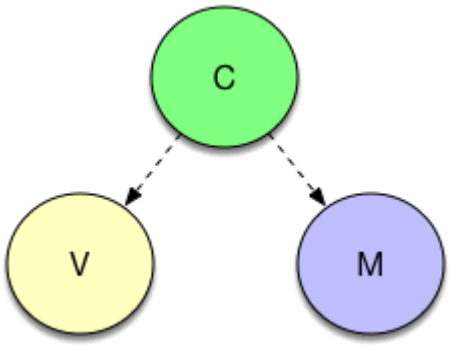
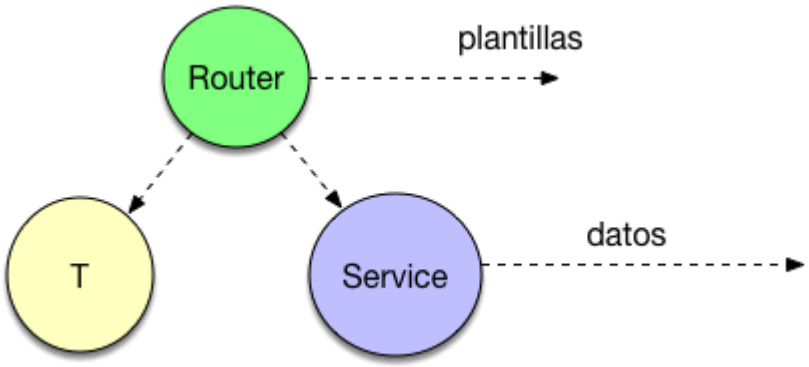
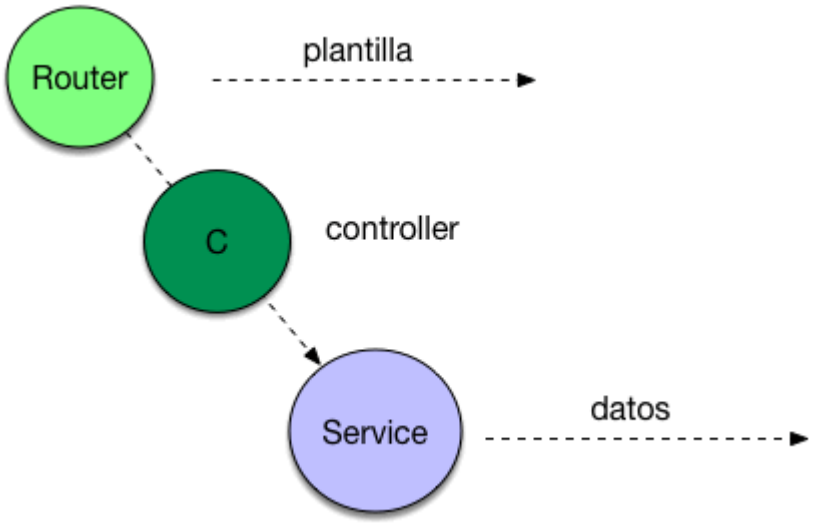
¿Para que sirve Angular resolve? . Hoy en día trabajamos cada día más con arquitecturas SPA (Simple Page Application) . En las cuales la mayor parte de las responsabilidades se ubican en el cliente implementando un modelo MVC.



En Angular este modelo MVC esta compuesto por las plantillas, Router/Controladores y los Servicios. Son estos últimos los que se encargan de solicitar datos al Servidor.

