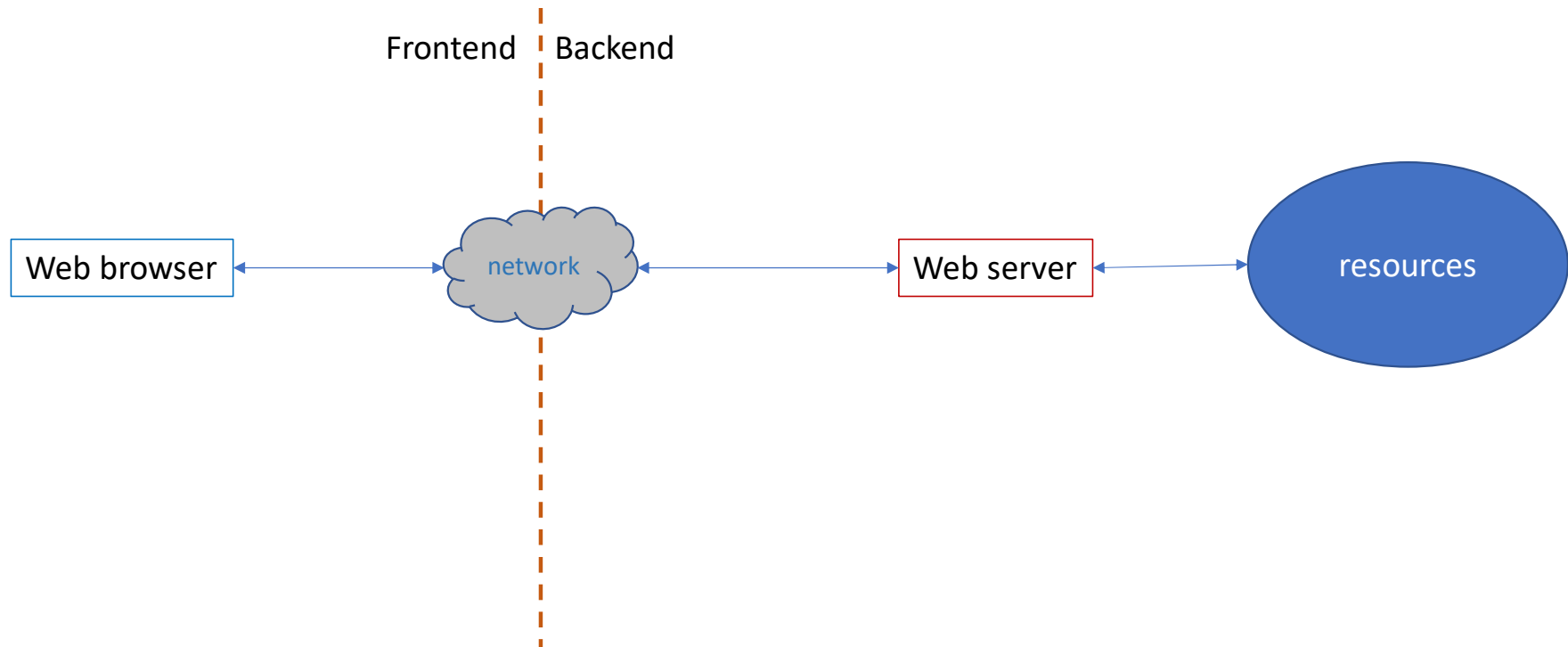


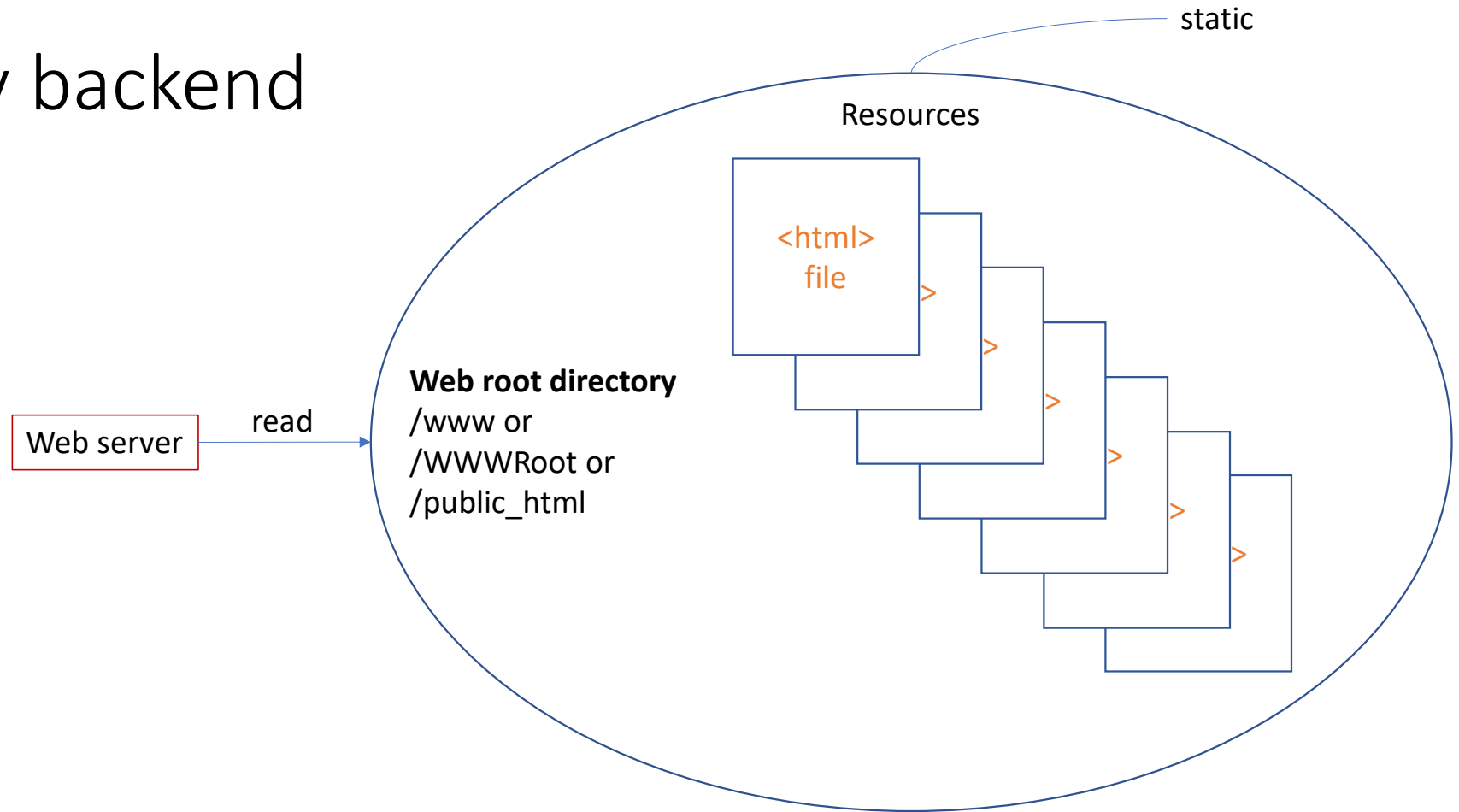
Backend

History & ASP.Net Core Introduction

