Angular 4

By Nisha Waikar

What is Angular 4?

- Angular 4 is a JavaScript framework for building web applications and apps in JavaScript, html, and TypeScript, which is a superset of JavaScript.
- Angular provides built-in features for animation, http service, and materials which in turn has features such as auto-complete, navigation, toolbar, menus, etc.
- The code is written in TypeScript, which compiles to JavaScript and displays the same in the browser.

Differences between Angular and AngularJS

- The architecture of an Angular application is different from AngularJS. The main building blocks for Angular are modules, components, templates, metadata, data binding, directives, services and dependency injection.

- Angular was a complete rewrite of AngularJS.

- Angular does not have a concept of “scope” or controllers instead, it uses a hierarchy of components as its main architectural concept.

Differences between Angular and AngularJS

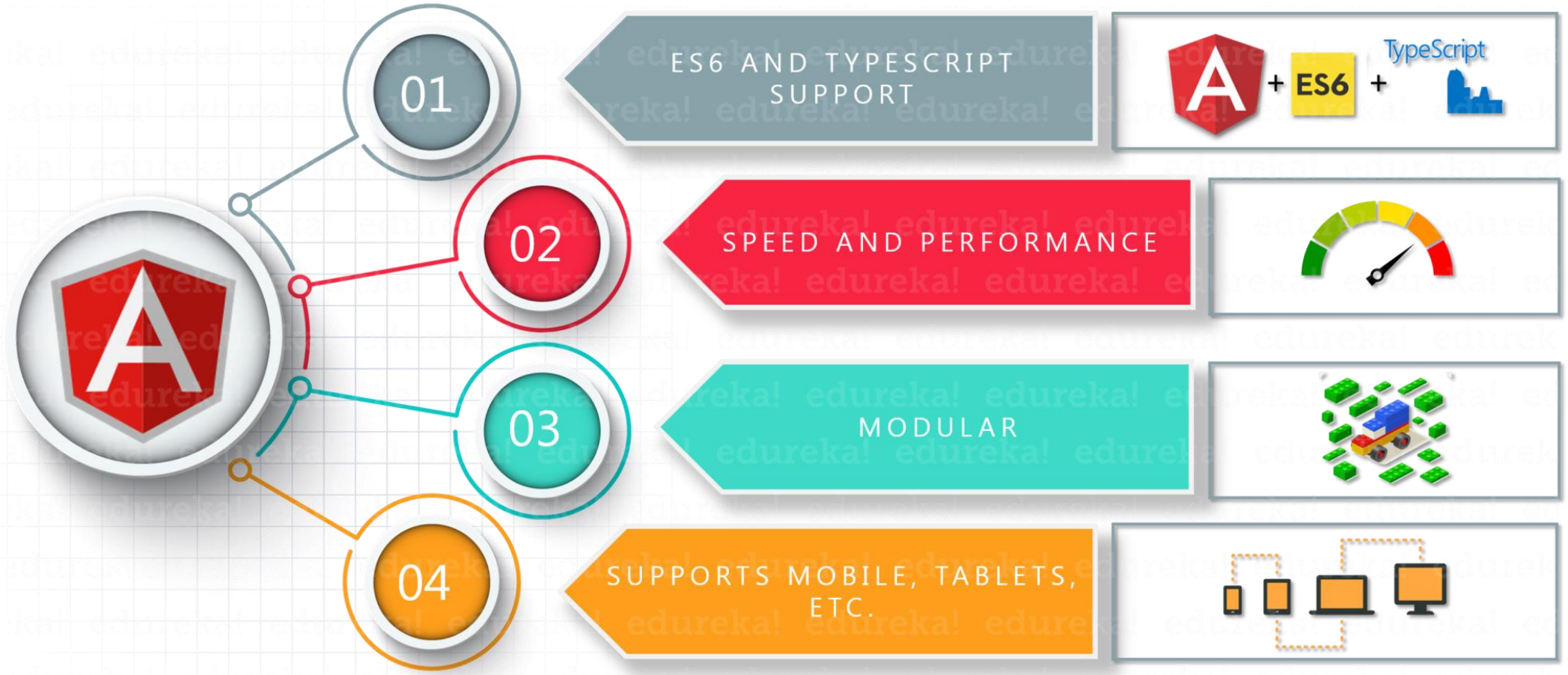
- Angular has a simpler expression syntax, focusing on “[ ]” for property binding, and “( )” for event binding

- **Mobile development** – Desktop development is much easier when mobile performance issues are handled first. Thus, Angular first handles mobile development.

