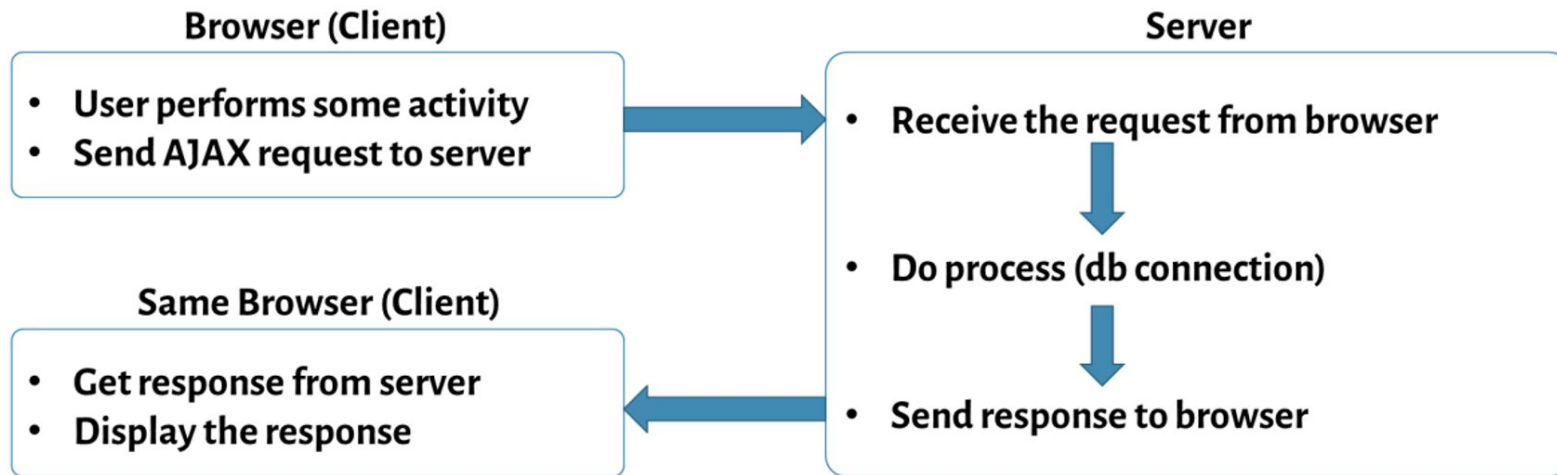


Angular

Anil K

AJAX

- ▶ AJAX (Asynchronous JavaScript And Xml) is not a language, but it is a “concept”, which is used to “send a background request from browser to server” and also “get background response from server to browser”, without refreshing (reloading) the web page in the browser.
- ▶ AJAX allows us to interact with the server and get some data from server, without refreshing the full web page.
- ▶ Ex: Facebook like button, comments, IRCTC search trains.



Execution Flow of AJAX

Advantages of AJAX

- ▶ Executes faster
- ▶ Less burden on browser (client) and server
- ▶ Better user experience.

Types of AJAX Request

- ▶ Get : Used to retrieve / search data from server
- ▶ Post : Used to insert data to server.
- ▶ Put : Used to update data on server.
- ▶ Delete : Used to delete data from server

“@angular/common/http” package

- ▶ The “@angular/common/http” package provides necessary services to send ajax request to server and get ajax response from server.

► **Steps for working with “@angular/common/http” package:**

```
import “@angular/common” package in “package.json”: “dependencies”: {  
  “@angular/common:”: “latest” }
```

► **Import “HttpClientModule” module:**

```
import { HttpClientModule, HttpClient } from “@angular/common/http”;
```

► **Import “HttpClientModule” in “AppModule”:**

```
@NgModule( { ..., imports: [ ..., HttpClientModule ] } )  
class AppModule { }
```

► **Import “HttpClient” service:**

```
import { HttpClient } from “@angular/common/http”;
```

► **Get “HttpClient” service in “AppComponent”:**

```
constructor(@Inject(HttpClient) private http : HttpClient) { }
```

