Aplicaciones de Internet

Practica 2

Objetivo

Realizar un webservice para una el control de los empleados de una empresa utilizando la arquitectura RESTful y consumirla con AngularJS.

Desarrollo

Para realizar esta práctica se utilizó la arquitectura de API REST para la creación de los servicios. Posteriormente se utilizó AngularJS para consumir la API y mostrarla al usuario (Front-end).

Primeramente se construyó el modelo de un empleado el cual contiene las siguientes propiedades.

```
    employee_id
    name
    last_name
    rfc
    gender
    birthday
    phone_number
    email
    employment
    hire_date
    salary
```

Posteriormente se definieron las rutas `api/employees` y `api/employees/:id`, la primera implementaba los métodos get y post los cuales servian para buscar a todos los empleados y agregar un empleado respectivamente. La segúnda ruta implementaba los métodos get put y delete, el primero para encontrar un usuario específico por su id, el segundo para editar un empleado y el tercero para eliminarlo.

Una vez implementados los métodos que realizaban dichas funciones se tenía una API totalmente funcional para poder consumirla.

Para esto se utilizo el framework AngularJS el cual tiene la arquitectura MVC vista en clase.

El primer problema al utilizar Angular fue que los archivos a utilizar para la vista serían formato html y no conseguia poder hacer un renderizado por lo cual la solución que encontre fue utilizar la herramienta ejs, instalandola y agregando las siguientes lineas de código para utilizarla

```
app.engine('html', require('ejs').renderFile);
app.set('view engine', 'html');
```

de esta manera ya podría renderizar mi vista principal utilizando `res.render();` Una vez hecho esto ya podía trabajar en la vista con Angular.

Para esto se creó una layout que sería donde se cargarían los demás elementos. Uno de ellos una barra de menú obtenida de bootstrap, una tabla donde se mostrarían los empleados, un form para poder realizar el add y update, y un footer.

Posteriormente en la lógica se creo un método con get para poder traer todos los empleados, el problema enfrentado en este punto fue que el método de la documentación estaba descontinuado por lo cual tuve que buscar el método actual quedando el siguiente.

```
$http.get('api/employees')
        .then(function(response) {
            $scope.employees = response.data;
        }, function(response) {
            $scope.employees = 'Something went wrong';
        });
```

Lo mismo realicé para los diferentes métodos como post, put, delete quedando diferente el post y put debido a que llevan un parámentro extra que es el objeto que se enviará.

Lo siguiente fue conectar estos métodos con la vista, utilicé el atributo `ng-click=""` que utiliza Angular para llamar a una funcion al hacer click en el elemento. También utilicé `ng-model=""` para poder obtener el valor que contenia un input. Y `{{ employee.sth }}` para obtener en la vista el valor de cualquier variable en el scope.

El método más complicado de implementar fue el de doble click en una fila de la tabla debido a que necesiaba obtener el valor del id de dicha fila. Al final comprendí que éste podía ser mandando como parametro al método en este caso `doubleClick(employee)` el cual en esta ocasión en un atributo `ng-DblClikc=""`.

Por ultimo agregue clases de boostrap para mejorar visualmente.

Conclusión

Al realizar esta práctica pude entender mejor como poder crear mi propio webservice. Entendí mejor los conceptos vistos en clase y me quedó más claro la diferencia entre el servidor y el usuario que era una de mis principales dudas. También pude aprender a utilizar angular y mejorar el uso de boostrap ya en ciertas ocasiones lo había utilizado pero solo elementos más básicos.