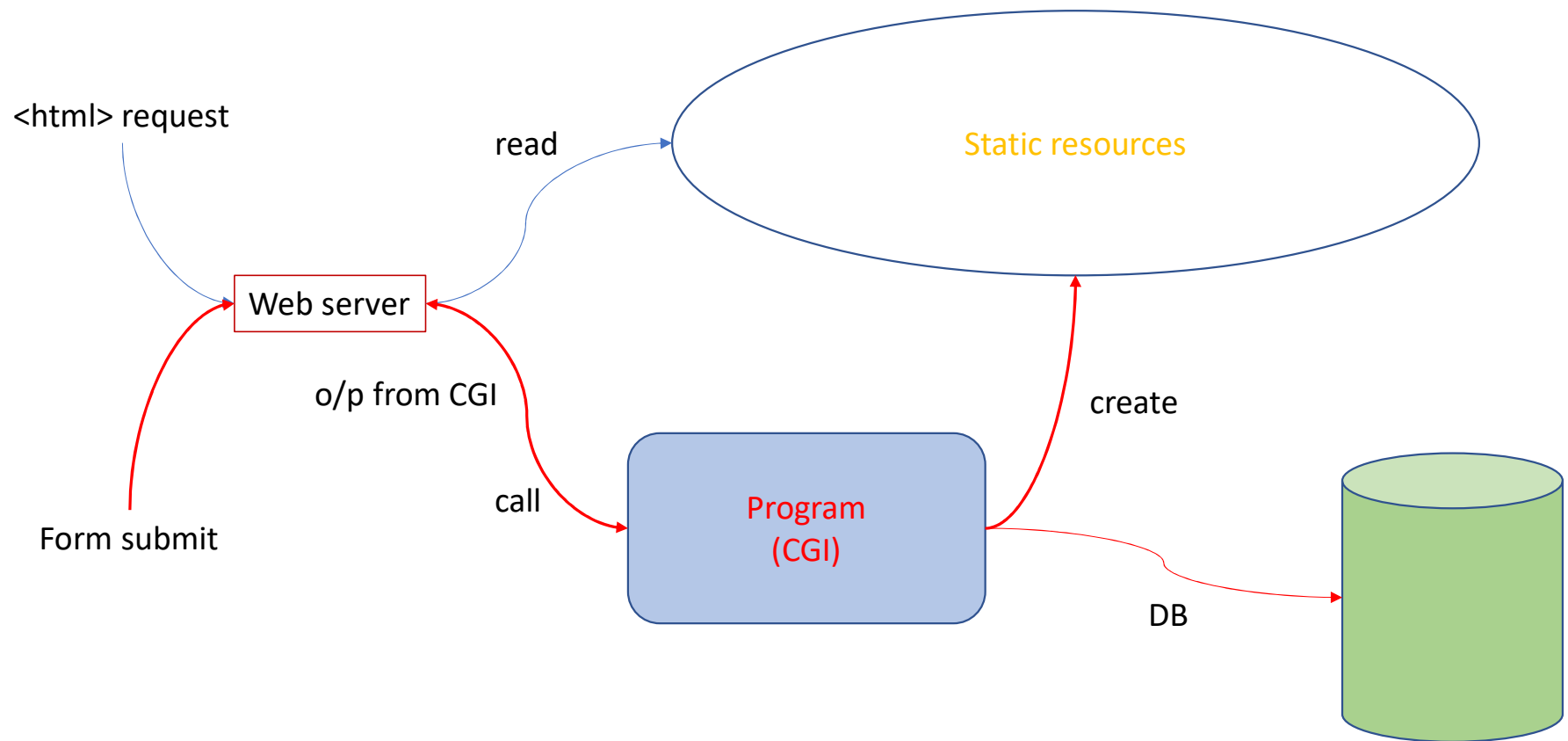
Web in general



Early backend

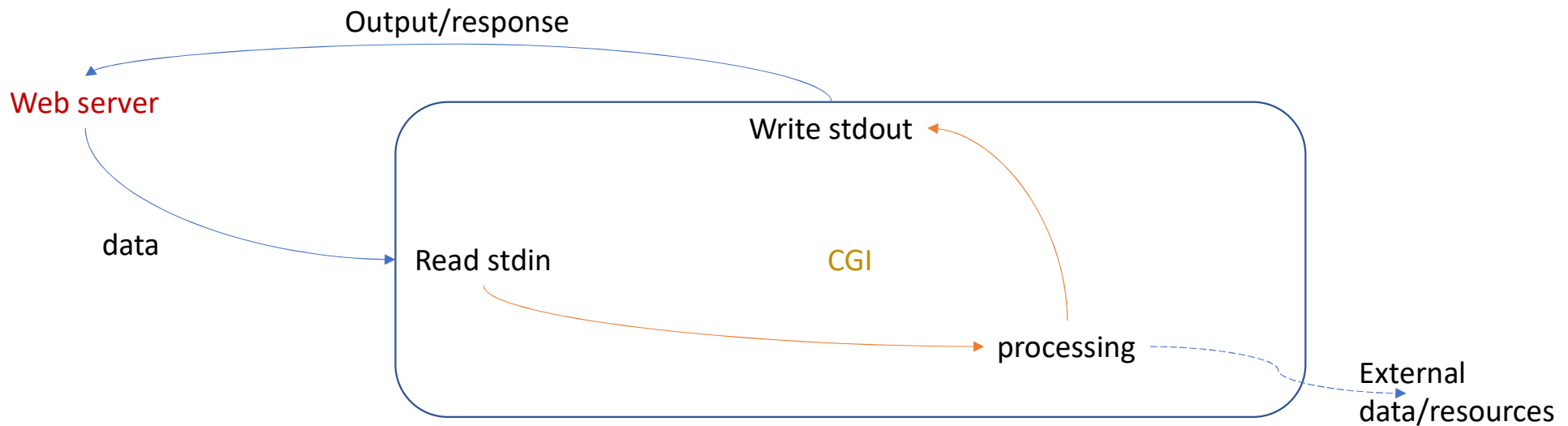


