DOWNLOAD PDF ANGULAR 4 EXAMPLE

Chapter 1 : Angular Examples

In this chapter, we will discuss a few examples related to Angular 4. To begin with, we have created an example which shows a login form with input as username and password. Upon entering the correct values, it will enter inside and show another form wherein, you can enter the customer details. In.

Angular Compiler CLI 4. For any request it returns the instance of Observable. Http has following methods. Performs any type of HTTP request. We will perform 1. READ operation using Http. Instantiate Http Claas Http is injectable and can be instantiated using constructor. We will perform dependency injection of Http using constructor. Interaction with HTTP should be done within a service class to separate this code from component and hence that will make code easy to understand and help in bug fixing. Find a sample code snippet to instantiate Http using constructor within a service. This is of any type object that will be passed to REST web service server. In our example we will create an Angular class as Article and pass its instance to body parameter. Using RequestOptions we pass request parameter, request headers etc. Observable is a representation of any set of values over any amount of time. Code Find the code to create the article. Here we will use Angular Http. After successful operation we are returning status code as an instance of Observable. Headers is the angular class that is used to configure request headers. Web service URL to read article. It is used to pass request parameter, headers etc. Code Find the Angular code using Http. This is used to pass request parameter, request headers etc. Code Find the angular code that is using Http. The article will be updated for the given article id. Web service URL to delete article. Code Find the code using Http. All angular Http methods return instance of Observable. Find some of its operators. It applies a function to each value emitted by source Observable and returns finally an instance of Observable. It is called when an error is occurred. To fetch data from instance of Observable we need to subscribe it using RxJS subscribe operator. The actual hit to server goes only when we call subscribe or use async pipe on instance of Observable. In our example we will use subscribe operator to fetch data from Observable. Find the sample code. It is very easy to install and use it with few steps. Open the command prompt and run the following command. In our demo we will perform operations on articles with three fields that are id, title and category. Now we need to run the json-server. Go to the directory where db. In this way, we are ready to use json-server with following URLs. Fetches article by id. Update article by id. Delete article by id. Find the link for json-server reference. Download source code using download link given below on this page. Use downloaded src in your angular CLI application. To install angular CLI, find the link. Run ng serve using command prompt. Now access the URL http: If we are using Angular version 4.

Chapter 2 : Angular 5 CRUD Tutorial Example From Scratch

The article provides a short angular 4 crud example and explains in concise manner. Github code for full app is also included.

Doing research in distributed database systems and caching. This post was originally written to describe the new concepts of Angular 2. It is now updated to Angular 4. Note that all described concepts in this post work in both Angular 2 and 4. Angular is one of the most talked-about JavaScript frameworks. In this transcript of a talk held at the HH. JS meetup in Hamburg we give our best to explain the major concepts of Angular using a real application you can play with: The code is hosted on Plunkr , an online code editor similar to CodePen with support for Angular, TypeScript and folder structures. A framework to build client-side applications Code can be written in Typescript, ES6, Dart or JavaScript without transpiling Has an expressive template language Powerful data binding Available as release 2. Fixes many performance pitfalls of Angular 1 Modern design using ES6 features, e. They are comprised of the template as well as controller methods and properties. Of course, the heart of every MVC framework is its templating, i. Angular supports three types of data binding: The control flow of when and how often data is output is controlled via structural directives. Structuring application logic The main part of the business logic usually resides in services. They encapsulate functionality that is used across many components. To access simple formatting and transformation routines from the templates, pipes are used, e. Routers are used to control the flow through the application with meaningful URLs and a history. Routing is either triggered programmatically or by using routerLink instead of hrefs. Components can subscribe to route changes to handle transitions more efficiently than redrawing the whole app. Directives are a clean and easy way to extend HTML with new elements or attributes. Model data can be the input of a directive. And the directive can also output data, e. Two-way binding with directives is just a combination of an input and an output. Note, that the output property must have the same name as the input property with an appended change to work properly. Get the sources from GitHub. On the left you have all the project files, i. On the right you can directly play with the code and see the results.