- **Modularity** – Angular follows modularity. Similar functionalities are kept together in same modules. This gives Angular a lighter & faster core.

Features of Angular

edureka!



How to install Angular4?

- To install Angular 4, we require the following
  - Nodejs – install Node JS to check version use “**node – v**” command
  - Npm – to check npm version use “**npm –v**” command
  - Angular CLI – use “**npm install –g @angular/cli”**, to install angular cli on your system.
- Create a folder to store codes.
- Go to that folder location on command prompt and execute “**ng new Angular 4-app**” comm and. (Specify your project name here instead of ‘Angular 4-app’)
- Start server using “**ng serve**” command

Folder structure in the project

- **e2e** – end to end test folder. Mainly e2e is used for integration testing and helps ensure the application works fine.

- **node_modules** – The npm package installed is node_modules. You can open the folder and see the packages available.

- **src** – This folder is where we will work on the project using Angular 4. The src folder is the main folder, which internally has a different file structure.

File structure in project

- **.angular-cli.json** – It basically holds the project name, version of cli, etc.
- **.editorconfig** – This is the config file for the editor.
- **.gitignore** – A .gitignore file should be committed into the repository, in order to share the ignore rules with any other users that clone the repository.
- **karma.conf.js** – This is used for unit testing via the protractor. All the information required for the project is provided in karma.conf.js file.
- **package.json** – The package.json file tells which libraries will be installed into node_modules when you run npm install.

Folder structure in the project

- The app folder contains following files.
  - **app.module.ts :** Angular-cli has used these default libraries for the import – angular/core, platform-browser. The names itself explain the usage of the libraries.
  - **app.component.css :** You can write your css structure over here.
  - **app.component.html :** This is the default html code currently available with the project creation.
  - **app.component.spec.ts :** These are automatically generated files which contain unit tests for source component.
  - **app.component.ts :** The class for the component is defined over here. You can do the processing of the html structure in the .ts file.

Folder structure in the project

- Assets
  - You can save your images, js files in this folder.
- Environment
  - This folder has the details for the production or the dev environment. The folder contains two files. Both the files have details of whether the final file should be compiled in the production environment or the dev environment.
    - environment.prod.ts
    - environment.ts

File structure in project

- favicon.ico
  - This is a file that is usually found in the root directory of a website.
- index.html
  - This is the file which is displayed in the browser.
- main.ts
  - This is the file from where we start our project development. It starts with importing the basic module which we need.

