About viralpatel.net

Home

Join Us

Advertise

Search

#### VIRALPATEL.NET

Android	
Java	
JavaEE	
Spring	
Spring 3 MVC	
Spring Roo	
Frameworks	
Hibernate	
Struts	
Struts 2	
JavaServer Faces	
Play Framework	
FreeMarker Template	
Database	
MySQL	
Oracle	
JavaScript	
AngularJS	

Smarty Web 2.0

Fun

CSS PHP

AJAX

JQuery Dojo **Web** HTML5

News

General

# AngularJS Routing and Views Tutorial with Example

BY VIRAL PATEL · OCTOBER 15, 2013



Welcome to the next tutorial of the ongoing series of tutorials on AngularJS. In previous tutorial we saw AngularJS Controllers and also created a Hello World example using Angular.

In this article we will go
through the next useful
feature of AngularJS called
Routing. Also we will see how
we can divide a single page
application in multiple views.
As we add more and more
logic to an app, it grows and
soon become difficult to
manage. Dividing it in Views
and using Routing to load
different part of app helps in
logically dividing the app and

**FOLLOW:** 





**Advertise Here** 

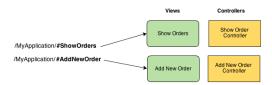
☑ Get our	Articles by	Email. En	ter your	email.

Subscribe

Your Email

making it more manageable.

Routing helps you in dividing your application in logical views and bind different views to Controllers.



In above diagram we create two Route url /ShowOrders and /AddNewOrder. Each points to a specific view and is managed by a controller. Don't panic if it does not make any sense. Soon we will see some code and it all will be clear.

## 1. Introductionto\$routeProvider

The magic of Routing is taken care by a service provider that Angular provides out of the box called \$routeProvider. An Angular service is a singleton object created by a service factory. These service factories are functions which, in turn, are created by a service provider. The service providers are constructor functions.

#### **RECENT POSTS**

How to secure WordPress Admin with HTTPS – Self-signed certificate

WordPress – Allow Contributors to Add / Upload Media

Excel Macro: Evaluating Formulas Dynamically

Getting Started With Yeoman (Introduction to Yeoman)

Navigating Spring Security from thick Client to REST Webservice

**SPONSORS** 

When instantiated they must contain a property called \$get, which holds the service factory function.

When we use AngularJS's dependency injection and inject a service object in our Controller, Angular uses \$injector to find corresponding service injector. Once it get a hold on service injector, it uses \$get method of it to get an instance of service object. Sometime the service provider needs certain info in order to instantiate service object.

Application routes in Angular are declared via the

\$routeProvider, which is the provider of the \$route service. This service makes it easy to wire together controllers, view templates, and the current URL location in the browser. Using this feature we can implement deep linking, which lets us utilize the browser's history (back and forward navigation) and bookmarks.

Syntax to add Routing
Below is the syntax to add
routing and views information
to an angular application. We







Advertise Here

defined an angular app
"sampleApp" using
angular.module method. Once
we have our app, we can use
config() method to
configure \$routeProvider.
\$routeProvider provides
method .when() and
.otherwise() which we can
use to define the routing for
our app.

```
var sampleApp =
angular.module('phonecatApp
', []);
sampleApp
.config(['$routeProvider',
  function($routeProvider)
    $routeProvider.
      when('/addOrder', {
        templateUrl:
'templates/add-order.html',
        controller:
'AddOrderController'
     }).
      when('/showOrders', {
        templateUrl:
'templates/show-
orders.html',
        controller:
'ShowOrdersController'
     }).
      otherwise({
       redirectTo:
'/addOrder'
     });
  }]);
```

In above code we defined two urls /addOrder and

/showOrders and mapped

them with views

It then invokes

templates/add-order.html

and templates/show-

orders.html respectively.

When we open
http://app/#addOrder url in
browser, Angular
automatically matches it with
the route we configures and
load add-order.html template.

AddOrderController where we can add logic for our view.

