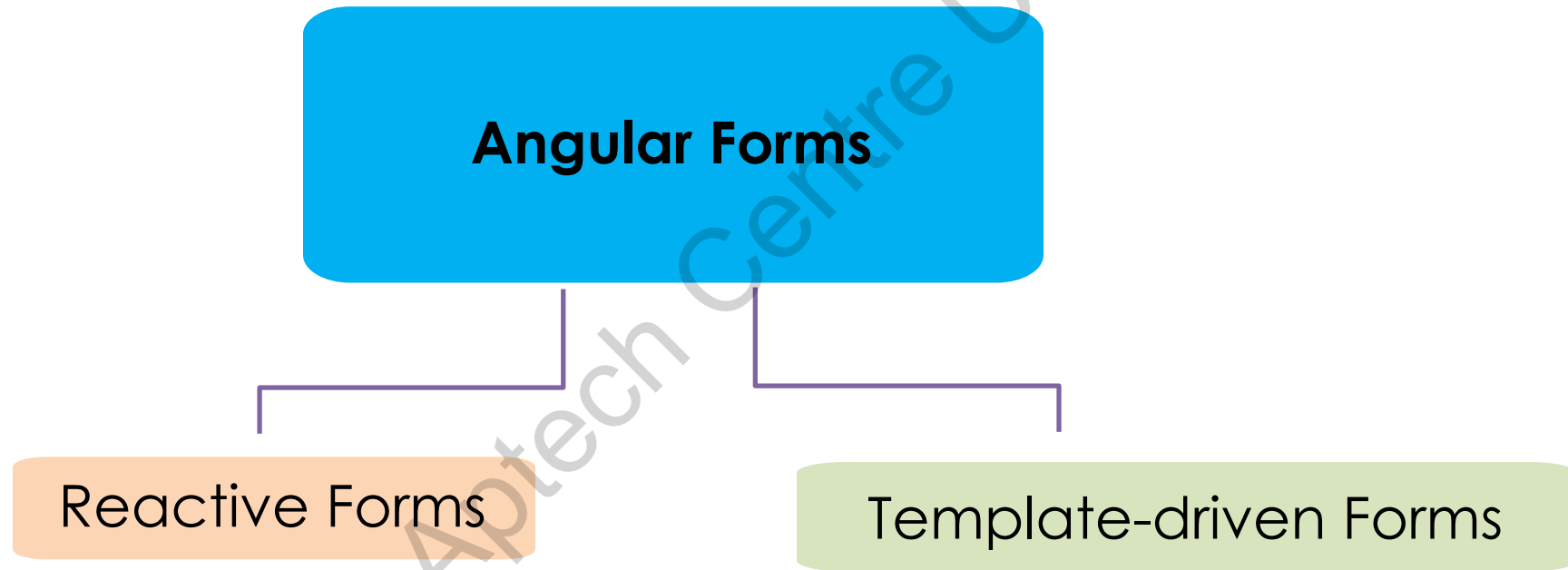
# Forms in Angular 9

## Reactive Forms

- Scalable, reusable, and testable
- More Robust
- Structured Data Model
- Form Validation through functions

## Template-driven Forms

- Simple forms with logic manageable in template
- Not suitable for much scaling
- Unstructured data model
- Form Validation through directives

# Forms in Angular 9

Features common in both include same essential building blocks, which are as follows:

FormGroup tracks the same values and status for a collection of form controls.

FormControl tracks the value and validation status of an individual form control.
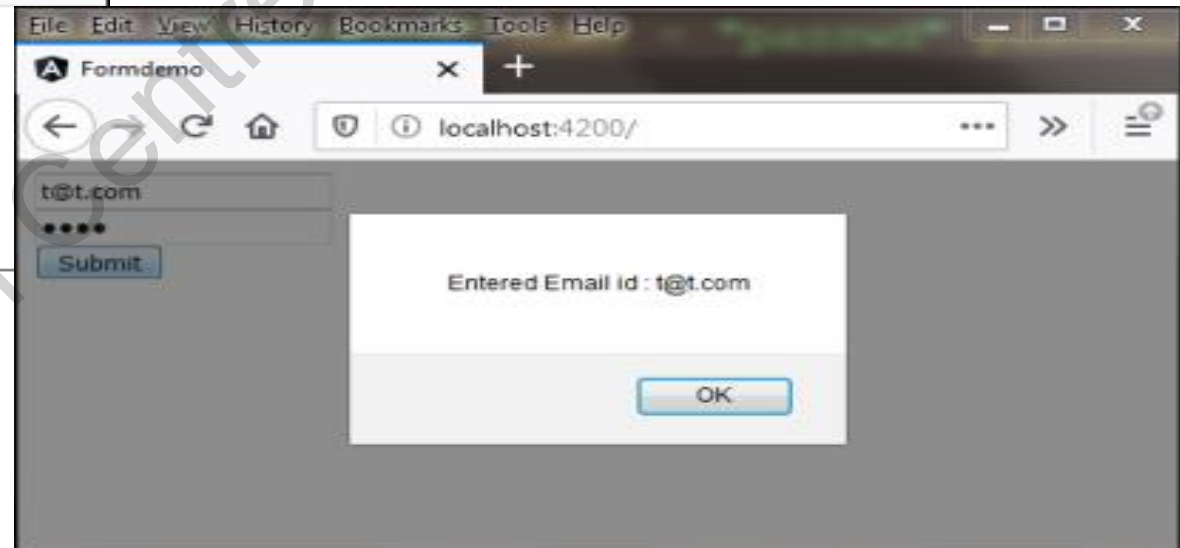
FormArray tracks the same values and status for an array of form controls.