► **Send “get” request to server:**

```
this.http.get<modelclassname> (“url”, { responseType: “json | text”  
}).subscribe(this.successcallback, this.errorcallback);
```

► **Send “post” request to server:**

```
this.http.post (“url”, { data }, { responseType: “json | text”  
}).subscribe(this.successcallback, this.errorcallback);
```

► **Send “put” request to server:**

```
this.http.put("url", { data }, { responseType: "json | text"
}).subscribe(this.successcallback, this.errorcallback);
```

► **Send “delete” request to server:**

```
this.http.delete("url", { responseType: "json | text"
}).subscribe(this.successcallback, this.errorcallback);
```

► **Define “success” callback function:**

```
successcallback = (response) => { //do something with response }
```

► **Define “error” callback function:**

```
errorcallback = (error) => { //do something with error }
```

Creating Application

c:\angular\app1\src\app\app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [ AppComponent ],
  imports: [ BrowserModule, FormsModule, HttpClientModule ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

c:\angular\app1\src\app\app.component.ts

```
import { Component, Inject } from '@angular/core';
import { HttpClient } from "@angular/common/http";
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  message: string;

  constructor(@Inject(HttpClient) private http: HttpClient) { }

  onDataGetClick() {
    this.http.get("/serviceUrl", {
      responseType: "text"
    }).subscribe(this.onAjaxSuccess, this.onAjaxError);
  }

  onAjaxSuccess = (response) => { this.message = response; }
  onAjaxError = (error) => { alert(error); }
}
```


c:\angular\app1\src\app\app.component.html

```
<div>
```

```
  <h4>Ajax - Java - Simple</h4>
```

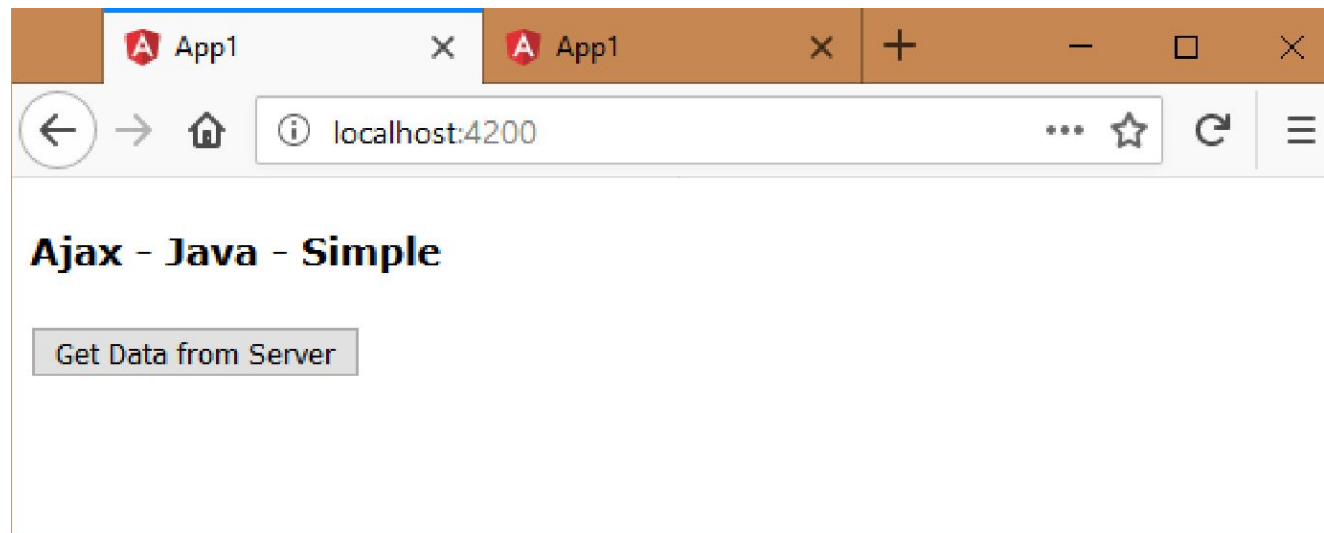
```
  <input type="button" value="Get Data from Server"  
  (click)="onGetDataClick()">
```

```
  <div>
```

```
    {{message}}
```

```
  </div>
```

```
</div>
```



