C#Corner
Kolkata Chapter

@csharpcorner
https://www.facebook.com/CsharpCorner

Deep Drive Into ASP.Net Core 2.1 (Day 3)

Presented by
DEBASIS SAHA
C#Corner MVP | Team Lead | Author | Blogger | Speaker

debasis.ds@outlook.com | @debasiskolsaha | https://github.com/debasis-saha

About Me

Debasis Saha

- Team Lead and Consultant
- C# Corner MVP
- DZone MVB

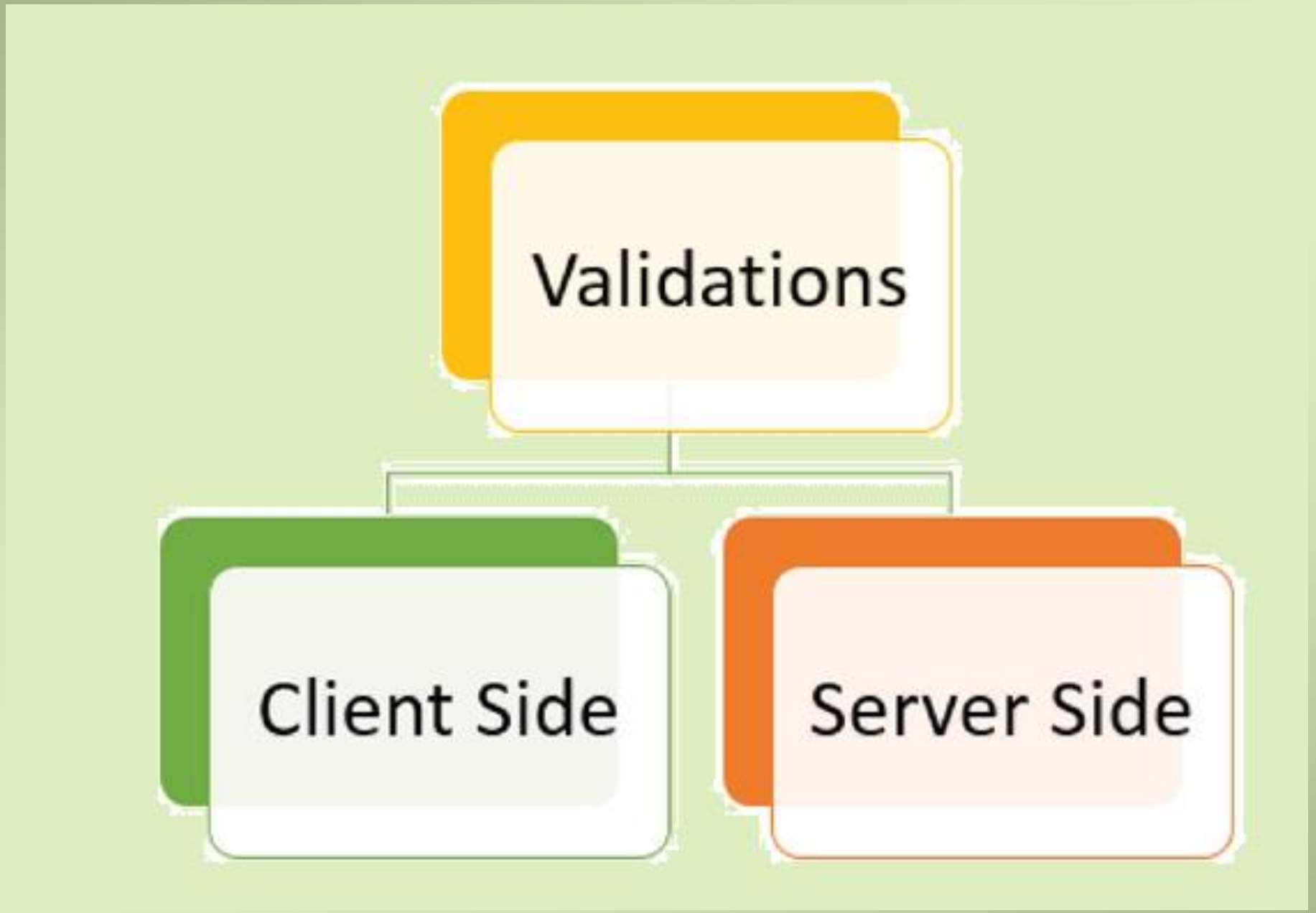
Email : debasis.ds@outlook.com
Twitter : @debasiskolsaha
LinkedIn : https://www.linkedin.com/in/sahadebasis

Agenda

- Form Validations
- Server Side Validation
- Client Side Validation
- Custom Validation
- Filter
- Routing
- Q n A

Form Validations

- Form Validations always ensure that the received data is correct and valid
- Data can be validate either at the Client Side or Server Side
- Form Validation always prevent any type of unwanted attack


Validations
Client Side
Server Side

Server Side Validation

- Server Side validation protect data from dirty input
- Server Side Validation ways
  - Explicit Model Validation (Traditional)
  - Data Annotations Model Validation

Client Side Validation

- Save a number of request from Client Side
- Help the UI Form as a User Friendly Environment
- Client Side Validation ways
  - Using JavaScript Library
  - Using Third Party JS Library / Framework Like Angular, React etc.