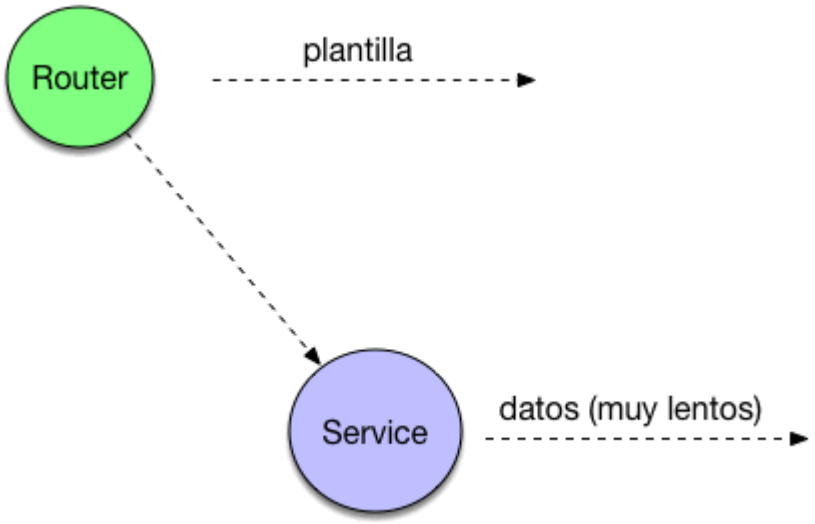


La forma más clásica de trabajar es usando el router y solicitando una nueva ruta. En este caso el controlador que se cargue invocará los servicios adecuados y nos mostrará la información.



Angular Route y sus problemas

Esta forma de trabajar no este libre de problemas ya que cuando solicitamos un cambio de ruta y cargamos un controlador este puede invocar a un servicio que tarde mucho en enviarnos los datos.



Por lo tanto nos encontraremos que el resultado es una página vacía que al cabo de un rato se rellena.

Sin Angular Route

Vamos a construir un ejemplo con Angular para darnos cuenta del problema. En primer lugar vamos a mostrar el código relativo al Router.

```
<html ng-app="miapp">

<head>
 <script type="text/javascript" src="angular.js">
 </script>
 <script type="text/javascript" src="angular-route.js">
 </script>
 <script type="text/javascript" src="servicios/servicios.js">
 </script>
 <script type="text/javascript" src="controladores/controladores.js">
 </script>
 <script type="text/javascript">
 angular.module("miapp", ["ngRoute","controladores",
"servicios"]).config(function($routeProvider) {

 $routeProvider
 .when('/', {
 templateUrl : 'plantillas/sencillo/inicio.html',
 controller : 'controlador'
 })
 .when('/libros', {
 templateUrl : 'plantillas/sencillo/libros.html',
 controller : 'controladorLibros'
 }).otherwise({
```

```
 templateUrl : 'plantillas/sencillo/inicio.html',
 controller : 'controlador'

 })


 });
 </script>
</head>

<body>
<div ng-view></div>
</body>
</html>
```

Como podemos observar solo tenemos dos plantillas la de inicio y la de libros. El código de la plantilla de inicio es :

```
<div>
Bienvenido 1<input type="button" ng-click="verLista()"
value="verlista"/>
</div>
```

Así pues cuando nuestra aplicación arranque mostrará:

El problema viene cuando solicitamos acceder a una lista de Libros. Esta lista de libros está definida en Node.js con el siguiente código:

```
var express=require("express");
var app=express();
app.use(express.static("publica"));

var lista=[{titulo:"el juego de ender",paginas:500},{titulo:"el
padrino",paginas:400}];
app.get("/libros",function(req,res) {

setTimeout(function() {

res.send(lista);
},4000);
});
var servidor=app.listen(3000,function() {
console.log("servidor iniciado");
});
```

El código es sencillo de entender , la url /libros nos devuelve unos datos en JSON con un listado de libros , pero tarda 4 segundos en devolverlos. Algo que en principio no debería

ser un problema. Vamos a ver el código de nuestro servicio que será el encargado de solicitar estos datos:

```
angular.module("servicios",[]).service("servicioLibros",function($http
) {

this.buscarTodos=function() {

return $http.get("libros");

}
});
```