Http Interceptor

- ▶ Http Interceptor is an angular service, that adds one or more request headers automatically for each ajax request sent from the application.
- ▶ These are used to implement authentication.
- ▶ Authentication is a process of checking whether the user is a valid user or not.
- ▶ First, the server generates a random number, stores it at server and also sends it to the client as response. The client (browser) stores it in the localStorage. Next, the client sends the same random number to the server via HTTP header, by using Http Interceptor. Then the server verifies whether the received random number is matching with the stored random number. If matching, it is a valid user; otherwise invalid user. Thus authentication is performed.
- ▶ Http Interceptors added as a provider in the module level; automatically executes for all ajax requests sent via HttpClient within the same module and its child modules.
- ▶ Http Interceptor is a service class that implements HttpInterceptor interface. This interface enforces that the service class must contain a method called “intercept” with two parameters “request” of HttpRequest type and “next” of HttpHandler type and returns HttpEvent type.
- ▶ The “intercept” method automatically executes before an AJAX request (Http Request) is sent to browser.
- ▶ The “request” argument represents the current request. The “next” argument represents the next interceptor.
- ▶ The “HttpEvent” represents each single process that happens before sending http request to server.
- ▶ The “request.clone” method is used to copy the request into a new request object; so that we can add headers to the request.
- ▶ The next.handle() method calls the next interceptor, if any.

► Steps for Working with Http Interceptor

► Create Service:

```
import { Injectable } from "@angular/core";
import { HttpInterceptor, HttpRequest, HttpHandler, HttpEvent } from
"@angular/common/http";
import { Observable } from "rxjs/Observable";
@Injectable( )
export class Serviceclassname implements HttpInterceptor {
  intercept(request, HttpRequest<any>, next:
  HttpHandler):Observable<HttpEvent<any>> {
    request = request.clone( { setHeaders: { "mykey": "myvaluue" } } );
    return next.handle(request);
  }
}
```

► Add Http Interceptor Service as provider in the component:

```
import { HTTP_INTERCEPTORS } from "@angular/common/http";
@Component( { ..., providers: [ { provide: HTTP_INTERCEPTORS, useClass:
SampleInterceptorService, multi: true } ] } )
export class Moduleclassname { }
```

- ▶ **AJAX - Http Interceptors - Get - Example**
- ▶ Creating Application
- ▶ Open Command Prompt and enter the following commands:
- ▶ `cd c:\angular`
- ▶ `ng new app1`
- ▶ `cd c:\angular\app1`
- ▶ `ng g class Employee`
- ▶ `ng g service Employees`
- ▶ `ng g service SampleInterceptor`

► c: \angular\ app1\ src\ app\ employee.ts

```
export class Employee {
```

```
  empid: number;
```

```
  empname: string;
```

```
  salary: number;
```

```
  constructor(a, b, c) {
```

```
    this.empid = a;
```

```
    this.empname = b;
```

```
    this.salary = c;
```

```
  }
```

```
}
```

► c: \angular\ app1\ src\ app\ employees.service.ts

```
import {Injectable,Inject} from '@angular/core';
import {Employee} from "../employee";
import {HttpClient} from "@angular/common/http";
@Injectable() export class EmployeesService {
  constructor(@Inject(HttpClient) private http: HttpClient) {}
  resolve: any;
  reject: any;
  getEmployees() {
    this.http.get < Employee > ("/SampleServlet", {
      responseType: "json"
    }).subscribe(this.onAjaxSuccess, this.onAjaxError);
    return new Promise(this.promiseCallback);
  }
  promiseCallback = (resolve, reject) => {
    this.resolve = resolve;
    this.reject = reject;
  }
  onAjaxSuccess = (response) => {
    this.resolve(response);
  }
  onAjaxError = () => {
    this.reject("Failed");
  }
}
```