## 1.1. Hello WorldAngularJS + Routing

Let us go through an example in AngularJS and use Routing to load different templates at runtime.

Below sample 1.html file is the main html file. It includes
AngularJS library and define structure for our app. We have two links: Add New Order and Show Order. Each link loads template in below section.
sample 1.html

```
Routing example</title>
 </head>
 <body ng-app="sampleApp">
   <div class="container">
              <div
class="row">
               <div
class="col-md-3">
                       <ul
class="nav">
<a href="#AddNewOrder">
Add New Order </a>
<a href="#ShowOrders">
Show Order </a>
</div>
               <div
class="col-md-9">
<div ng-view></div>
               </div>
               </div>
   </div>
       <script
src="http://ajax.googleapis
.com/ajax/libs/angularjs/1.
0.7/angular.min.js">
</script>
       <script
src="app.js"></script>
 </body>
</html>
```

#### ng-view

One thing worth noting is the ng-view directive. In our angular app, we need to define ng-app directive once. This becomes the placeholder for views. Each view referred by

the route is loaded in this section of document.

You can define ng-view in main html file in one of the below way.

```
<div ng-view></div>
..
<ng-view></ng-view>
..
<div class="ng-view"></div>
```

### 1.2. Add Routing in AngularJS

In above sample 1.html file we included a javascript file app.js which holds the application logic. Below is the content of app.js.

app.js

```
//Define an angular module
for our app
var sampleApp =
angular.module('sampleApp',
[]);

//Define Routing for app
//Uri /AddNewOrder ->
template add_order.html and
Controller
AddOrderController
//Uri /ShowOrders ->
template show_orders.html
and Controller
AddOrderController
sampleApp.config(['$routePr
```

```
ovider',
  function($routeProvider)
    $routeProvider.
     when ('/AddNewOrder',
{
templateUrl:
'templates/add order.html',
                controller:
'AddOrderController'
       }).
      when('/ShowOrders', {
templateUrl:
'templates/show orders.html
                controller:
'ShowOrdersController'
      }).
      otherwise({
                redirectTo:
'/AddNewOrder'
     });
}]);
sampleApp.controller('AddOr
derController',
function($scope) {
        $scope.message =
'This is Add new order
screen';
});
sampleApp.controller('ShowO
rdersController',
function($scope) {
        $scope.message =
'This is Show orders
screen';
});
```

We first use .config() method

to define \$routeProvider
configuration. Also in the same
file we define two controllers
AddOrderController and
ShowOrdersController. In a
real world application these
controllers will hold a lot of
logic but for example sake we
just define a message property
on \$scope which later we use
to display on view.

Notice how we used otherwise() method to define a default route. In case routeProvider does not matche with any url, it redirects to default route.

```
otherwise ({
          redirectTo:
          '/AddNewOrder'
});
```

### 1.3. Add HTML template files

Our app.js is ready. We still needs to define two html templates. These are partial templates of our app.

templates/add\_order.html

```
<h2>Add New Order</h2>
{{ message }}
```

add\_order.html template should have an html form for adding new orders. For sake of simplicity we just show a message.

templates/show\_orders.html

```
<h2>Show Orders</h2>
{{ message }}
```

#### 1.4. Online Demo

Click links in below example to load different template based on Uri.

Add New Order

**Show Order** 

Demo link: Plnkr.co

#### 2. How to pass

## Parameters in Route Urls

We saw how to define route in above example. Now let us see how can we define parameters in route urls.

Consider below scenario. We want to display details of different orders. Based on a parameter order\_id we will define order details in view.



In angular while define route we can define parameters using orderId in url. For example:

```
when('/ShowOrder/:orderId',
{
    templateUrl:
   'templates/show_order.html'
,
    controller:
   'ShowOrderController'
});
```

And we can read the parameter in ShowOrderController by using

\$routeParams.orderId.

```
$scope.order_id =
$routeParams.orderId;
```

Let us checkout a sample application. Below sample 2.html is main html page which describe the structure. It contains order information data in tabular format. Each order has a "show details" link. This link uses parametrize route url to load order detail screen.