Chapter 3 : Angular 4 ngFor Example

This page will walk through Angular 4 CRUD example using Http class. We will perform create, read, update and delete operation here using Http www.nxgvision.com performs HTTP request using XMLHttpRequest as default backend.

Let us move ahead and try to understand the evolution of Angular. So, let us first understand Javascript. JavaScript runs on the client side of the web, which can be used to design or program how the web pages behave on the occurrence of an event. Typically, JavaScript is used for interface interactions, slideshows and other interactive components. JavaScript evolved quickly and has also been used for server-side programming like in Node. JavaScript deals with the dynamic content, which is an important aspect of web development. Dynamic content refers to constantly changing content and it adapts to specific users. For example, JavaScript can be used to determine whether or not to render the mobile version of the website by checking the device, which is accessing the website. Since jQuery has no real structure, the developer has full freedom to build projects as they see fit. For these situations, a framework like Angular can be a big help. Angular Tutorial â€" Angular vs jQuery Angular is a client-side JavaScript framework that was specifically designed to help developers build SPAs Single Page Applications in accordance with best practices for web development. So, you must be wondering what is SPA? The most notable difference between a regular website and SPA is the reduced amount of page refreshes. SPAs have a heavier usage of AJAX- a way to communicate with back-end servers without doing a full page refresh to get data loaded into our application. As a result, the process of rendering pages happens mostly on the client-side. Angular Tutorial â€" Single Page Application For example, if you go through Gmail, you will notice that while opening mail from the inbox will only fetch the email and display it in place of the e-mail list. The rest of the components like sidebar, navigation bar etc. So, this reduces the overhead loading of the website. Moving ahead in Angular Tutorial, we will look through the features of angular and understand how to work with Angular. Angular Tutorial Angular is a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS. But let me tell you that Angular is completely different from AngularJS. Let us understand the differences between Angular and AngularJS. The main building blocks for Angular are modules, components, templates, metadata, data binding, directives, services and dependency injection. We will be looking at it in a while. Angular was a complete rewrite of AngularJS. Thus, Angular first handles mobile development. Modularity â€" Angular follows modularity. Similar functionalities are kept together in same modules. Angular also includes the benefits of ES6:

Chapter 4 : Angular Tutorial for Beginners | Getting Started with Angular 4 | Edureka

Angular 2 & 4 by Example. UPDATE: This post was originally written to describe the new concepts of Angular 2. It is now updated to Angular 4. Note that all described concepts in this post work in both Angular 2 and 4. Angular is one of the most talked-about JavaScript frameworks.

