

## Ítem 9 (A++) - Report

Diseño y Pruebas

Grado de Ingeniería del Software

Curso 3

Armando Garrido Castro  
César García Pascual  
Jorge Puente Zaro  
Manuel Enrique Pérez Carmona  
Pablo Tabares García  
Rafael Trujillo González

Fecha: 9 de mayo de 2018

## Índice

1. Introducción.....	2
2. Problema propuesto.....	2
3. Solución .....	3
4. Conclusión .....	5
5. Ejemplo de uso .....	6

## 1. Introducción

La vista de edición de un Volume, en la que añadimos Newspapers a éste nos planteó la duda de si existía alguna forma de implementar esta funcionalidad de forma que se pudiese realizar de una forma intuitiva. Así descubrimos **Vue.js** (nombrado comúnmente como **Vue**).

Vue es un framework de código abierto de JavaScript para crear interfaces de usuario. Aunque está diseñado para ser integrado en proyectos que usen otras librerías de JavaScript (es incrementalmente adoptable), puede llegar a funcionar como un framework para aplicaciones web.

Entre sus características podemos destacar:

- **Plantillas**  
Vue utiliza plantillas basadas en HTML que permiten enlazar el DOM renderizado a los datos instanciados de Vue, pudiendo realizar cambios de forma transparente cada vez que el estado de la aplicación cambia.
- **Reactividad**  
Los modelos que utiliza son simplemente objetos JavaScript, de manera que cuando los modificas, la vista se actualiza. Cada componente realiza un seguimiento de todas sus dependencias, de forma que cuando se realiza un renderizado, el sistema sabe cuándo y dónde renderizar.
- **Componentes**  
Suponen uno de los mayores potenciales de Vue. No son más que simples objetos que extienden elementos básicos de HTML. Pero en alto nivel, son elementos personalizables a los que podemos asignar un comportamiento determinado
- **Transiciones**  
Vue permite asignar animaciones y efectos de transición a sus elementos, bien utilizando código CSS o JavaScript (personalizado o procedente de una librería externa).
- **Rutas**  
También permite gestionar las rutas de nuestra aplicación web. Esto en aplicaciones de poco tamaño es de gran utilidad puesto que reduce notoriamente el uso de controladores.

## 2. Problema propuesto

Como se ha mencionado anteriormente, en el formulario de edición de un volumen, hemos utilizado Vue para implementar la funcionalidad de añadir o quitar Newspapers a un Volume.

### 3. Solución

Primero, incluiremos Vue en nuestra vista añadiendo la siguiente línea al principio del código de nuestra vista:

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

Tras esto, añadiremos dos contenedores. Uno a la izquierda con los Newspapers disponibles para añadir al volumen, y otro a la derecha con los que ya están añadidos al volumen:

```
<h5><strong><spring:message code="volume.newspapers"/></strong></h5>
<div id="app" class="well well-sm col-md-12" style="background-color: white;">

  <div class="col-md-6">
    <h4><spring:message code="volume.newspapers.all"/></h4>
    <ol>
      <li v-for="item in allNP">
        <a v-on:click="left(item.id)">{{ item.name }}</a>
      </li>
    </ol>
  </div>
  <div class="col-md-6">
    <h4><spring:message code="volume.newspapers.selected"/></h4>
    <ol>
      <li v-for="item in selectedNP">
        <a v-on:click="right(item.id)">{{ item.name }}</a>
      </li>
    </ol>
  </div>
</div>
```

Mediante los atributos “v-for” y “v-on:click” de las etiquetas, hacemos que dichos elementos pasen a ser componentes de Vue, los cuales luego pasarán de un contenedor al otro al hacer click sobre ellos.

Tras esto, creamos el objeto Vue. Primero hacemos que cargue en el lado izquierdo todos los Newspapers del usuario, asignando cada uno a una línea, y que inicialice el contenedor derecho sin ninguno (elementos a los que llamamos antes con el atributo “v-for”).

Acto seguido, creamos los métodos que hacen que al hacer click sobre uno de los Newspapers, éste pase de un contenedor a otro:

```

var app = new Vue({
  el: '#app',
  data: {
    allNP: [
      <jstl:forEach items="{{$allNewspapers}}" var="item">
        {
          id : '{$item.id}',
          version : '{$item.version}',
          name: '{$item.title}',
        }<jstl:if test="{{$not x.last}}">,</jstl:if>
      </jstl:forEach>
    ],
    selectedNP:[],
  },
  methods: {
    left: function (id) {
      console.log(id);
      for(var i=0; i< this.allNP.length; i++){
        if(this.allNP[i].id == id){
          this.selectedNP.push(this.allNP[i]);
          this.allNP.splice(i,1);
        }
      }
    },
    right: function (id) {
      console.log(id);
      for(var i=0; i< this.selectedNP.length; i++){
        if(this.selectedNP[i].id == id){
          this.allNP.push(this.selectedNP[i]);
          this.selectedNP.splice(i,1);
        }
      }
    }
  }
});

```

La línea subrayada en rojo se debe a que la versión de eclipse es bastante anterior a la aparición de Vue, pero al tratarse éste de un framework de JavaScript se ejecuta sin ningún tipo de problema.