#### sample2.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <title>AngularJS
Routing example</title>
 </head>
 <body ng-app="sampleApp">
   <div class="container">
      <div class="row">
      <div class="col-md-
9">
             <table
class="table table-
striped">
<thead>
#Order No.
Details
```

```
</thead>
11234
15" Samsung Laptop
<a
href="#ShowOrder/1234">show
details</a>
25412
2TB Seagate Hard
drive
<a
href="#ShowOrder/5412">show
details</a>
39874
D-link router
<a
href="#ShowOrder/9874">show
details</a>
<div ng-
view></div>
     </div>
     </div>
  </div>
<script
src="http://ajax.googleapis
```

```
.com/ajax/libs/angularjs/1.
0.7/angular.min.js">
</script>
<script src="app.js">
</script>

</body>
</html>
```

#### app.js

```
var sampleApp =
angular.module('sampleApp',
[]);
sampleApp.config(['$routePr
ovider',
  function($routeProvider)
    $routeProvider.
when('/ShowOrder/:orderId',
        templateUrl:
'templates/show order.html'
        controller:
'ShowOrderController'
      });
}]);
sampleApp.controller('ShowO
rderController',
function ($scope,
$routeParams) {
        $scope.order id =
$routeParams.orderId;
});
```

Note: Don't forget to inject \$routeParam parameter in controller. Otherwise you wont be able to use it.

templates/show\_order.html

```
<h2>Order #{{order_id}}
</h2>
Here are the details for
order <b>#{{order_id}}</b>.
```

#### 2.1 Online Demo

Click the show details links on different orders.

#	Order No.	Details	
1	1234	15" Samsung Laptop	show details
2	5412	2TB Seagate Hard drive	show details
3	9874	D-link router	show

Demo link: Plnkr.co

## 3. How to Load local views(Views within

#### script tag)

It is not always that you want to load view templates from different files. Sometimes the view templates are small enough that you might want them ship with main html instead of keeping them in separate html files.

### 3.1 ng-template directive

You can use ng-template to define small templates in your html file. For example:

Here we defined a template
"add\_order.html" inside
<script> tag. Angular will
automatically load this
template in ng-view whenever
add\_order.html is referred in
route.

Let us quickly go through a sample app where we use

local view definitions.

sample3.html defines
structure of app. It is similar to
the first example that we saw
(sample1.html) with a bit of
change. We defined two views
add\_order.html and
show\_orders.html within
sample1.html as <script> tag.
sample3.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <title>AngularJS
Routing example</title>
   link
href="http://netdna.bootstr
apcdn.com/bootstrap/3.0.0/c
ss/bootstrap.min.css"
rel="stylesheet">
 </head>
 <body ng-app="sampleApp">
   <div class="container">
               <div
class="row">
               <div
class="col-md-3">
class="nav">
<a href="#AddNewOrder">
Add New Order </a>
<a href="#ShowOrders">
Show Order </a>
</div>
               <div
class="col-md-9">
```

```
<div ng-view></div>
                </div>
                </div>
    </div>
        <script
type="text/ng-template"
id="add order.html">
                <h2> Add
Order </h2>
                {{message}}
        </script>
        <script
type="text/ng-template"
id="show orders.html">
                <h2> Show
Orders </h2>
                {{message}}
        </script>
        <script
src="http://ajax.googleapis
.com/ajax/libs/angularjs/1.
0.7/angular.min.js">
</script>
        <script
src="app.js"></script>
  </body>
</html>
```

The app.js is similar to first sample. There is no change in app.js. Refer to first demo example if you wanna check app.js.

#### 3.2 Online Demo

Click the show details links on

different orders.

Add New Order

**Show Order** 

Demo link: Plnkr.co

## 4. Add Custom Data to RouteProvider

The \$routeProvider provides methods when() and otherwise() which we used to define url routes. Sometime we might want to pass custom data based on certain route. For example you might use same Controller in different routes and use some custom data. For example:

In above code we defined a new property "foodata" while defining route. This property is then accessed in controller using

\$route.current.foodata.