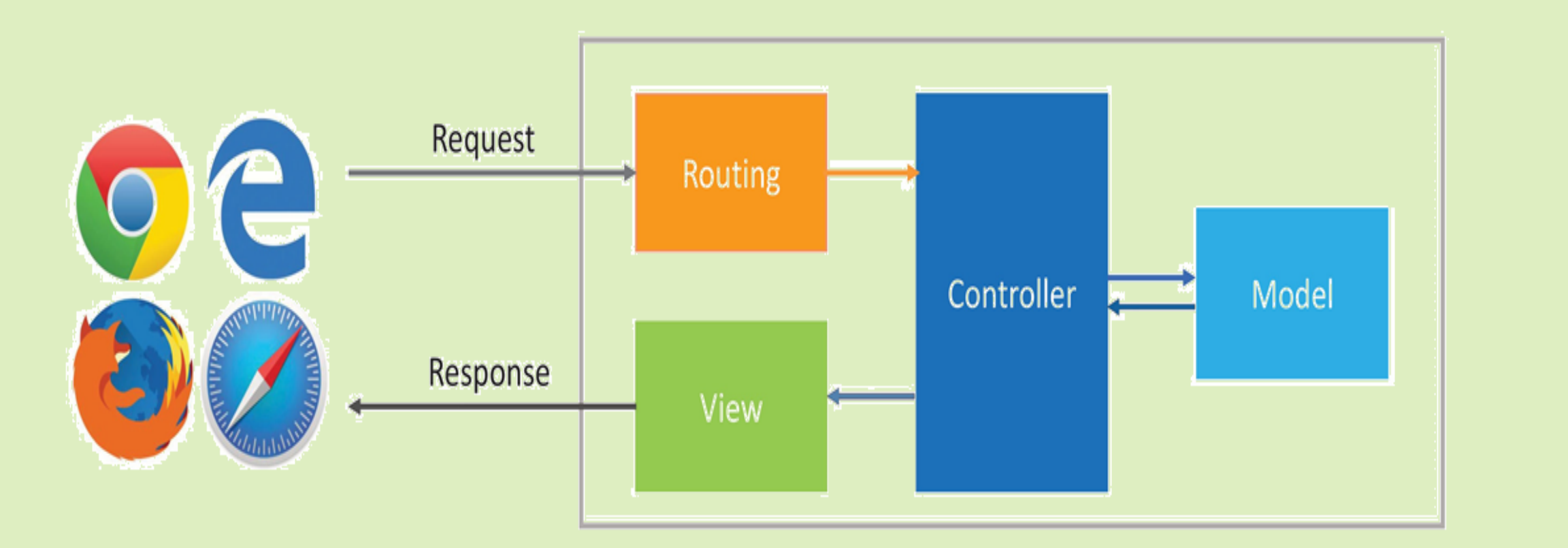
DEMO

On Server Side & Client Side Validation

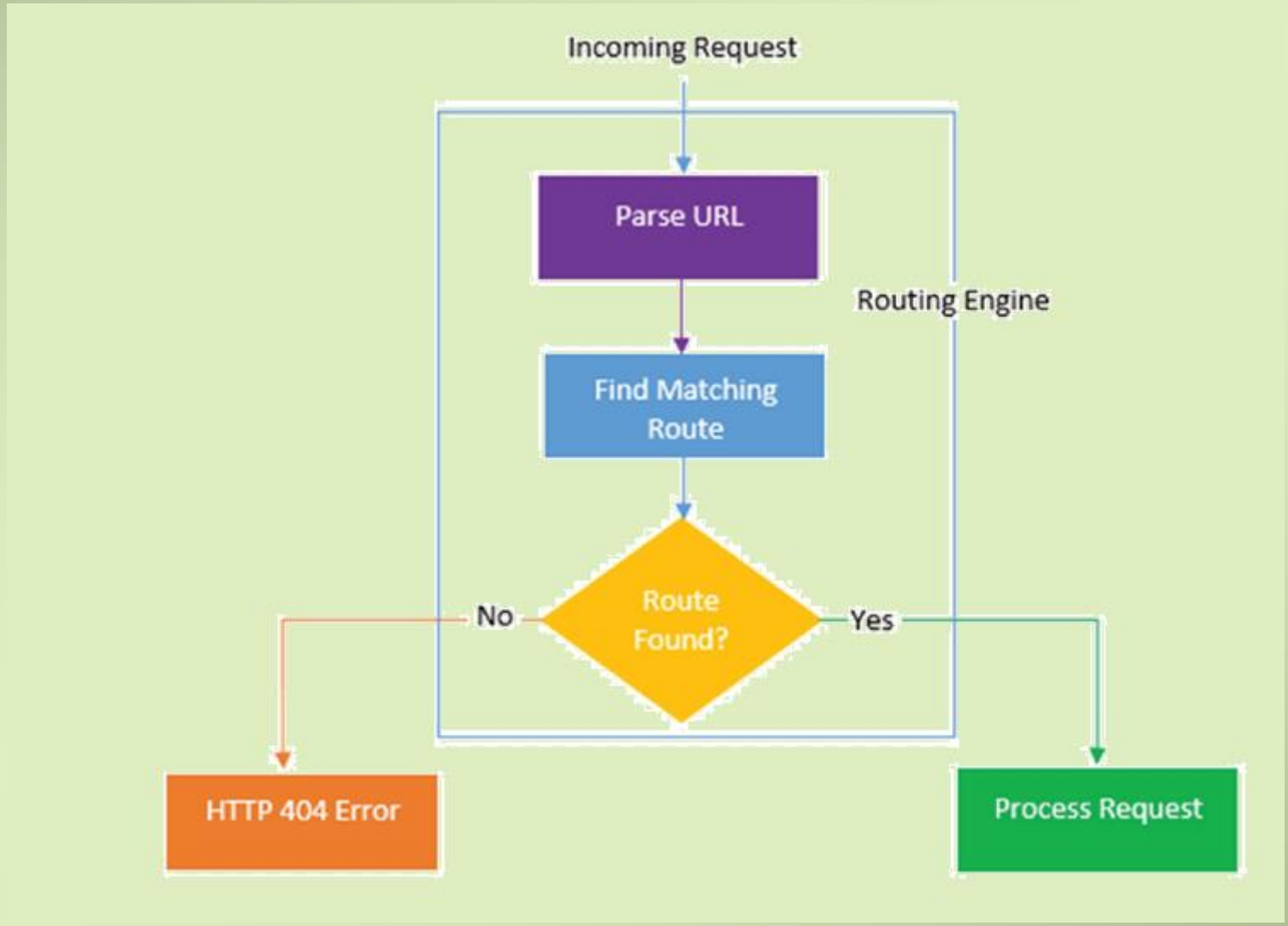
Routing

- Routing is basically a pattern matching system which analysis the incoming request and then decide what need to do this incoming request.
- It s an way of serve user request

Routing Flow



Routing Request Handling



Types of Routing

- In Asp.Net Core 2.1, Routing can be implement by two different ways
  - Convention Based Routing
  - Attribute Routing

Convention Based Routing

```
public class ProductController : Controller
{
    // GET: Product
    0 references
    public ActionResult Index()
    {
        return View();
    }
    // GET: Product/Details/5
    0 references
    public ActionResult Details(int id)
    {
        return View();
    }
    // GET: Product/Create
    0 references
    public ActionResult Create()
    {
        return View();
    }
}
```

```
routes.MapRoute(
    name: "Default",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "Home", action = "Index", id =
      UrlParameter.Optional }
);
```

Attribute Based Routing

```
public class RouteConfig
{
    1 reference
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.MapMvcAttributeRoutes(); //configure attribute routing

        //convention based route
        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id =
              UrlParameter.Optional }
        );
    }
}
```

Attribute Based Routing

```
public class ProductController : Controller
{
    [Route("Product")] // GET: Product
    0 references
    public ActionResult Index()
    {
        return View();
    }
    [Route("Product/Details/{id}")] // GET: Product/Details/5
    0 references
    public ActionResult Details(int id)
    {
        return View();
    }
    [Route("Product/Create")] // GET: Product/Create
    0 references
    public ActionResult Create()
    {
        return View();
    }
}
```

DEMO

Routing & Filter

THANK YOU For Attending the Session

Session Resource is Available at –

- Presentations & Sample Code :-

https://github.com/debasis-saha/Asp.Net_Core_Session