Early backend with data handling

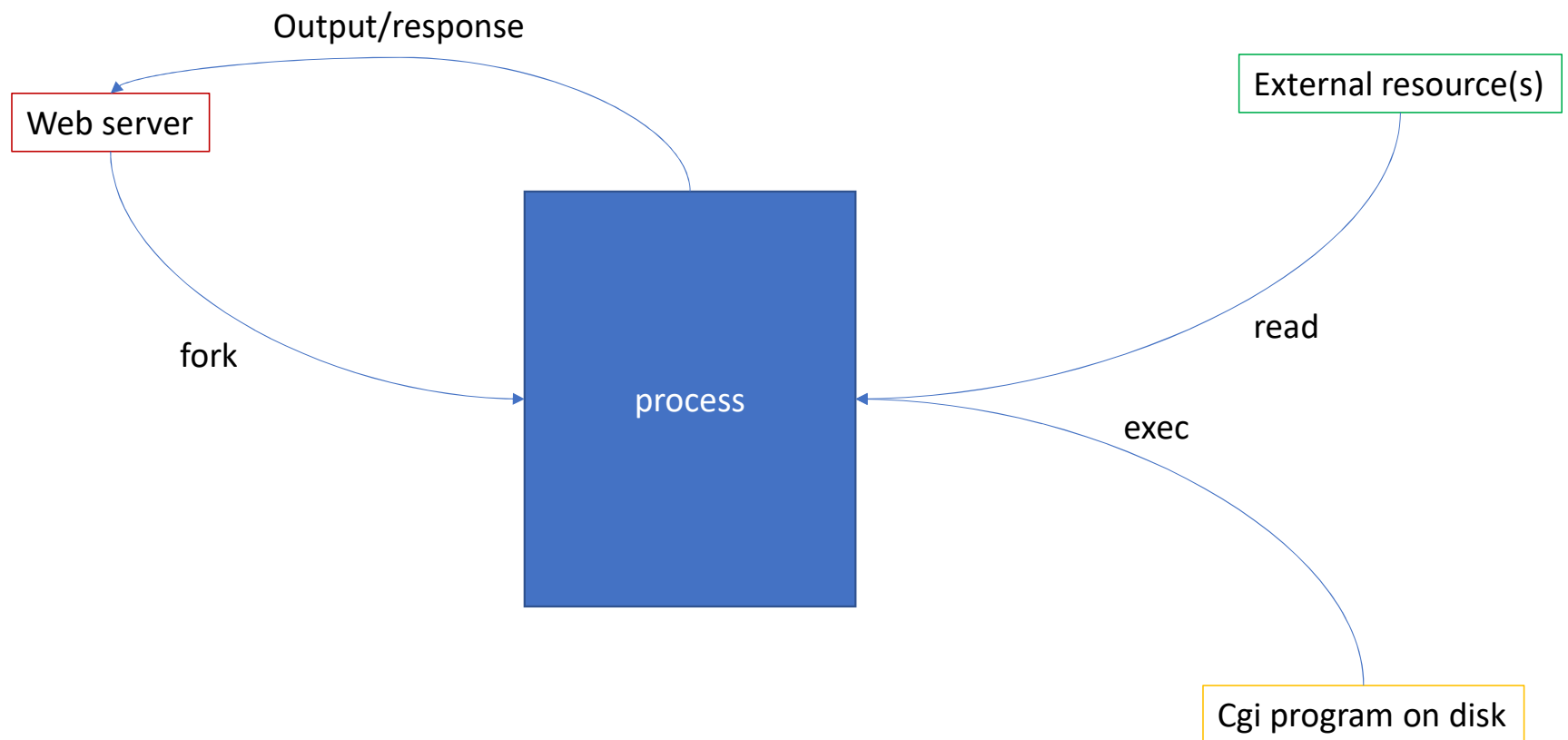


Early CGI

- Develop with language such as C/C++, perl, etc.
- Data transfer from web server to CGI via standard input
- Data transfer from CGI to web server via standard output