In this tutorial, we will explore how to develop an Angular 4 application using TypeScript. In this tutorial, we will build an Angular 4 application using TypeScript. AngularJS has had a few major releases of its own with v1. However, Angular v2 to v4 is a very different story. It is a rather progressive enhancement. A majority of changes are non-breaking. What happened to Angular v3? Considering this approach, it was important to have a consistent version numbering among various packages. Hence, the Angular team skipped a major version 3. Doing so will help avoid confusions with certain parts of the framework on version 4, while the others on version 3. Angular 4 Enhancements Consider the following enhancements in Angular v4, - This release has considerable improvements in bundle size. This article and the code sample demonstrates an Angular 4 application using TypeScript and Bootstrap 4. Please note that Bootstrap 4 is an alpha release at the time of writing this article. This article assumes you are familiar with developing applications using Angular. If you are new to Angular, check some of our Angular tutorials. It not only saves time, but also makes it easy to maintain the code base during the course of the project, with features to add additional components, services, routing etc. Refer to the appendix at the end of this article for an introduction to Angular CLI. Get Angular CLI at https: You may optionally use --routing parameter to add routing to the Angular project. However, if you prefer to configure and install Angular and Webpack manually, refer to the Get started with Angular 4 manual way section in the appendix. The latest version of Angular CLI v1. If you are using an older version, upgrade to Angular CLI. Refer to figure 1. Scaffolding makes it possible to add new components, routes or services etc. Version details for various libraries with ng -v command Upgrade from Angular v2 to v4 However, to upgrade an existing project to Angular 4, run following npm install commands which upgrades Angular and TypeScript on a Mac or Linux machine. At the time of writing, this article uses the latest stable version for libraries which is Angular v4. Optionally, run the following command to install the latest pre-release package. Modify styles configuration to add Bootstrap CSS styles": It has three views. More details described later in the article. A home page that shows dinosaur list. The list is responsive using Bootstrap 4 Alpha. Cards and the content inside align according to screen size. Randomly picks and shows a dinosaur card. By design, the response is delayed by four seconds. This is to simulate a loading message while an observable is asynchronously retrieving data. With the new features in TypeScript 2. Angular 4 TypeScript Snippets is one such extension. See figure 3 for details on Angular artifacts that we can create with the extension. Imagine we chose to create a component. Tab through to provide component selector, template file name and component class name etc. The Component skeleton file is ready to use. This is an alternative to using Angular CLI for creating a new component. Figure 3 Angular 4 TypeScript snippet extension options Angular 4 New Features and Angular v2 vs 4 Differences The following sections elaborate Angular 4 features and certain differences with Angular 2. Template changes to ng-template If you are upgrading from Angular 2. The following code will result in a warning. It restricts assigning null or undefined only to variables that are defined with that type. Consider the following example, let num1: We can explicitly declare num1 to be of type null to allow assigning to it: Angular made relevant changes to support the feature in Angular 4. However, a bug in TypeScript is keeping the feature at bay. Keep an eye on these Angular and TypeScript issues for updates â€" 1. With Angular 4, support for else has been added. Consider the following template code: It has three conditional blocks. A dinosaur could be: The ngIf checks dino object weight property: When true, show the large template 3. When both the above conditions fail, show the extraLarge template. Specify else template to show while async observable is not ready with data. The directive also supports creating a local variable. Notice a local variable dino let dino to refer to the async object. Refer to code sample for complete code. Refer to following code snippet, this. Access the tool at https: It was part of the bundle even if the application never used animations. It helps reduce production bundle size. Reference the module in

imports array of NgModule Angular future release schedule Refer to the major release schedule in Table 1. The details are quoted from this link bit. Please keep a tab on the page for any changes to the schedule, considering it is still tentative. Tentative Schedule after March Bootstrap 4 in Angular Bootstrap 4 is a rewrite of its previous version v3. One of the core features of Bootstrap is the grid system, which helps develop responsive layout. Bootstrap 4 has enhanced breakpoints with better support for handheld mobile devices. It now supports five breakpoints as opposed to four in Bootstrap 3. Refer to following details,.

Chapter 5 : Best 10+ Angular 4 Loading Spinner Examples [Angular 2/4 Compatible]

This page will walk through Angular 4 ngFor example. Angular provides NgForOf directive. It instantiates a template for every element of given iterator. NgForOf has different local variables that can be used in iteration.