► c: \angular\ app1\ src\ app\ sample - interceptor.service.ts

```
import {Injectable} from '@angular/core';  
import {HttpInterceptor,HttpRequest,HttpHandler,HttpEvent} from  
"@angular/common/http";  
import {Observable} from "rxjs/Observable";
```

```
@Injectable() export class SampleInterceptorService implements HttpInterceptor  
{  
  intercept(request: HttpRequest < any > , next: HttpHandler): Observable <  
    HttpEvent < any >> {  
    request = request.clone({  
      setHeaders: {  
        "mykey": "myvalue"  
      }  
    });  
    return next.handle(request);  
  }  
}
```

► c: \angular\ app1\ src\ app\ app.module.ts

```
import {BrowserModule} from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from "@angular/forms";
import { HttpClientModule } from "@angular/common/http";
import { AppComponent } from './app.component';
import { EmployeesService } from './employees.service';
import { HTTP_INTERCEPTORS } from "@angular/common/http";
import { SampleInterceptorService } from "./sample-interceptor.service";
@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, FormsModule, HttpClientModule],
  providers: [EmployeesService, {
    provide: HTTP_INTERCEPTORS,
    useClass: SampleInterceptorService,
    multi: true
  }],
  bootstrap: [AppComponent]
}) export class AppModule {}
```


► c: \angular\ app1\ src\ app\ app.component.ts

```
import {Component,Inject} from '@angular/core';
import {EmployeesService} from "../employees.service";
import {Employee} from "../employee";
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
}) export class AppComponent {
  employees: Employee[] = [];
  constructor(@Inject(EmployeesService) private employeesService: EmployeesService) {}
  onGetDataClick() {
    this.employeesService.getEmployees().then(this.onPromiseSuccess, this.onPromiseError);
  }
  onPromiseSuccess = (response) => {
    this.employees = response;
  }
  onPromiseError = () => {
    alert("error");
  }
}
```

► c:\angular\app1\src\app\app.component.html

<div>

<h4>Ajax - Http Interceptors - Promise - Java - Get</h4>

<input type="button" value="Get Data" (click)="onGetDataClick()">

<table border="1">

<tr>

<th>Emp ID</th>

<th>Emp Name</th>

<th>Salary</th>

</tr>

<tr *ngFor="let employee of employees">

<td>{{employee.empid}}</td>

<td>{{employee.empname}}</td>

<td>{{employee.salary}}</td>

</tr>

</table>

</div>

AJAX - Java - Http Interceptors - Get - Example

- ▶ Setting-up Environment for Java
- ▶ Install Java from “<https://java.com/en/download>”.
- ▶ Add “C:\Program Files\Java\jdk1.8.0_172\bin” as “Path” of system variables.
- ▶ Add “JAVA_HOME” with “C:\Program Files\Java\jdk1.8.0_172” in system variables.
- ▶ Download tomcat from “<https://tomcat.apache.org/download-90.cgi>”. Click on “zip” in “Core”. You will get a file called “apache-tomcat-9.0.7.zip”. Right click on “apache-tomcat-9.0.7.zip” and click on “Extract All”. Copy all contents of the extracted folder into “c:\tomcat” folder.
- ▶ Open Command Prompt and enter the following commands: `cd c:\tomcat\bin`
`startup.bat`

► c: \tomcat\ webapps\ ROOT\ WEB - INF\ classes\ SampleServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SampleServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        PrintWriter out = response.getWriter();
        if (request.getHeader("mykey").equals("myvalue")) {
            out.println("[ { \"empid\": 1, \"empname\": \"Scott\", \"salary\": 4000 }, {
                \"empid\": 2, \"empname\": \"Allen\", \"salary\": 7500 }, { \"empid\": 3, \"empname\":
                \"Jones\", \"salary\": 9200 }, { \"empid\": 4, \"empname\": \"James\", \"salary\": 8400 }, {
                \"empid\": 5, \"empname\": \"Smith\", \"salary\": 5600 } ]");
        } else {
            out.println("[ { \"empid\": 0, \"empname\": \"no data\", \"salary\": 0 } ]");
        }
    }
}
```

► c:\tomcat\webapps\ROOT\WEB-INF\web.xml