#### References

- AngularJS documentation on \$routeProvider
- AngularJS documentation on View & Routing
- Mapping AngularJS Routes
   Onto URL Parameters And
   Client-Side Events

#### That's All Folks

Hope this tutorial helped you in understanding Routing and

views concept of AngularJS. In case you missed previous tutorials on AngularJS, do checkout those.

**Update**: AngularJS Service Tutorial published.

#### **Related Articles**

- AngularJS Controller Tutorial with Example
- AngularJS Service / Factory Tutorial with Example
- 3. AngularJS: Introduction and Hello World example
- 4. Eclipse Tip: Add your own Code Template in Eclipse
- Getting Started With Yeoman (Introduction to Yeoman)
- Your first Play! GAE –
   Siena application: Tutorial with Example
- Spring 3 MVC: Tiles Plugin Tutorial with Example in Eclipse

[	Get our Articles via Email. Enter your email address.		
	Your Email		
	Send Me Tutorials		

Tags: angularjs JavaScript



#### YOU MAY ALSO LIKE...

Ang	Ang	Ang
ular	ular	ular
JS	JS:	JS
Con	Intr	Serv
troll	odu	ice /
er	ctio	Fact
Tut	n	ory
orial	and	Tut
with	Hell	orial
Exa	0	with
mpl	Wor	Exa
е	ld	mpl
	exa	е
	mpl	
	е	

#### **101 COMMENTS**

```
Riyas Ebrahim
① 18 June, 2014, 16:47
good article..this helped me
a lot...thanks...keep going
Viral...
Reply
chetan
25 June, 2014, 17:42
I have one problem using
your structure, I am using
your code for the
routeing.like below code,
But I have problem for the
login page. login page has
no
lefttemplate,footertemplate,headertemplate,
How we can set for this
type of the code structure
for login page with out
header/footer/left section.
.provider('myPageCtx',
function () {
var defaultCtx = {
title: ",
//headerUrl:
'partials/default-
header.tmpl.html',
headerUrl:
'partials/default-
header.tmpl.html',
leftUrl: 'partials/default-
leftnav.tmpl.html',
footerUrl: 'partials/default-
footer.tmpl.html',
};
var currentCtx =
angular.copy(defaultCtx);
return {
$get: function ($rootScope)
```

```
$rootScope.$on('$locationChangeStart',
function () {
angular.extend(currentCtx,
defaultCtx);
});
return currentCtx;
}
};
})
.config(function
($routeProvider) {
$routeProvider.when('/login',
templateUrl:
'partials/login.html'
});
$routeProvider.when('/view1',
templateUrl:
'partials/partial1.html'
});
$routeProvider.otherwise({
redirectTo: "
});
});
When i try to add login
page without
header/footer, it is not
showing properly. I
comment out lefturl, it is
not available for all pages.
if i put
lefturl/footerurl/headerurl
in the login config box, and
remove out the
lefturl/headerurl/footerurl
from the provider section,
it is not seen for all the
pages and login pages also.
How to set login page so
we can display without
```

left/header/footer for the login page. And other pages it is seen left/header/footer url to add it(i.e show left/header/footer section there).. \$routeProvider.when('/login', //headerUrl: 'partials/defaultheader.tmpl.html', headerUrl: 'partials/defaultheader.tmpl.html', leftUrl: 'partials/defaultleftnav.tmpl.html', footerUrl: 'partials/defaultfooter.tmpl.html', templateUrl: 'partials/login.html' **})**; Reply

#### Ravichandran Jayaramegowda

② 25 June, 2014, 19:37

Hey Patel,

Thanks for nice tutorial to get started with \$routeProvider.