Download source - 9. I was surprised how different it was from the Angular 1. The most difficult aspect about Angular 4 is that the whole application architecture has changed. Even though I thought I was pretty good with Angular 1. The focus of this tutorial is about how to setup a basic Angular 4 application for page navigation. I would also like to discuss some of the gotchas I have noticed. Here are the summary of steps: Install the Latest Node JS. Get the Hello World sample app. Run the Sample App. It was probably nine months old. After some struggle, I just gave up and installed newest version and everything just worked. JS you are using. Upgrade if you must and see if it solves your problem. The download page on nodejs. I think it was a wonderful tutorial. Once you completed all the tutorial, you will no longer be a novice and can sufficiently work with Angular 4. Anyways, I got sidetracked. For me, the easiest way to get started with Angular 4 is to get a working Hello World sample app. Do some customization of the application architecture, and I get an Angular application of my own. Official Angular 4 tutorial provided a Hello World sample project, which is the one I used. The link to the sample app can be found: At this page, on the top you can see the link called "download example", here is the link: Once you download it, unzip it and put the unzipped source code in a meaningful folder, preferrably a folder with a short name like C: Sometimes, this can be problematic if the base directory has a long file name. First you need to unzip the sample application. Go to the unzipped file directory, in there, should be a src directory. Go to this src directory, and run the following command: It would take some time get all the packages. I would wait until all packages are downloaded and installed then go to the next step. In case you are wondering where the packages are defined, go to the directory above src, and you will find a file called package. You can open this in a text editor, and in it, you will find two sections, 1 dependencies, and 2 devDependencies. These are the places where dependencies can be modified. I used bootstrap for my sample application. So I added two new dependencies bootstrap version 3. And it will just add the newly specified packages. I believe this has something to do with the relative path ".. That is all you need to worry about relative paths, which will be described in the next section. Running the Sample Project To run the sample project, here is all you need to do: Even though I made the change to the package. If you can see Hello World on the browser, then you are ready for the second part of this tutorial, modified to have navigation, and some basic Angular 4 functionality for user interaction. To make this sample app a little bit more meaningful, I also included a little UI interaction on one of the pages. I will explain how that works as well. When I looked into this, I found page navigation in Angular 4 as an interesting challenge. And it is very important to learn. So I decided to use this as an opportunity to learn the basic of it. Angular 4 is used to create single page application. But it does not mean the applicaion is composed with just one page. This would make the application very clogged. A better way is to split the functionality of an application into multiple functionality, and each functionality is served by its own pages. This feature is supported since Angular 1. I thought it would be simple to grasp. It turned out to be quite complicated. I started out by reading the tutorial on angular. Then I read the source code of a project from angular. It turned out to be complicated. With that source code and some other resources I have forgotten where I found them sorry , I was able to get it to work. This file will be compiled into main. The content of the main. You can see the reference in above code snippet -- ". It tells where the file is, under the sub directory app, there is the file app. The content of this file is like this: These are all the modules and components I have made. There is just one class in this file, called "AppModule". The class has a decoration NgModule, something similar to attribute in. NET framework and annotation in Java. The decoration does three different things: It marks the class AppComponent as a bootstrapping class -- the start up class. It marks three classes as controllers. And it imports two modules. One of the module is the routing modulde. It is defined in a file called app. The content of this file looks like this: The first two are from the Angular library. They are needed because 1 the AppComponent class uses the Component decorator; 2 the template which AppComponent is using, references the routerLink and

router-outlet which is from the Routes class. The second half of the file is the definition of the AppComponent. This class is like the two classes from previous sections, it has no properties or methods. And it has one decorator called Component. For this class, the decorator Component has two properties, one is called the selector. The other is called the templateUrl. The selector property defines on the index page what element will be replaced with a new UI template. The templateUrl defines what page template will be used. Essentially, AppComponent is the controller, and the templateUrl references the view which the controller can render. The selector is assigned with value "my-app". So where is this my-app selector used? You can find it in the index. And if you remembered from previous sections, one of the modules loaded is the app. That is how AppComponent is getting bootstrapped as a start up class. Now, take a look at the template file for AppCompoment: The top part is a navigation menu. There is a lot of html code used to create this navigation menu. The real good part about it is the two anchors: The mark up in the template html file does not make the navigation work. What makes it work is my AppRoutingModule. Which will be discussed next. The Routing Module The routing module is defined in the file app-routing. The class is called AppRoutingModule. Here is something I learned when I finally got this class integrated with the rest of the project. The file name matters.