Architecture of Angular Apps

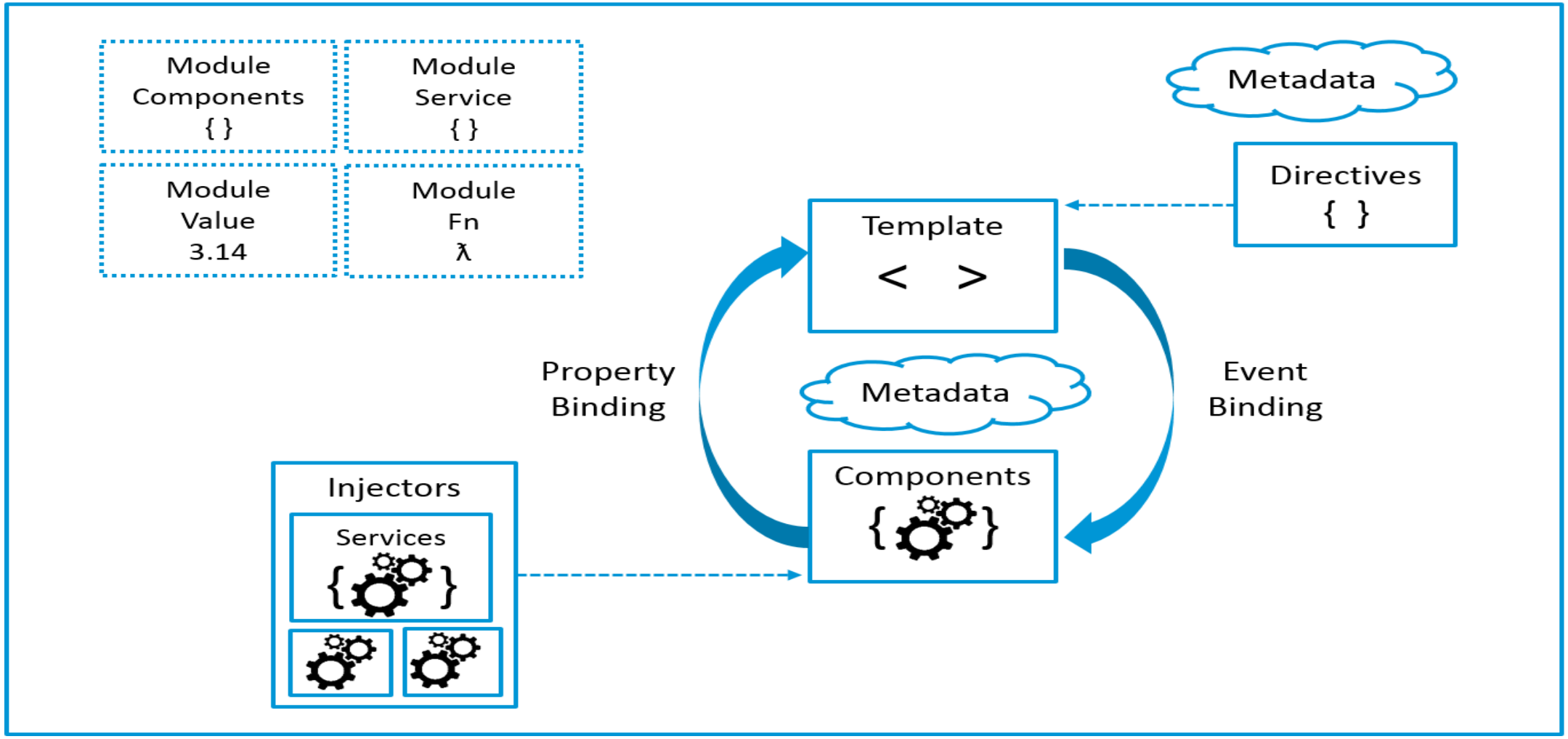
- **HTML markup**: to render that view
- **State**: the data to display on the view
- **Behavior**: the logic behind that view. For example, what should happen when the user clicks a button.

Building Blocks of Angular

- The main building blocks of Angular are:
  - Modules
  - Components
  - Templates
  - Metadata
  - Data binding
  - Directives
  - Services
  - Dependency injection

Building Blocks of Angular

edureka!



Modules

- Angular apps are modular and to maintain modularity, we have *Angular modules* or you can say *NgModules*.
- Every Angular app contains at least one Angular module, i.e. the root module. Generally, it is named as*AppModule*.
- The *root module* can be the only module in a small application. While most of the apps have multiple modules.

- A module is a cohesive block of code with a related set of capabilities which have a specific application domain or a workflow. Any angular module is a class with @NgModule decorator.

Components

- Major part of the development with Angular 4 is done in the components.
- Components are basically classes that interact with the .html file of the component, which gets displayed on the browser.
- A *component* controls one or more section on the screen called a *view*. For example, if you are building a movie list application, you can have components like App Component (*the bootstrapped component)*, Movielist Component, Movie Description Component, etc.

Components

- Each component in an Angular project is physically implemented using four files:
  - **A CSS file**: where we define all the styles for that component. These styles will only be scoped to this component and will not leak to the outside.
  - **An HTML file**: contains the markup to render in the DOM.
  - **A spec file**: includes the unit tests.
  - **A TypeScript file**: where we define the state (the data to display) and behavior (logic) of our component.

```
ng g component mynewcomponent
```

Components

```
import { Component } from '@angular/core';

@Component({
selector:'app-root',
templateUrl:'./app.component.html',
styleUrls: ['./app.component.css']
})

export class AppComponent{
title = My Angular Application!';
}
```

Templates

- In Angular, your views are defined within HTML templates.
- Templates are defined within the **@Component** decorator.
- Templates can be defined as inline HTML templates as well as external templates within HTML files.

Defining Inline HTML Templates

- Change the **templateUrl** property to **template.**

- Declare a single line of HTML.