Execution



Example: CGI in C

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

char *unescstring(char *, int, char *, int);
```

```
char *cgigetval(char *fieldname)
```

```
{
    int fnamelen;
    char *p, *p2, *p3;
    int len1, len2;
    static char *querystring = NULL;
    if(querystring == NULL)
    {
        querystring = getenv("QUERY_STRING");
        if(querystring == NULL)
            return NULL;
    }
}
```

```
if(fieldname == NULL)
    return NULL;
```

```
fnamelen = strlen(fieldname);
```

```
for(p = querystring; *p != '\0';)
{
    p2 = strchr(p, '=');
    p3 = strchr(p, '&');
    if(p3 != NULL)
        len2 = p3 - p;
    else
        len2 = strlen(p);

    if(p2 == NULL || p3 != NULL && p2 > p3)
    {
        /* no = present in this field */
        p3 += len2;
        continue;
    }
    len1 = p2 - p;

    if(len1 == fnamelen && strncmp(fieldname, p, len1) == 0)
    {
        /* found it */
        int retlen = len2 - len1 - 1;
        char *retbuf = malloc(retlen + 1);
        if(retbuf == NULL)
            return NULL;
        unescstring(p2 + 1, retlen, retbuf, retlen+1);
        return retbuf;
    }

    p += len2;
    if(*p == '&')
        p++;
}

/* never found it */

return NULL;
}
```

Ref:<https://www.eskimo.com/~scs/cclass/handouts/cgi.html>

CGI in C (output)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

extern char *cgetval(char *);

main()
{
    char *browser;
    char *editttype;
    char *text;

    printf("Content-Type: text/html\n\n");

    printf("<html>\n");
    printf("<head>\n");
    printf("<title>CGI test result</title>\n");
    printf("</head>\n");
    printf("<body>\n");

    browser = getenv("HTTP_USER_AGENT");

    printf("<p>\n");
    printf("Hello, ");
    if(browser != NULL && strstr(browser, "Lynx") != NULL)
        printf("Lynx");
    else if(browser != NULL && strstr(browser, "Mosaic") != NULL)
        printf("Mosaic");
    else if(browser != NULL && strstr(browser, "Mozilla") != NULL)
        printf("Netscape");
    else printf("unknown browser");
    printf(" user!\n");
```

```
        printf("<p>\n");

        printf("Here is your modified text:\n");
        printf("<br>\n");

        text = cgetval("textfield");
        editttype = cgetval("editttype");

        if(text == NULL)
        {
            printf("You didn't enter any text!\n");
        }
        else if(editttype != NULL && strcmp(editttype, "reverse") == 0)
        {
            reverse(text);
            printf("%s\n", text);
        }
        else if(editttype != NULL && strcmp(editttype, "upper") == 0)
        {
            upperstring(text);
            printf("%s\n", text);
        }
        else if(editttype != NULL && strcmp(editttype, "lower") == 0)
        {
            lowerstring(text);
            printf("%s\n", text);
        }
        else
        {
            printf("You didn't select a transformation!\n");
        }

        printf("</body>\n");
        printf("</html>\n");
    }
}
```

Ref:<https://www.eskimo.com/~scs/cclass/handouts/cgi.html>

PHP (ASP)

- Easier data input handling
 - Automatic generate \$_GET or \$_POST
- Easier coding
 - Coding in html file
 - <?php ?>
- Integrates PHP interpreter into web server (option)
 - Speed up execution

PHP Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>PHP Form</title>
</head>
<body>
  <main>
    <?php if ($_SERVER['REQUEST_METHOD'] === 'GET') : ?>
      <form action="<?php htmlspecialchars($_SERVER['PHP_SELF']) ?>" method="post">
        <div>
          <label for="name">Name:</label>
          <input type="text" name="name" required="required" placeholder="Enter your name" />
        </div>

        <div>
          <label for="name">Email:</label>
          <input type="email" name="email" required="required" placeholder="Enter your email" />
        </div>

        <button type="submit">Subscribe</button>
      </form>
    <?php else : ?>
```



Ref:<https://www.phptutorial.net/php-tutorial/php-form/>

PHP Example


```
<?php
//-----
// WARNING: this doesn't include sanitization
// and validation
//-----
if (isset($_POST['name'], $_POST['email'])) {
    $name = htmlspecialchars($_POST['name']);
    $email = htmlspecialchars($_POST['email']);

