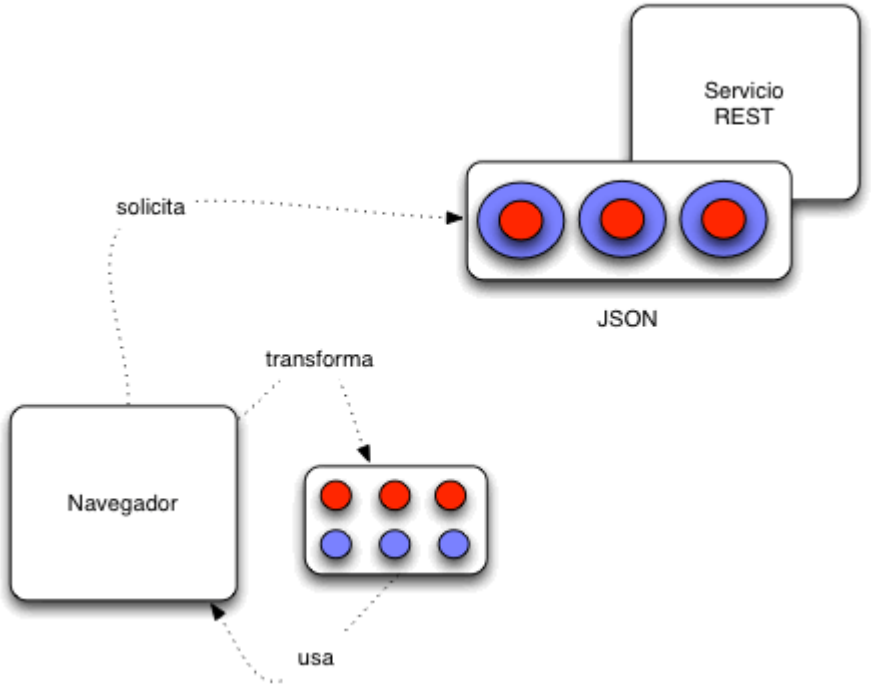
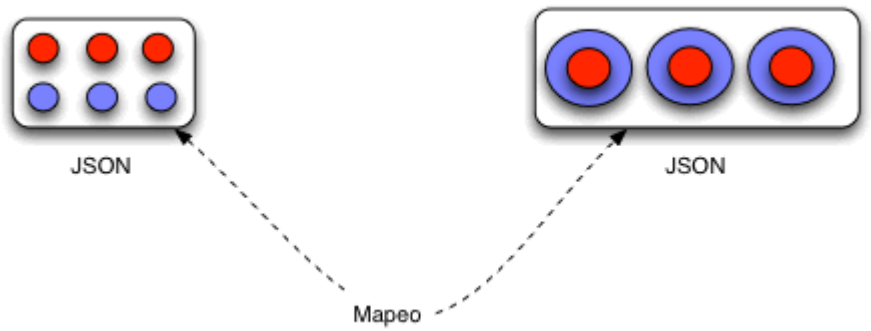
En muchas ocasiones nos encontramos realizando peticiones AJAX desde JavaScript a servicios REST. En un porcentaje importante de las situaciones los datos que recibimos en JSON no son los datos que deseamos y deberemos realizar alguna transformación o modificación.



JavaScript Map

Muchas veces para realizar esta operación recorremos la lista de objetos y obtenemos los nuevos datos que queremos a través de alguna operación un tanto compleja. Sin embargo nos olvidamos muchas veces que JavaScript soporta la función map que nos permite realizar un mapeo rápido.



Vamos a ver un ejemplo sencillo para ello partiremos de la siguiente estructura JSON que nos devuelve el servidor en la url gastos.

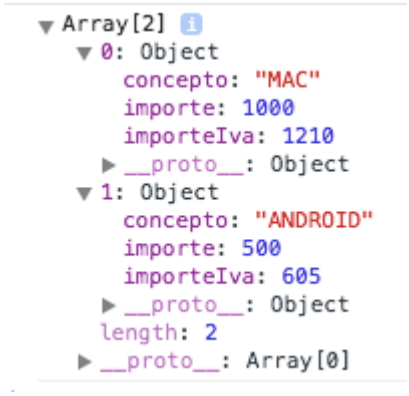
```
[{"concepto":"mac","importe":1000},{"concepto":"android","importe":500
}]
```

Son los datos de una sencilla compra los cuales queremos transformar a un formato diferente en el cual nos aparezca el concepto en mayúscula y dispongamos tanto del importe como el importe con iva.

```
<html>
<head>
<script src="jquery-1.11.2.min.js"></script>
<script>
$(document).ready(function() {

$("#json").click(function() {

$.getJSON("gastos",function(datos) {

var nuevosDatos=datos.map(function(elemento) {
```

```
return {

concepto:elemento.concepto.toUpperCase(),
importe:elemento.importe,
importeIva:elemento.importe*1.21
}

});
console.log(nuevosDatos);

});
});

});

</script>
</head>
<body>
<input type="button" value="json" id="json"/>

<div id="zona">
</div>
</body>
</html>
```

Como podemos ver la función map transforma los datos en otra estructura la cual imprimimos por la consola.



Acostumbremonos a usar más la programación funcional de Javascript a la hora de trabajar en el día a día.

Otros artículos relacionados : JavaScript NameSpaces , JavaScript y consola , Javascript threads