

MVC → open-source web App's flow
Core Web APIs

1.0 → 2015

2.0



6.0 

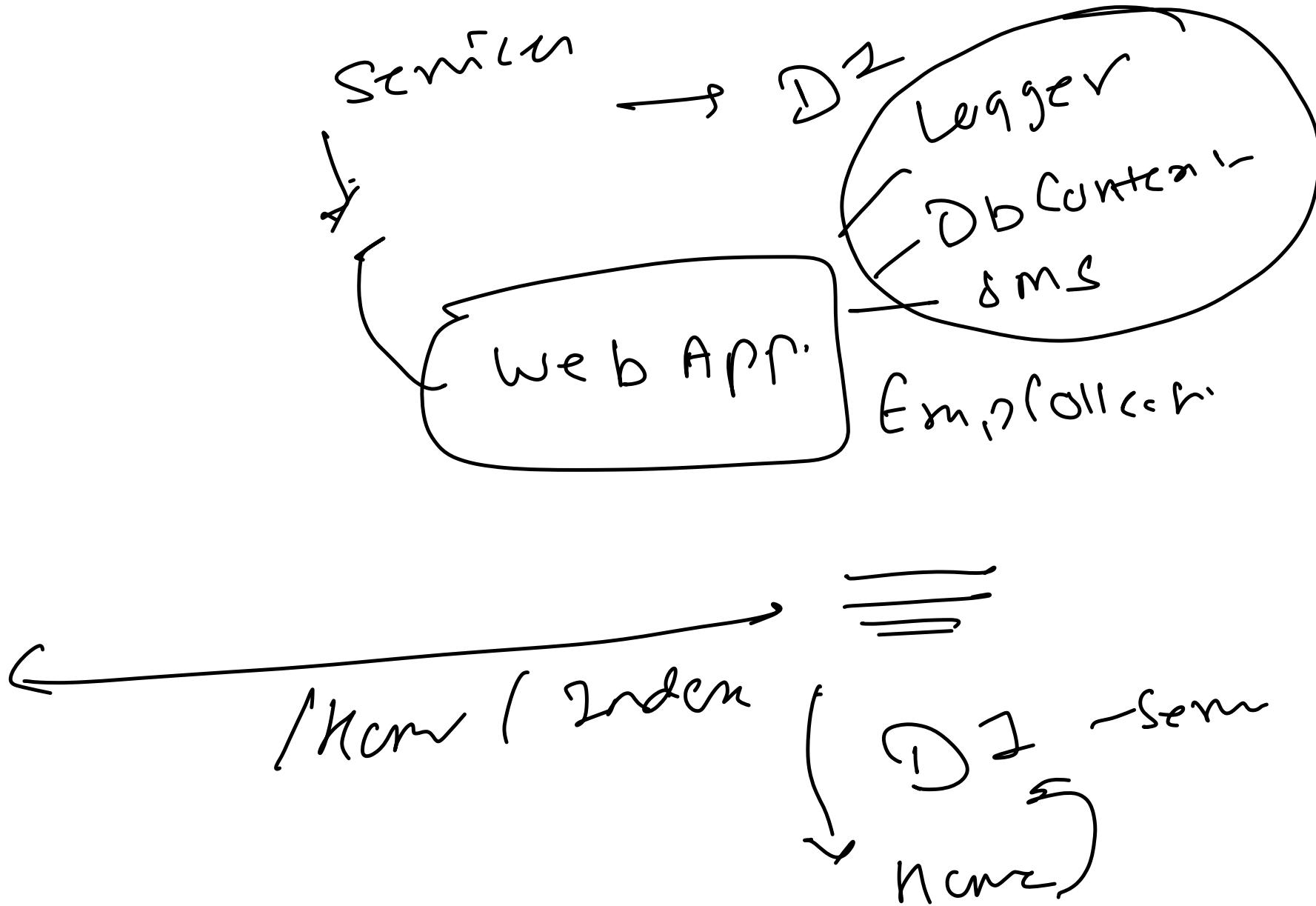
8.0 →

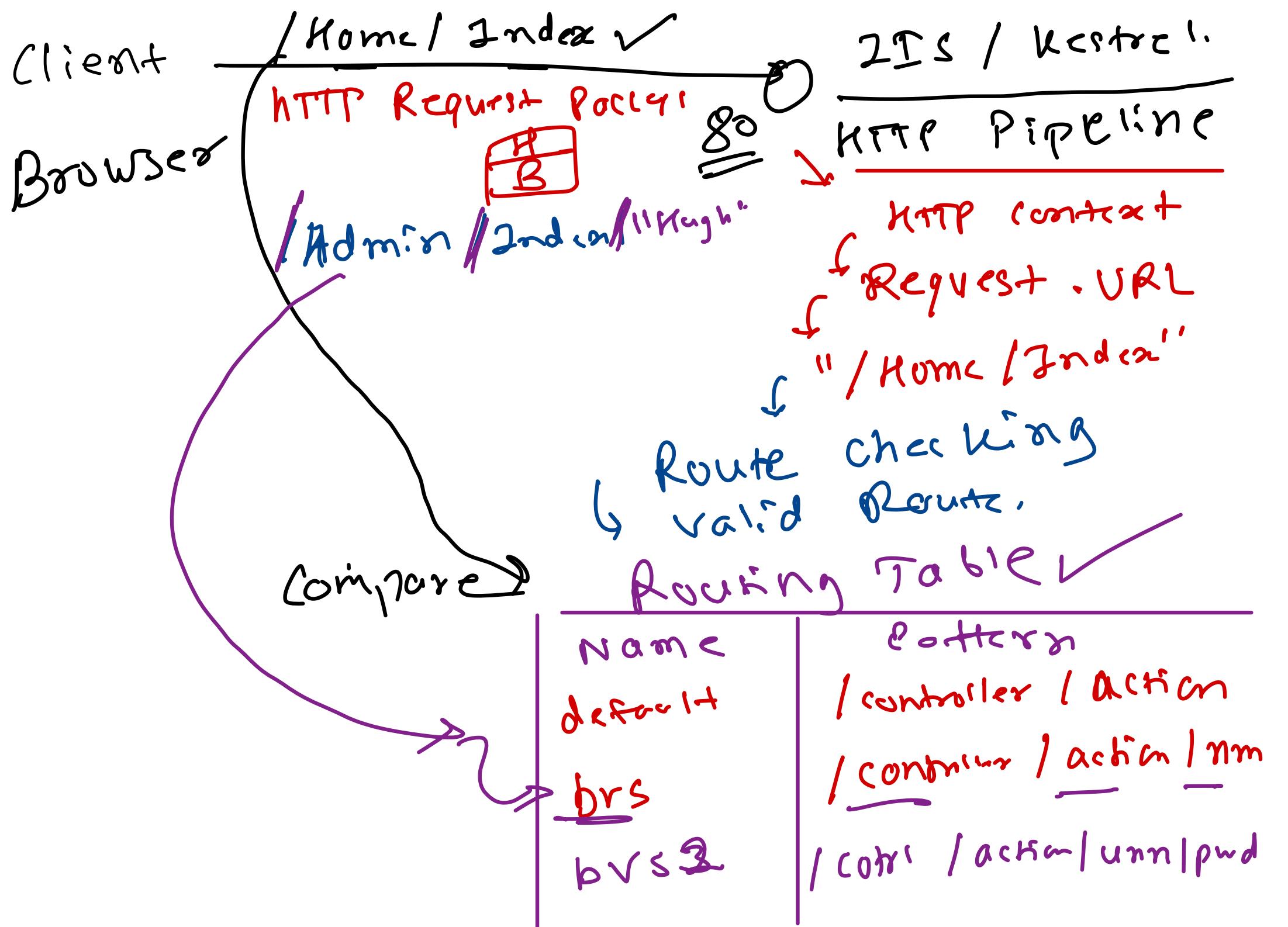
Model — view — Controller

Architectural Based Design
Pattern

based on this pattern Microsoft
has developed Asp.NET MVC Core
framework.

M — V — C





client
Browser

/Home/Index → IIS / IIS

1. HTTP Pipeline
2. Route Collection Table map.

name	pattern
defaults	/controller / action
brs1	/controller / action / id
brs2	/ctrl / act / nm1 / pwd
brs3	/ctrl / act / number

3. If URL matches with Route ^{for} _{place}

URL . Split () → { "Home", "Index" }

o/p → URL = { "Home", "Index" }

4. Load CCR

5. Microsoft.AspNetCore.Mvc.dll

* Load

6. →

ControllerFactory factory = new ControllerFactory;

Controller controller = factory.GetController("Home Controller")

↓
{ ← controller. InvokeMember()
 | method: "Index"
 | parameters: new object[]

 | → Execute →

Controller

.

→

index Execute()

return View("Index")
 .cshtml
 Razor @
 wefor <%> ?
 csharp
 html

① find Index.cshtml file.