Thanks,

Ravi

Reply

#### youhammi

**2** 26 June, 2014, 13:53

thanks a lot for this good article

ar crere

Reply

**\_24**\_ **②** 1 July, 2014, 15:48

```
hi sir,
can you help me on my
delete function?
i love your tutorial..
tnx
Reply
_24_ ② 1 July, 2014, 16:13
$scope.delete = function
(models){
for (i in $scope.results){
if ($scope.results[i].id ==
clientId ){
$scope.results.splice(i, 1);
}
}
yes
Reply
Mahesh
① 7 July, 2014, 17:48
var sampleApp =
angular.module('phonecatApp',
['ngRoute']);
I came to know after
googling, ngRoute required
in the above statement for
angular 1.1.6 or later
Thanks for the tutorial Viral
Patel you really rocks
Reply
        Manjit
        ② 12 August, 2014,
        11:50
        This is also required
        Reply
```

#### Ronan

① 11 July, 2014, 18:10

Many thanks for this

article. I read that there is also a UIRouter module that we can use to adress routing in AngularJS.

Reply

#### Shehab

② 16 July, 2014, 15:23
You are wonderful.....your articles helped me alot:)

Reply

#### Paul Fortin

① 17 July, 2014, 7:24

To often articles on new technologies are written for the already initiated and riddled full of that technologies special language which makes that article useless to learn the basic concepts of that technology. I have read all your Angular JS articles and some of the others and I think they are well written and written in a common language that all programmers can understand - this allows us to learn the technology.

Thanks for your great work!

Paul

Reply

#### Viral Patel

**②** 17 July, 2014,

13:12

Thanks a lot Paul.. am really glad you liked these tutorials. Reply

#### Prasobh

21 July, 2014, 16:56

I've gone through your articles about angular js and are really great..what are all the script files that i need to download for implementing all the angular js functionalities?I would like to implement everything locally and so that iam not actually giving reference to the link in the script's source(the link you are mentioned) attribute.Please help me with the script files **Thanks** 

Reply

#### **Taranjeet Singh**

25 July, 2014, 14:55

(Re: 4. Add Custom Data to RouteProvider)

Will it still work if the controller is not common between the two routes? I mean your example show intra-controller communication, can we have inter-controller communication using custom data feature?

Reply

#### **Andrew Chan**

② 28 July, 2014, 21:09 Great article. Very well written and clearly explained.

Thanks alot!

Reply

#### Ravichandran Jayaramegowda

② 31 July, 2014, 17:29

Hi,

It is not working for version "1.1.0" and above. Please suggest.

Thanks,

Ravi

Reply

#### Sims

② 2 August, 2014, 12:05

Thanks alot for ur nice turtorial Viral....

Can u please put tutorials on tagging pictures or videos and we would be able to save that tags or can be deleted when needed.

[on-tagadded="{expression}"] . we can use this but can u explain how it is possible?

• • •

Reply

#### robert

② 31 August, 2014, 13:23

Thanks.

was so helpful;)

Reply

#### Jorge

② 3 September, 2014, 21:29

Don forget to include ngRoute in your module as well as including the route.js in your head. It is

an external module now in AngularJS. Otherwise, this whole thing won't work!

var sampleApp = angular.module('sampleApp', ['ngRoute']);
Reply

#### Sagar Mali

② 24 September, 2014, 16:08 Hi Jorge,

I agree with you, i have implemented this scenario.

Can you please tell

me

> why is it so? > how can i add ngAnimate with ngRoute?

Reply

#### satish prasad

**②** 8 October, 2014,

14:40

Hi Jorge,

letscatchup.net this

is my job portal,

and I want to add

two more page in

my home page

career and blogs.

but my link is not

working when i

click on career or

blog it came like

shadow (black)

page

please guide me

Reply

#### Lakhwinder

28 January, 2015,

8:15

Thanks Jorge, it resolved the issue

at my end also. and thanks Viral for a wonderful article for starters.

Reply

#### Anna

② 9 September, 2014, 20:35 Why would you use routing instead of a directive?

Reply

#### giriswamy

② 11 September, 2014, 15:21

nice

Reply

#### Jayesh Patel

② 23 September, 2014, 10:37

Hi Viral,

Great job...the level of

explanation with Simple

words ... highly

appreciable.

if you can add one keyword

to your code can help

starters of Angular JS.

var sampleApp = angular.module('sampleApp', ['ngRoute']);

"ngRoute" is missing, due

to which when I copy paste

your code it was giving

error.