```
<?xml version="1.0" encoding="UTF-8"?> <web-app  
xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" version="4.0"  
metadata-complete="true">
```

```
<display-name>Welcome to Tomcat</display-name>
```

```
<description>    Welcome to Tomcat  </description>
```

```
<servlet>
```

```
<servlet-name>SampleServlet</servlet-name>
```

```
<servlet-class>SampleServlet</servlet-class>
```

```
</servlet>
```

```
<servlet-mapping>
```

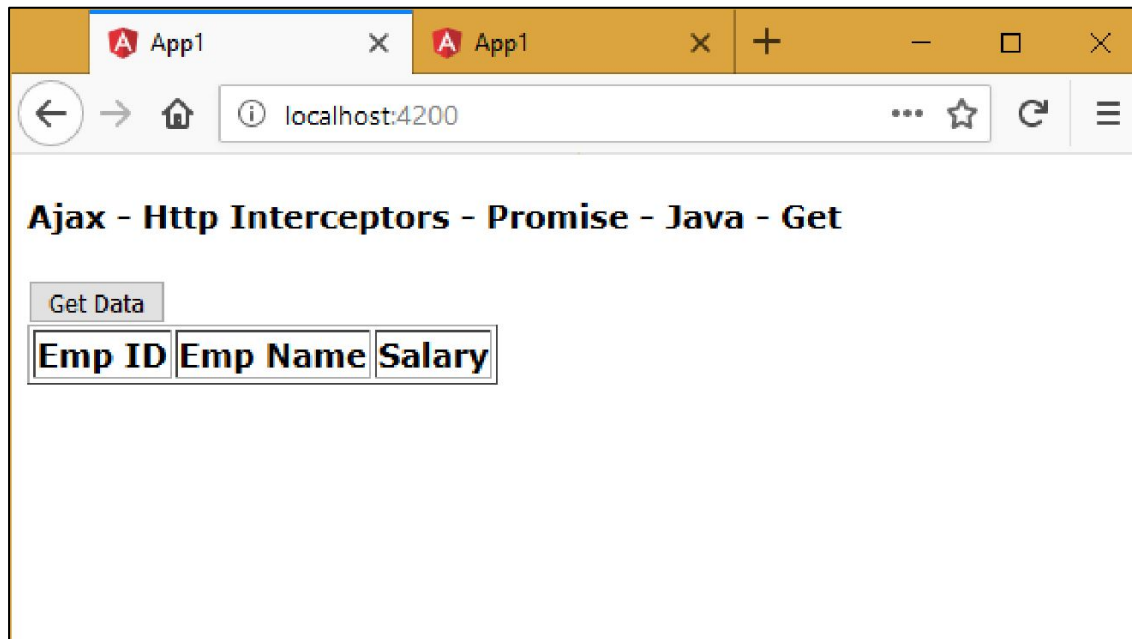
```
<servlet-name>SampleServlet</servlet-name>
```