ControlValueAccessor creates a bridge between Angular FormControl instances and native DOM elements.

# Form Demo Example
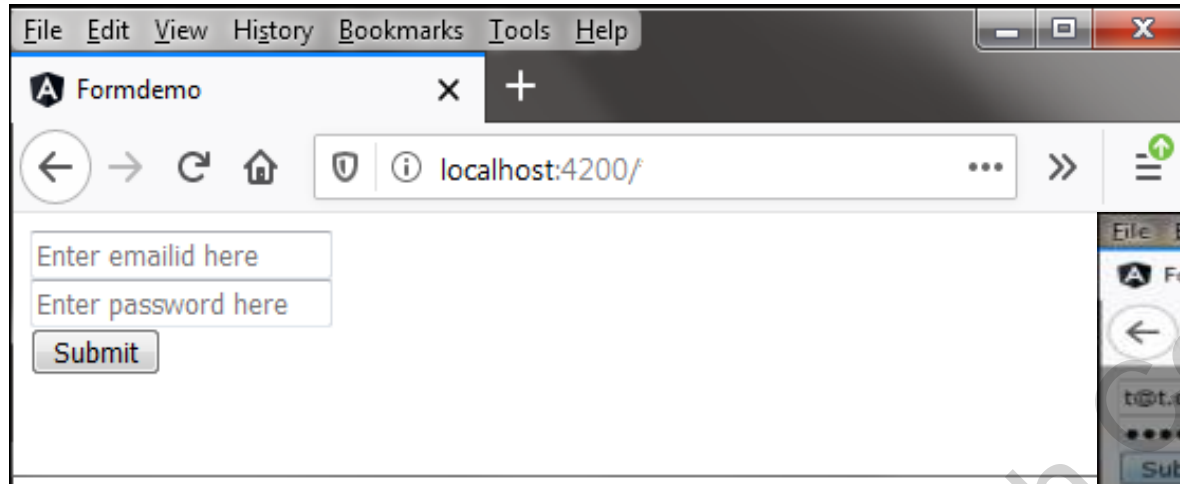


*FormDemo Application – Initial View*

*FormDemo Application After Email Id Supplied*

# Form with Additional Components

# HTTP Communication in Angular 9

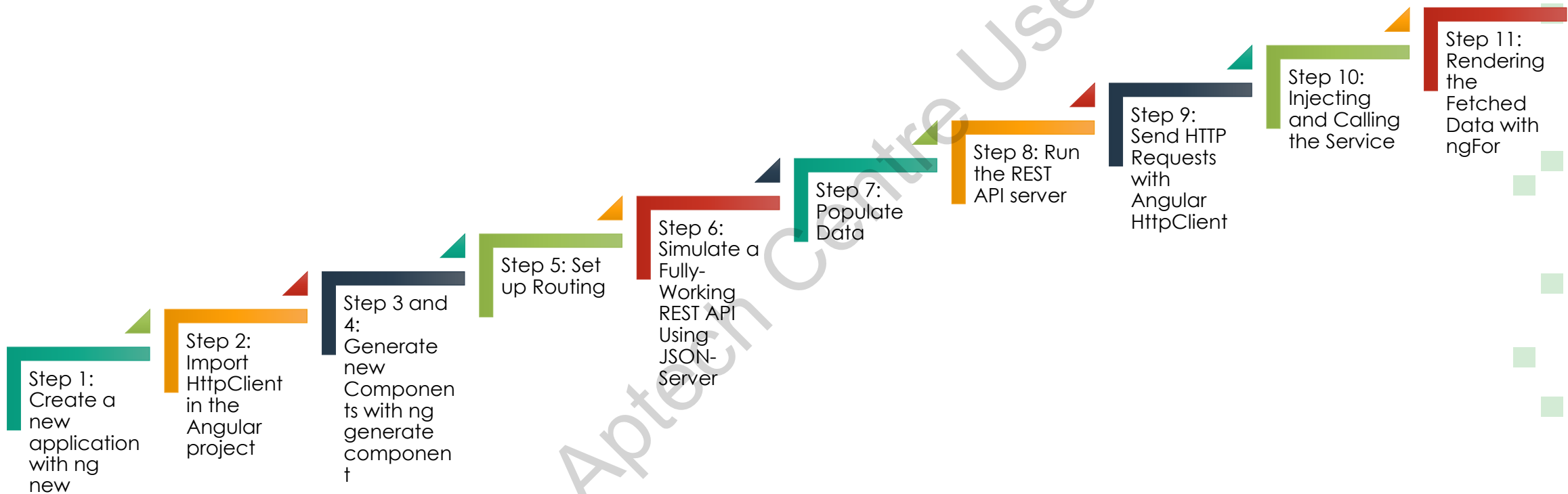- Angular provides HttpClient service to communicate with a remote server over HTTP.