Chapter 6 : Angular 4 application development with Bootstrap 4 and TypeScript | DotNetCurry

*Angular 4 - Code Sample. Let us explore the Angular and Bootstrap features with the code sample provided with this article. It has three views. a. Basic: For demonstrating ng-template feature using *ngIf directive. More details described later in the article. b. Dinosaur List: A home page that shows dinosaur list. The list is responsive using Bootstrap 4 (Alpha).*

Angular Compiler CLI 4. Index of current item. True for an even index. True for an odd index. True for first item. True for last item. It is useful to alias when expression is more complex than a property access, for example using Async pipe such as obsPersons async. Suppose we have an instance of Observable as following. Find the code snippet. If an item is added then new instance of template is added in DOM. If an item is removed then its respective template is removed from DOM. If items are reordered, respective templates are reordered in DOM. If no change for an item, its respective template in DOM will be unchanged. Angular uses object identity to maintain the entire above task of change propagation. If we want that Angular should not use object identity for change propagation but use user provided identity, then we can achieve it using trackBy. It assigns a function that accepts index and item as function arguments. So NgForOf will re-draw the template even as there is no change in any object value. That impacts the performance and hence in this case we should use trackBy or ngForTrackBy function to change the default tracking id. Complete Example Project Structure angular-demo --src --app --person. Download source code using download link given below on this page. Use downloaded src in your angular CLI application. To install angular CLI, find the link. Run ng serve using command prompt. Now access the URL http:

Chapter 7 : Angular 4 Firebase Tutorial: Make a Simple Angular 4 App

Learn Angular 4 by building 7 interesting applications. In this course we cover many important and advanced Angular features that you can use right away in your own applications. For each example app, we will download a simple boilerplate and then jump right in and start coding our apps.

Next Page In this chapter, we will discuss a few examples related to Angular 4. To begin with, we have created an example which shows a login form with input as username and password. Upon entering the correct values, it will enter inside and show another form wherein, you can enter the customer details. In addition, we have created four components - header, footer, userlogin and mainpage. Here, the pages are login-form and once it is successful it will redirect to the mainpage, i. To get the login-form first and later get the mainpage. Only the default details are present in it. To start with, we will first take the header component. For the new component, four files are created header. This makes the header. For the header, we will draw a horizontal line. A logo or any other detail can be added to make the header look more creative. Let us now consider creating a footer component. For the footer component, footer. For the footer, we will just draw a horizontal line as shown in the. Let us now see how the userlogin component works. The following files for userlogin component created are userlogin. This is a model driven form approach and the details of the same are explained in Chapter 14 - Forms. We consider the username and password mandatory, hence the validation for the same is added in the ts. Upon clicking the submit button, the control is passed to the onClickSubmit, which is defined in the ts file. We need the username to be more than six characters and the field is mandatory. The same condition applies to password too. Upon clicking submit, we can check if the username is systemadmin and the password is admin If yes, a dialog box appears that shows Login Successful and the router navigates to the app-mainpage, which is the selector of the mainpage component. There is css added for the form in userlogin. Let us now discuss how the mainpage component works. The files created for mainpage component are mainpage. The validation of the same is done with the ngOnInit function. Upon clicking submit, the control comes to the function onClickSubmit. Here, the table which is used to display the entered details is made visible. The customerdata is converted from json to array so that we can use the same in ngFor on the table, which is done in the. The display of the userlogin and the customer details is as shown below. This is the page with login form and header and footer. Once you enter the details, the display is as shown below Upon clicking submit, a dialog box appears which shows Login Successful.

Chapter 8 : Angular 4 Components Tutorial

For example, we can use Bootstrap 4 cards to render the product inside a card. But for now, we don't want to waste our time and make our application pretty. Our focus is on understanding Angular.

Chapter 9 : Angular 4 CRUD example - theJavaGeek

We will be talking about Angular instead of Angular 2 or Angular 4 as from Angular 2 onwards Angular community has decided to refer any version of Angular as just Angular. In this Angular Tutorial blog, we will be covering.