```
<url-pattern>/SampleServlet</url-pattern>
```

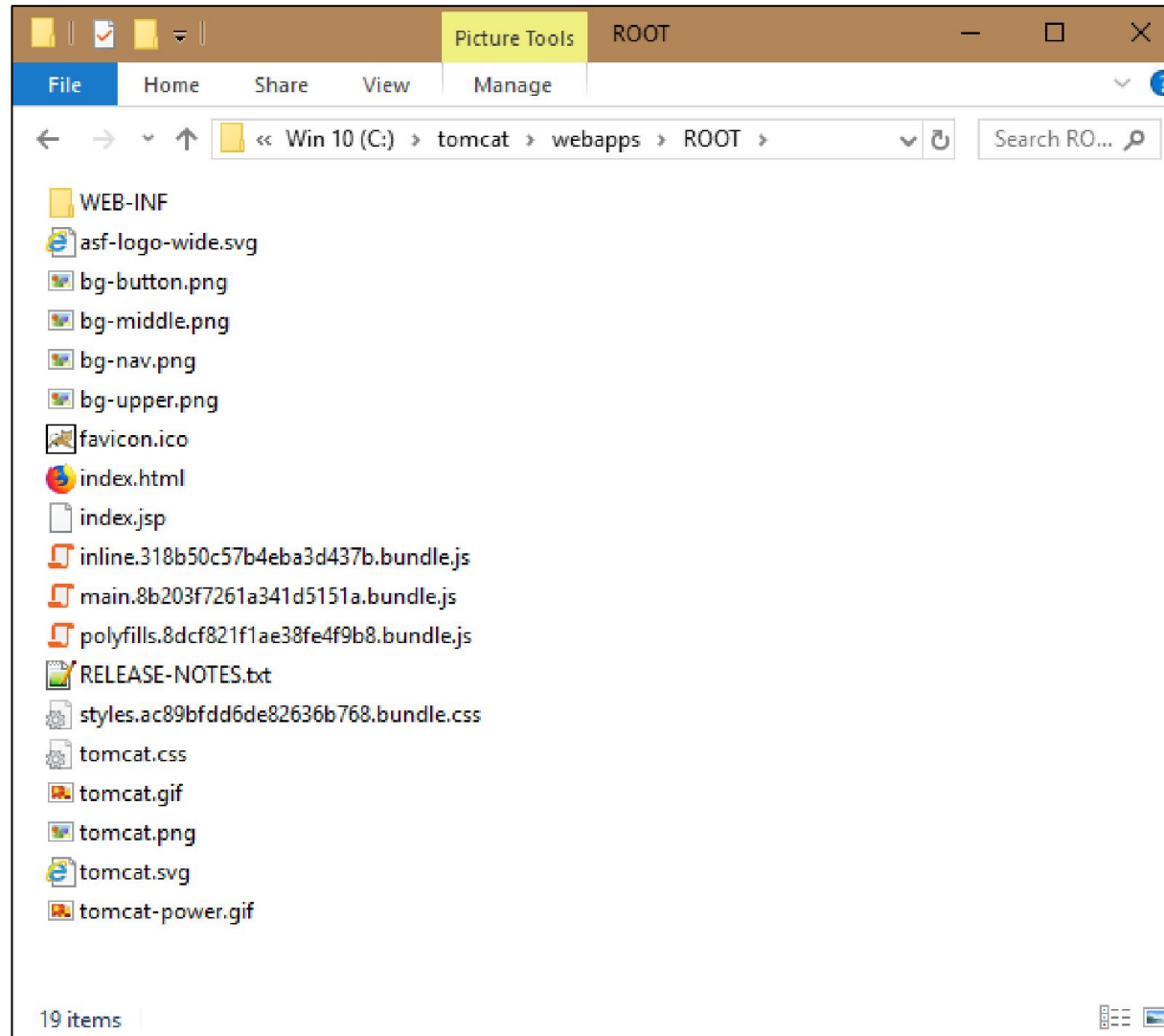
```
</servlet-mapping>
```

```
</web-app>
```

- ▶ **Executing the application:**
- ▶ Open Command Prompt and enter the following commands:
- ▶ `cd c:\tomcat\webapps\ROOT\WEB-INF\classes`
- ▶ `javac -cp c:\tomcat\lib\servlet-api.jar SampleServlet.java`
- ▶ `cd c:\angular\app1`
- ▶ `ng build --prod`
- ▶ `ng serve`
- ▶ Open the browser and enter the following URL: `http://localhost:4200`



- Copy all files from “c:\angular\app1\dist” folder to “c:\tomcat\webapps\ROOT”.



- ▶ Open the browser and enter the following URL:
`http://localhost:8080/index.html`