- Using this class, you can fetch data in your Angular application from a REST API server.

- The primary purpose of HttpClient is to facilitate HTTP requests. This class provides various methods, using which developers can perform different kinds of requests.

**HTTP**

Web Browser

Angular application

Remote Web Server

Step 1: Create a new application with ng new

Step 2: Import HttpClient in the Angular project

Step 3 and 4: Generate new Components with ng generate component

Step 5: Set up Routing

Step 6: Simulate a Fully-Working REST API Using JSON-Server

Step 7: Populate Data

Step 8: Run the REST API server

Step 9: Send HTTP Requests with Angular HttpClient

Step 10: Injecting and Calling the Service

Step 11: Rendering the Fetched Data with ngFor

A route is a JavaScript object containing properties such as:

| path | contains the path segment of the URL of the route |
| pathMatch | specifies how the router should match the path |
| redirectTo | specifies the path to redirect to |
| component | specifies the component to associate with a specific path |

# Faker

Faker.js is a readymade JavaScript library that generates massive amounts of realistic fake data in Node.js and the browser.

Some of the databases available with readymade data in Faker:

- address
- commerce
- company
- database
- date
- finance
- hacker
- helpers
- image
- Internet

# Working with Animation API in Angular 9

## Animations API in Angular:

- Provides a **declarative API** to build and reuse animations throughout components.
- Is not part of default Angular installation
- Is shipped as a separate package.
- Should be installed as follows:

```
npm install angular/animations
```
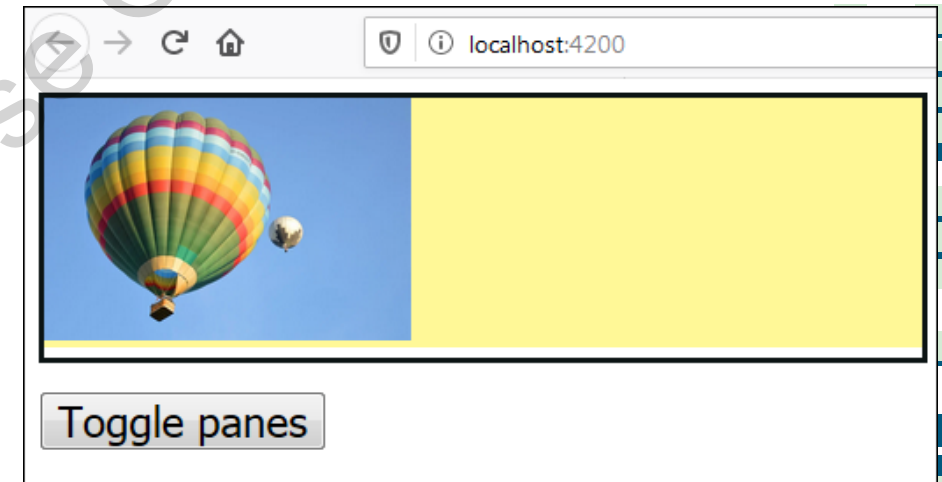
# Summary

➢ Angular 9 provides two types of approaches for handling user input through forms: reactive and template-driven.

➢ Reactive Forms are scalable, reusable, and testable, more robust and have a structured data model.

➢ Template-driven forms are simple forms with logic manageable in template, they are not suitable for much scaling, and have an unstructured data model.

➢ When we import FormsModule in our application, the ngForm directive becomes active by default on all <form> tags.

➢ The ngSubmit event is notified when user triggers a form submission by clicking Submit button.

➢ Angular provides the HttpClient service as an injectable class for communicating with a remote server over HTTP. The primary purpose of HttpClient is to perform HTTP requests.

➢ Tools such as json-server enable to simulate a fully working backend API.

➢ The Animations API in Angular provides a declarative API to build and reuse animations throughout our components.