No tiene nada de especial , así pues simplemente nos queda ver el código de los controladores que son los que invocan a este servicio:

```
angular.module("controladores", [ "servicios","ngRoute"])
.controller("controlador", function($scope,$location) {


$scope.verLista=function() {

$location.path("/libros");

}
}).controller("controladorLibros",function($scope,servicioLibros) {
```

```
servicioLibros.buscarTodos().then(function(response) {

$scope.libros=response.data;

})
})
```

En este caso el primer controlador se encarga de cambiar de ruta en el evento verLista y el segundo invoca al servicio y carga los datos. Nos queda por ver la plantilla que muestra la informacion:

```
</pre>
<div>
<table>
<tr>
<th>titulo</th>
<th>paginas</th>
</tr>
<tr ng-repeat="libro in libros">
<td>{{libro.titulo}}</td>
<td>{{libro.autor}}</td>
</tr>
</table>
</div>
```

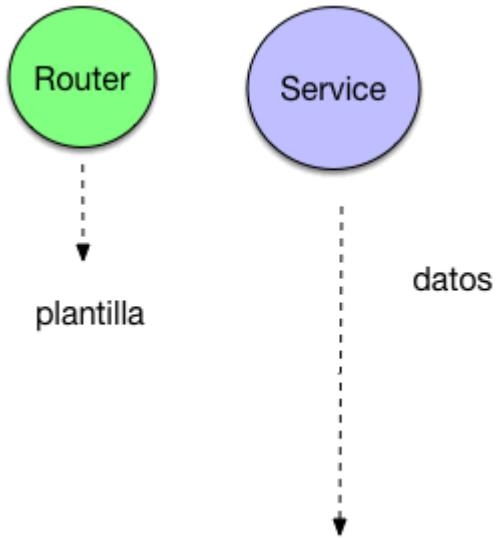
Todo debería funcionar perfectamente. Sin embargo cuando pulsamos el botón de verLista el resultado no es el esperado.

titulo paginas

Los datos no aparecen ,¿ cual es el problema?. Realmente ninguno si esperamos 4 segundos el servicio recibirá los datos y se mostrarán.

titulo	paginas
el juego de ender	
el padrino	

El código funciona pero no es el funcionamiento que nosotros queremos. No podemos presentar algo así . Nuestro problema es un problema de sincronía. El router solicita la plantilla y la carga antes de que los datos del servicio nos lleguen.



Sincronizando con Angular Resolve

Vamos a ver como solventar este problema utilizando Angular Resolve. Para ello deberemos en primer lugar que modificar el código del enrutador que hace referencia a la ruta libros.

```
.when('/libros', {
templateUrl : 'plantillas/sencillo/libros.html',
controller : 'controladorLibros',
resolve: {
libros: function(servicioLibros){
return servicioLibros.buscarTodos().then(function(response) {

return response.data;

});
}
}
}
```

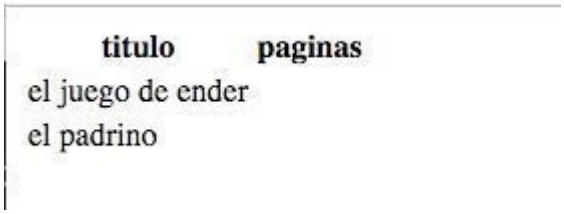
Como podemos observar se ha añadido una nueva propiedad “resolve” esta propiedad se encarga de cargar la lista de libros en la propiedad “libros” antes de que se resuelva la ruta. Nos queda por ver la modificación del controlador para que acepte esta información.

```
controller("controladorLibros",function($scope,libros) {

$scope.libros=libros;
```

```
})
```

El controlador recibe la propiedad de libros que asignamos en el método resolve del enrutador. Ahora ya no tendremos problemas , la plantilla y los datos se cargarán a la vez. Acabamos de construir un ejemplo con Angular Resolve:



titulo	paginas
el juego de ender	
el padrino	

Otros artículos relacionados:

1. El concepto de Angular watch
2. Angularjs Batarang y Scopes
3. Angular.js inyección de dependencia y COC