File Exist("~/views/Kernel/Index.cshtml")
 → Syntax @ → Razor View Engine

ViewEngineFactory fac =
 new ViewEngineFactory()
 IViewEngine ve = fac.GetViewEngine("Razor")
 ve.Render()

read file Index.cshtml

HTML Template

<html>
 :<table>
 <tr>
 <td>Through Chart
 </td>
 </table>

call CLR
 List Comp
 .Count = 5

ViewEngine + CLR Create
 blank

```

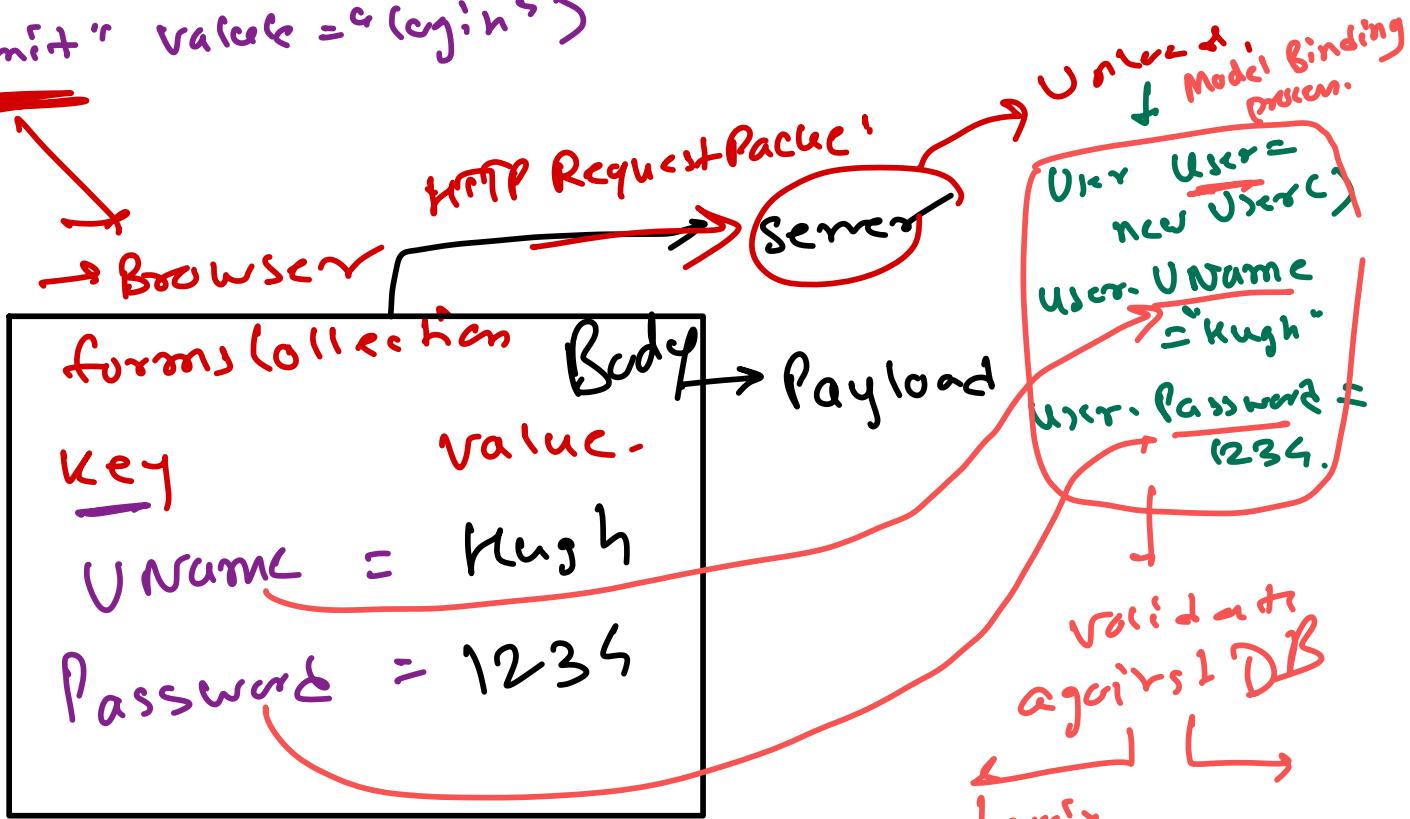
<form action="fHome/GetNewUserDetails"
      method="post">
  <label for="UName" >UName </label>
  <input type="text" name="UName" />
  <label for="Password" >Password </label>
  <input type="password" name="Password" />
  <input type="submit" value="Login" />
</form>

```

UName: Kugh

Password: 1234

Login



```

class User {
  prop string UName
  prop string Password
}

```

```
public ActionResult ValidateUser(User user)  
{  
    if (user.UName == "Kugh" &&  
        user.Password == "1234")  
    {  
        return Log on  
    }  
}
```

Made(Binder -