1)

A single-page application works in the browser and requires no page reloads and no extra time for waiting. The page doesn’t need to be updated since content is downloaded automatically. We use these types of applications every day. Just think of Facebook or Trello. These are excellent examples of single-page apps.

2)

Functionality out of the box

TypeScript

Consistency

Maintainability

Productivity

3)

JavaScript Object - Define JavaScript object for employee

<script>

var emp = {firstName:"John",lastName:"Stoke", salary:5000, permanentStaff:false};

console.log(emp);

</script>

JSON - Define JSON for employee details and parse

<script>

var emp = JSON.parse('{"firstName":"John", "lastName":"Stoke", "salary":"5000","permanentStaff":"false"}');

console.log("firstName : "+emp.firstName);

console.log("lastName : "+emp.lastName);

console.log("salary : "+emp.salary);

console.log("permanentStaff : "+emp.permanentStaff);

JSON - Add department details to employee

<script>

const data = '{"firstName":"John", "lastName":"Stoke", "salary":"5000", "permanentStaff":"false", "Department":[{"id":"3","name":"payroll"}]}';

var emp = JSON.parse(data);

console.log("firstName : "+emp.firstName);

console.log("lastName : "+emp.lastName);

console.log("salary : "+emp.salary);

console.log("permanentStaff : "+emp.permanentStaff);

console.log("Department id : "+emp.Department[0].id);

console.log("Department name : "+emp.Department[0].name);

JSON - Add skill details to employee

<script>

const data = '{"firstName":"John", "lastName":"Stoke", "salary":"5000", "permanentStaff":"false", "Department":[{"id":"3","name":"payroll"}], "skill":[{"id":"1", "value":"HTML"},{"id":"2", "value":"CSS"},{"id":"2", "value":"Javascript"}]}';

var emp = JSON.parse(data);

console.log("firstName : "+emp.firstName);

console.log("lastName : "+emp.lastName);

console.log("salary : "+emp.salary);

console.log("permanentStaff : "+emp.permanentStaff)

console.log("Department id : "+emp.Department[0].id);

console.log("Department name : "+emp.Department[0].name);

for(var i=0; i<emp.skill.length; i++)

{

console.log("Skill "+(i+1)+" : "+emp.skill[i].id+", "+emp.skill[i].value);

}

</script>

Session 2:

Employee.ts

export interface Employee {

id:number;

name:string;

salary:number;

permanent:boolean;

}

Employee-test.ts

import { Employee } from './Employee';

var emp:Employee = {

id:3,

name:"John",

salary:5000,

permanent:true,

}

console.log("id : "+emp.id);

console.log("name : "+emp.name);

console.log("salary : "+emp.salary);

console.log("permanent : "+emp.permanent);

Department.ts

export interface Department{

id:number;

name:string;

}

Employee.ts

import { Department } from './Department';

export class Employee implements Department{

id:number;

name:string;

salary:number;

permanent:boolean;

constructor(id:number, name:string, salary:number, permanent:boolean){

this.id = id;

this.name = name;

this.salary = salary;

this.permanent = permanent;

}

}

Employee-test.ts

import { Department } from './Department';

import { Employee } from './Employee';

var emp1 = new Employee(123, "John", 5000, true);

console.log("id: "+emp1.id);

console.log("name: "+emp1.name);

console.log("salary: "+emp1.salary);

console.log("permanent: "+emp1.permanent);

var dept:Department = {

id:1,

name:"Payroll",

}

console.log("Department id : "+dept.id);

console.log("Department name : "+dept.name);