Interpolation

- Interpolation is just a fancy word for displaying data in the template.
- This data is usually defined within the component class.
- If a component property consists of an object, you can reference a specific object property

```
<h1> {{title}} </h1>
```

Directives

- **Directives** in Angular is a **js** class, which is declared as **@directive**.
- Three types
  - Component Directives : These form the main class having details of how the component should be processed, instantiated and used at runtime.
  - Structural Directives : A structure directive basically deals with manipulating the dom elements. Structural directives have a * sign before the directive. For example, ***ngIf** and ***ngFor**.
  - Attribute Directives :Attribute directives deal with changing the look and behavior of the dom element.

Custom Directives

- Command to create custom directive

```
ng g directive nameofthedirective
```

Pipes

- The | character is used to transform data.

- It takes integers, strings, arrays, and date as input separated with | to be converted in the format as required and display the same in the browser.

Pipes

- Angular 4 provides some built-in pipes. The pipes are listed below –
  - Lowercasepipe
  - Uppercasepipe
  - Datepipe
  - Currencypipe
  - Jsonpipe
  - Percentpipe
  - Decimalpipe
  - Slicepipe

Routing

- Routing basically means navigating between pages.

Forms

- Accept input from user

- Two types:
  - Template driven : most of the work is done in the template
  - Model driven : most of the work is done in the component class

Template Driven Forms

- Add form module in app.module.ts :

  import { FormsModule } from '@angular/forms';

- In template driven forms, the model form controls are created by adding the **ngModel** directive and the **name** attribute.

- Thus, wherever need to access data from forms, add **ngModel** to that tag.

Model Driven Forms

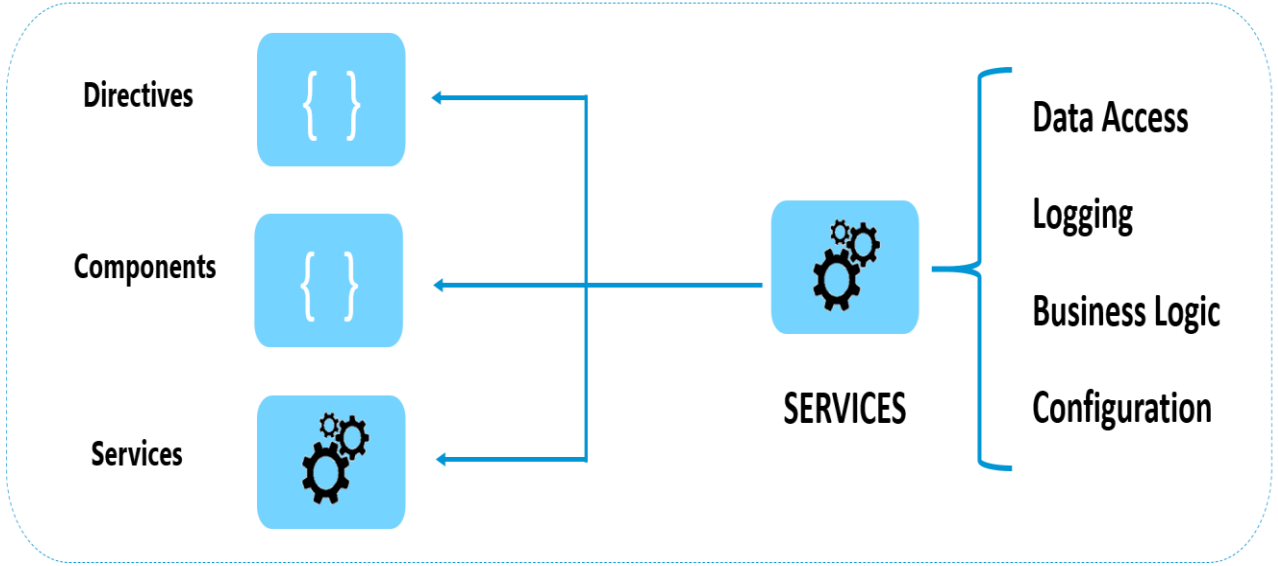
- In the model driven form, the **ReactiveFormsModule** from **@angular/forms** needs to be imported in **app.module.ts** and use the same in the imports array.

