**Lesson33 Promises VS Observable in Angular**

**Notes:-**

**1- The differences are shown in the table below**

|  |  |
| --- | --- |
| **Promise** | **Observable** |
| Emits a single value | Emits multiple values over a period of time |
| Not Lazy | Lazy. An Observable is not called until we subscribe to the Observable |
| Cannot be cancelled | Can be cancelled using the unsubscribe() method |
|  | Observable provides operators like map, forEach, filter, reduce, retry, retryWhen etc. |

**2-You can think of an Observable like a stream which emits multiple items over a period of time and the same callback function is called for each item emitted.**

**3-we use the same API to handle asynchronous data whether that data is emitted as a single value or multiple values over a period of time.**

**4-A Promise is not lazy where as an Observable is Lazy**.

**5-we see that in the Promise, we see that it will fire the request off since we call the component, while in the Observable it will be called when we reach to the command of calling method**

**ngOnInit() {**

**let empCode: string = this.\_activatedRoute.snapshot.params['code'];**

**this.\_employeeService.getEmployeeByCode(empCode)**

**.then((employeeData) => {**

**if (employeeData == null) {**

**this.statusMessage =**

**'Employee with the specified Employee Code does not exist';}**

**else {**

**this.employee = employeeData;}},**

**(error) => {**

**this.statusMessage =**

**'Problem with the service. Please try again after sometime';**

**console.error(error);**

**});}**

**getEmployeeByCode(empCode: string): Promise<IEmployee> {**

**return this.\_http.get("http://localhost:31324/api/employees/" + empCode)**

**.map((response: Response) => <IEmployee>response.json())**

**.toPromise()**

**.catch(this.handlePromiseError);}**