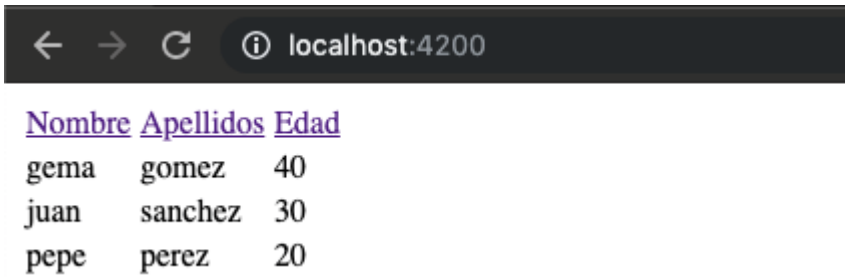
El uso de Angular Sorting es muy necesario ya que casi siempre necesitamos ordenar una lista de resultados de una típica tabla . La ordenación más común es por cada una de las columnas. Ahora bien mientras que Angular clásico tenía este soporte a nivel de directivas la nueva versión de Angular reconstruida entera con TypeScript no lo tiene. ¿Cómo podemos realizar unas tareas de ordenado de forma relativamente sencilla?. Vamos a suponer que disponemos de un componente que nos muestra un listado de personas.



Hasta aquí todo correcto , vamos a ver el código fuente del componente que muestra la tabla.

```
<table>
  <tr>
    <td>
      <a href="#" (click)="orden('nombre')">Nombre</a>

    </td>

    <td>
      <a href="#" (click)="orden('apellidos')">Apellidos</a>

    </td>

    <td>
      <a href="#" (click)="orden('edad')">Edad</a>
```

```
    </td>


  </tr>

  <tr *ngFor="let persona of personas">

    <td>{{persona.nombre}}</td>
    <td>{{persona.apellidos}}</td>
    <td>{{persona.edad}}</td>
  </tr>
</table>
```

Angular Sorting lodash

Como podemos ver vamos a usar una función a nivel del componente que nos permita ordenar los datos. En este caso lo primero que voy a hacer es instalar lodash a través de npm para ello ejecutamos el comando:

```
npm install --save-dev lodash
```

Una vez instalada la librería la podremos usar en nuestro componente para solicitar que haga cada una de las ordenaciones basándonos en cómo se ordena cada columna. Para ello usaremos el método .sortBy de lodash y pasaremos como parámetro la propiedad que tenemos de tipoOrden esta manera la librería se encargará de ordenar por nosotros de forma prácticamente transparente.

```
import { Component, OnInit, ChangeDetectionStrategy } from
'@angular/core';
import { Persona } from '../persona';
import _ from "lodash";
```

```
@Component({
  selector: 'app-lista-personas',
  templateUrl: './lista-personas.component.html',
  styleUrls: ['./lista-personas.component.css'],
})
export class ListaPersonasComponent implements OnInit {

  private tipoOrden: string = "nombre";
  private _personas: Persona[] = [];
  constructor() {

    this._personas.push(new Persona("pepe", "perez", 20));
    this._personas.push(new Persona("juan", "sanchez", 30));
    this._personas.push(new Persona("gema", "gomez", 40));



  }

  get personas() {


    return _.sortBy(this._personas, this.tipoOrden);

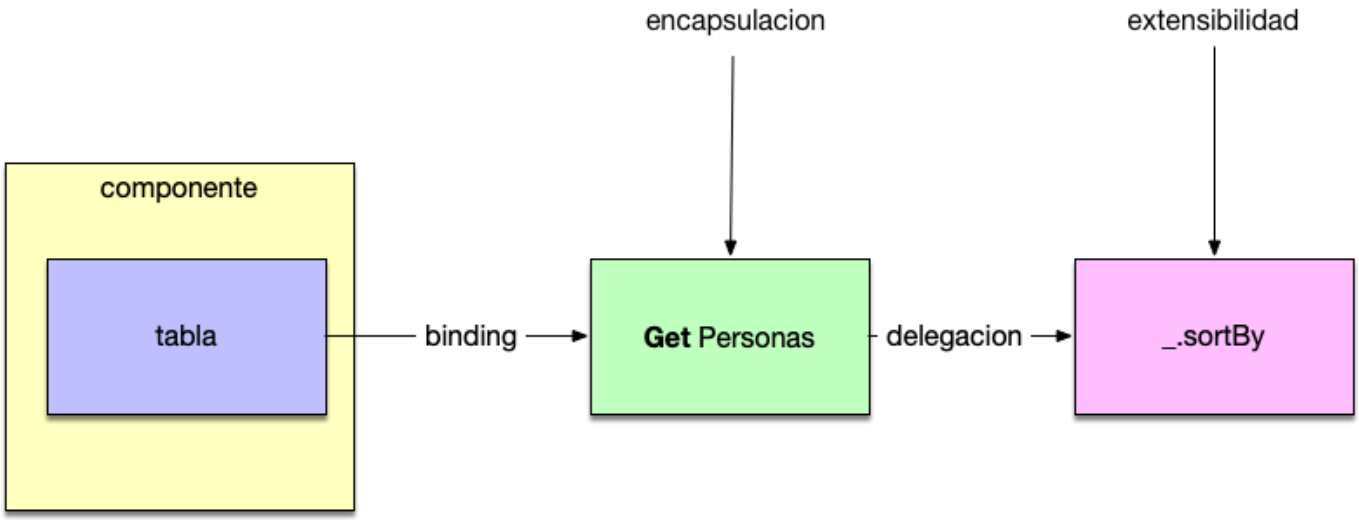
  }
  ngOnInit() {
  }

  orden(tipoOrden) {

    this.tipoOrden = tipoOrden;
  }
```

}

En este caso estamos usando el concepto de encapsulación junto con la librería de lodash para que cada vez que se ejecute una ordenación nos vuelva a redibujar la lista.



De esta manera conseguiremos que al pulsar en cada una de las columnas la tabla se ordene según el criterio que nosotros necesitamos.

Nombre	Apellidos	Edad
gema	gomez	40
pepe	perez	20
juan	sanchez	30

Acabamos de usar lodash como librería de JavaScript extendiendo las capacidades de Angular para permitir la ordenación de forma rápida en el cliente sin tener que ejecutar continuamente llamadas Ajax.

Otros artículos relacionados

1. Angular Lazy Loading Modules y sus opciones
2. Angular custom Directive y sus tipos
3. Angular ngIf, opciones y componentes
4. https://lodash.com/