Por último, creamos una función que coja los periódicos del contenedor de la derecha y los añada al campo newspapers que se enviará al realizar un “submit” del formulario, junto con otra añada los Newspapers del volumen al contenedor de la derecha en el caso de que la acción a realizar sea la edición del Volume y este ya cuente con Newspapers:

```

$( '#volume' ).submit( function() {
    var list = "";
    for( var i=0; i< app.selectedNP.length; i++) {
        list += app.selectedNP[i].id;
        if( i != app.selectedNP.length-1 ) list += ",";
    }
    $( '#newspapers' ).val( list );
    return true;
});

$( function() {
    if( $( '#newspapers' ).val().length > 0 ) {
        var newspapersId = $( '#newspapers' ).val().split(",");
        $.each( newspapersId, function( key, val ) {
            app.left( val );
        });
    }
});

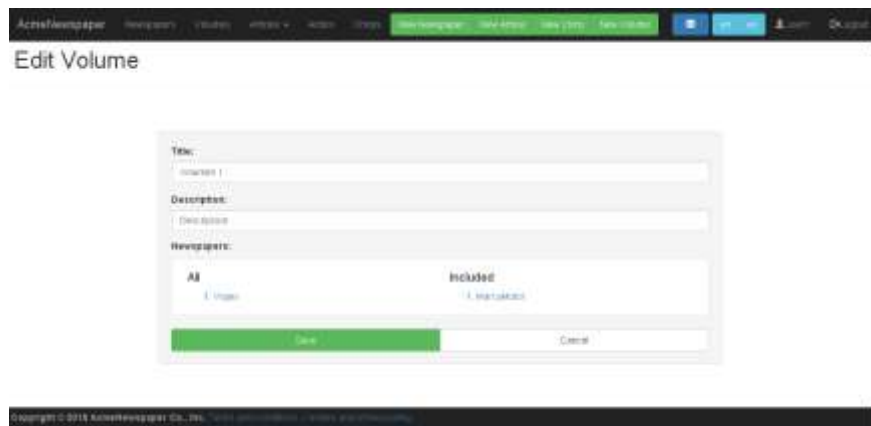
```

## 4. Conclusión

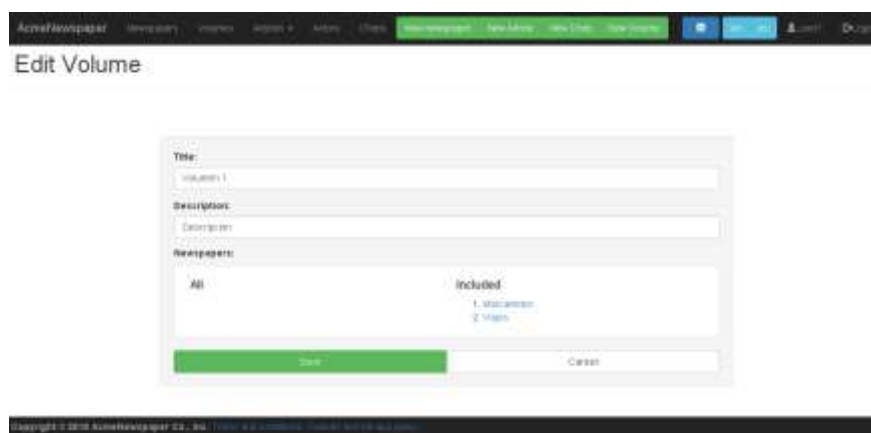
En los últimos años han surgido numerosos frameworks que hacen que el desarrollo de front-end sea mucho más fácil, rápido y cómodo (bootstrap, Angularjs, Vue, etc). Esto soluciona el problema de lo tedioso que puede suponer dar estilo a las vistas de una aplicación web utilizando CSS directamente, ya que hoy en día es muy raro el hecho de programar sin utilizar un framework que evite tener que implementar todo a mano.

## 5. Ejemplo de uso

Accedemos a la vista de edición de un volumen, y observamos que tiene un Newspaper asignado, pero que podemos asignar un segundo:



Si hacemos click en el Newspaper “Viajes”, podemos observar que pasa a la lista “Included”:



Guardamos el cambio y accedemos al display de este Volume:



Como podemos observar, el Newspaper se ha añadido correctamente.