- In **component.component.ts**,

  import { FormGroup, FormControl } from '@angular/forms'.

Services

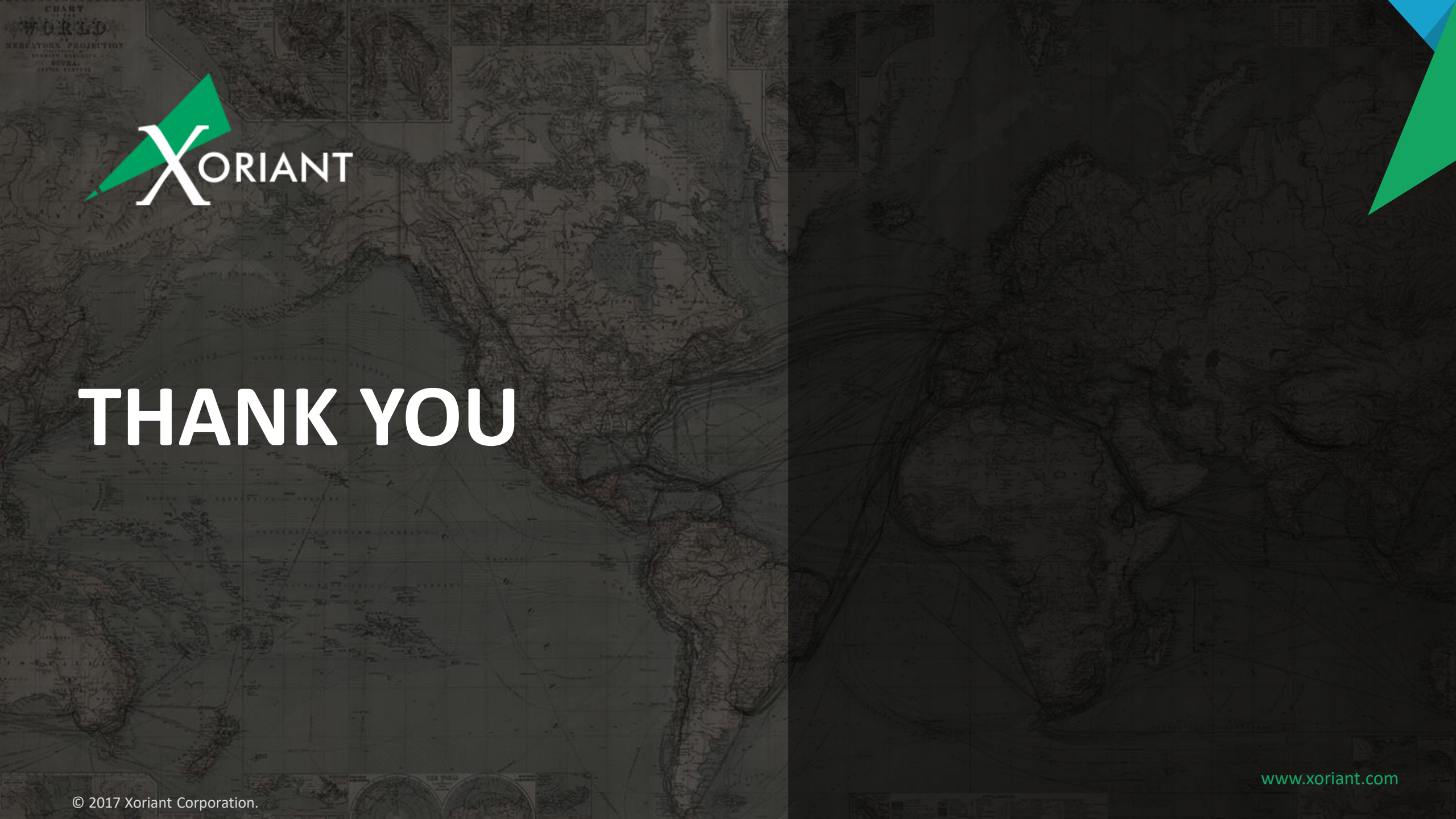
- If some code needs to be used everywhere on the page. It can be for data connection that needs to be shared across components, etc. Services help us achieve that.
- With services, we can access methods and properties across other components in the entire project.



Services

- The command to create a service

```
ng g service servicename
```



THANK YOU

www.xoriant.com