

# Module 5 Practicals

## Part 1 - ASP.NET Web Forms

### WebForm1.aspx :

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            width: 100%;
        }
        .auto-style2 {
            margin-left: 0px;
        }
        .auto-style3 {
            width: 121px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Welcome to the Web Forms!</h2>
            <table class="auto-style1">
                <tr>
                    <td class="auto-style3">
                        <asp:Label ID="Label1" runat="server" Text="User
Name"></asp:Label></td>
                    <td>
                        <asp:TextBox ID="TextBox1" runat="server"
CssClass="auto-style2"></asp:TextBox></td>
                </tr>
                <tr>
                    <td class="auto-style3">
                        <asp:Label ID="Label2" runat="server" Text="Upload
a File"></asp:Label></td>
                    <td>
                        <asp:FileUpload ID="FileUpload1" runat="server"

/></td>
                </tr>
            </table>
        </div>
```

```
</form>
</body>
</html>
```

## **Part 2 - ASP.NET Web API**

### **Global.asax.cs :**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Http;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace WebApplication2_WebAPI
{
    public class WebApiApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            GlobalConfiguration.Configure(WebApiConfig.Register);
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);

            //serialization

            //effect can be seen in internet explorer whenit downloads json file
            //as chrome default content is xml , effect can't be seen in json or xml
            via browser or postman
            var json =
            GlobalConfiguration.Configuration.Formatters.JsonFormatter;
            json.SerializerSettings.Formatting =
            Newtonsoft.Json.Formatting.Indented;

        }
    }
}
```

### **WebApiConfig.cs :**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Http;
```

```

namespace WebApplication2_WebAPI
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services
            // Web API routes

            //enable attribute routing
            config.MapHttpAttributeRoutes();

            //custom route - work in postman , this route will be matched first (first
come first serve)
            config.Routes.MapHttpRoute(
                name: "emp",
                routeTemplate: "api/emp",
                defaults: new { controller="employee" }
            );

            //add default route using convention-based routing
            //priority will be given to attribute routing
            config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}"
            );
        }
    }
}

```

### **EmployeeController.cs :**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using WebApplication2_WebAPI.Models;

namespace WebApplication2_WebAPI.Controllers
{
    public class EmployeeController : ApiController
    {
        Employee[] employees = new Employee[] {
            new Employee { id = 1, name = "Mark", age = 30 },
            new Employee { id = 2, name = "Allan", age = 35 },
            new Employee { id = 3, name = "Johny", age = 21 }
        };
    }
}

```

```

//attribute routing - enabled in WebApiConfig class
[Route("api/employee/details")]
public IEnumerable<Employee> GetAllEmployees()
{
    return employees;
}

[Route("api/employee/details/{id}")]
public IHttpActionResult GetEmployee(int id)
{
    var employee = employees.FirstOrDefault((p) => p.id == id);
    if (employee == null)
    {
        return NotFound();
    }
    return Ok(employee);
}

//parameter binding - primitive->from uri
//                      - complex->from body

// to force opp rules i.e primitive from body & complex from uri
public Employee Post([FromUri] Employee emp) //if above att. route not
given,then this method will
    //not be executed.Instead,GetEmployee method with id executes even
though all 3 params are passed
    //in query string.If both are get,either should exist.
{
    return emp;
}

public Employee Get([FromBody] string name)
{
    var employee = employees.FirstOrDefault((p) => p.name == name);
    return employee;
}
}

```

#### **Employee.cs :**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication2_WebAPI.Models
{
    public class Employee

```

```
{  
    public int id { get; set; }  
    public string name { get; set; }  
    public int age { get; set; }  
}  
}
```