Thanks for developing my interest on learning new

technology.

Reply

#### sunny

**②** 14 October, 2014, 12:59

You are wonderful.....your articles helped me alot:)

Reply

#### John

Reply

#### Fabian

② 15 November, 2014, 14:22 Awesome tutorial, thank you for taking the time of doing this article!!!!

Reply

#### Vineet

① 16 November, 2014, 13:42 Viral,

Thanks for a great (and thorough) tutorial on this subject. I am pretty new to Javascript and AngularJS and something that I couldn't quite understand was the argument to the .config. According to the angular. Module, config method takes configFn as a parameter, yet your example shows ['\$routeProvider', function ...] as an arg. I was wondering what does the square brackets mean here?

BTW, I am going through your other articles as well on AngularJS. Appreciate them all.

Reply

#### prasanth mohan

① 17 November, 2014, 17:13

thank u sir , very helpful article

Reply

#### Anil

② 18 November, 2014, 12:42 Really a cool tutorial, i got to learn a lot by this tutorial. Am new to angular, if we can some more working examples. Thanks

Reply

#### santosh shimpi

② 10 December, 2014, 15:40 Thanks for the article, i have 1 problem

when running the application, it changes url in browser to 'http://localhost:8080'

Could you please inform on this

Reply

#### Denis

① 18 December, 2014, 16:58

Big thanks for this tutorial, it a lot more clear than oficial one and certainly better than dry documentation.

Reply

#### thomas

② 7 January, 2015, 13:38 Nice tutorial. Thanks a lot.

Reply

#### Aparna

② 30 January, 2015, 11:40

Thanks buddy, it helped a

lot.

Reply

#### Rajesh

**1** 16 February, 2015, 12:07 hi Viral,

i want to load multiple html's along with controller's in to single html or single. Please let me if we have any way.

Thanks in advance.

Reply

#### Jhondoe

© 20 February, 2015, 1:54 Easy, basic and really power for begin point of learn, THANKS!

Reply

#### sourav bhusan

#### venkatraja

② 5 March, 2015, 23:01

Sir,

Reply

Your angularis tutorials are very fine. Why you not cover Directives in this tutorial. This is very useful for us. Thanks for your service.

By, K.T.Venkat Raja

Reply

#### Ajit Kumar

① 13 March, 2015, 19:37

sir,

when i am using the latest angular in this code its not working.. Please help me find a solution

used version

latest version

Reply

#### **Tridip**

● 19 March, 2015, 15:56 which code we need to stored in app.js?

you could provide a source code download link as a result people can download and run the apps in their pc. thanks

Reply

Chris ② 14 May, 2015, 14:21 Hi Viral, I just want to thank you for a great tutorial. This has helped me a great deal.

Reply

shiva ② 17 May, 2015, 13:36
I have play framework
(java) providing services.
Planning to have Angularjs
for UI. Both Angular and
Play do have their own
routes. I want the
Angularjs making calls to
Play services through

Angularis controllers. How do I make these routers work in this fashion? My application is going to Play and gives up saying that route not found.

Any examples are appreciated.

Reply

#### sachin verma

⊙ 3 June, 2015, 16:42 very nice tutorial...as u always did great things....keep helping..:)

Reply

#### Vijay Soni

● 8 June, 2015, 11:58 i would like to thanks for this artical... pls keep writing...

Reply

#### Jyotirmoy Pan

② 15 June, 2015, 17:58

There are several tutorials on angular but you have explained the concepts with much ease an clarity. Great going. You can also provide some plunkr examples, it would be great if we can play with some of the features giving more clarity.

Reply

#### « Older Comments

#### LEAVE A REPLY

Name *	Email *
Website	
Comment	
Post Commer	nt
Current ye@r *	
3.5	

ViralPatel.net © 2015. All Rights Reserved