    // show the $name and $email
    echo "Thanks $name for your subscription.<br>";
    echo "Please confirm it in your inbox of the email $email.";
} else {
    echo 'You need to provide your name and email address.';
}

?>

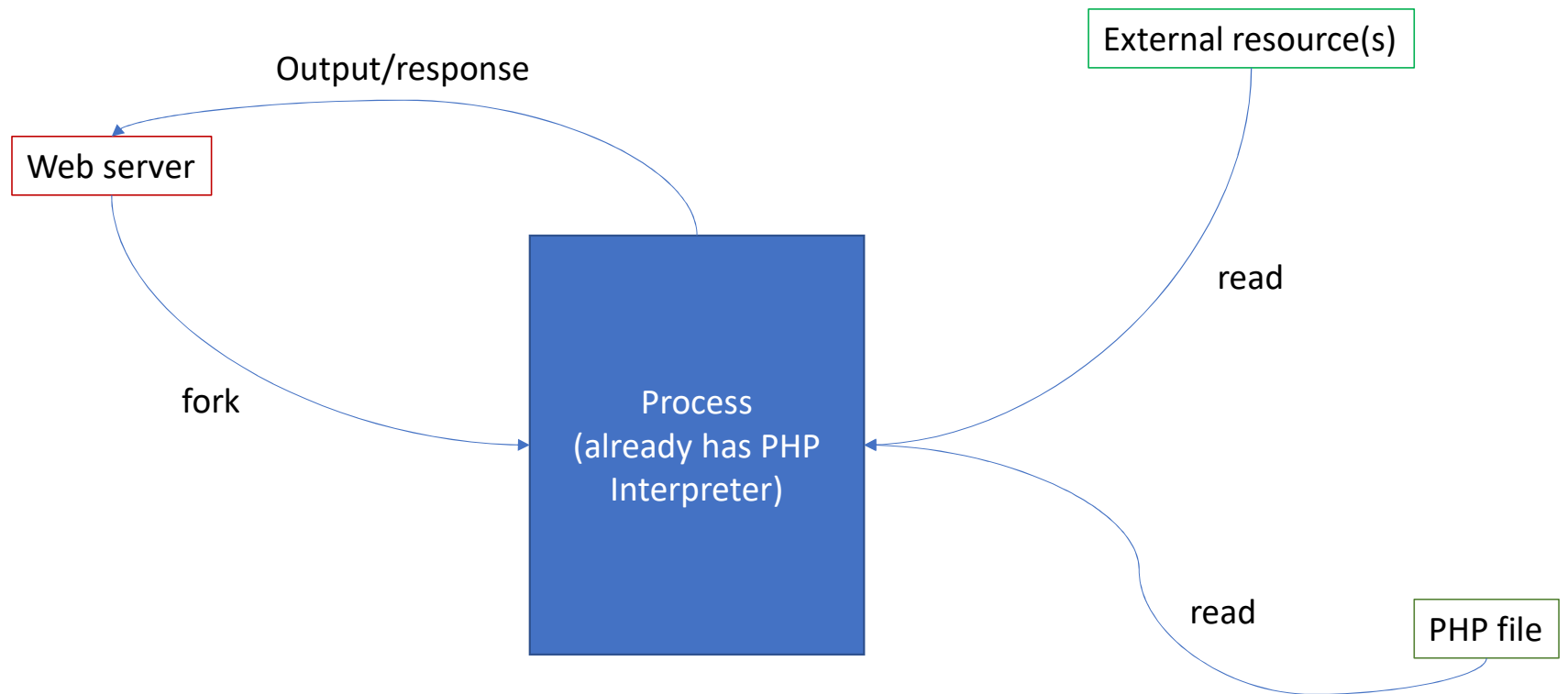
<?php endif ?>
</main>
</body>

</html>
```



Ref:<https://www.phptutorial.net/php-tutorial/php-form/>

Execution



ASP

- Classic ASP
- Active Server Page
- Developed by Microsoft
- Same concepts as PHP
- ASP command written within `<% ... %>`
- Based on VBScript syntax

Example: ASP code

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<p>ASP example: output HTML tag</p>
```

```
<%  
response.write("<h1>Hello World!</h1>")  
%>
```

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

Example: what browser receive

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>ASP example: output HTML tag</p>
```

```
<h1>Hello World!</h1>
```

```
</body>
```

```
</html>
```

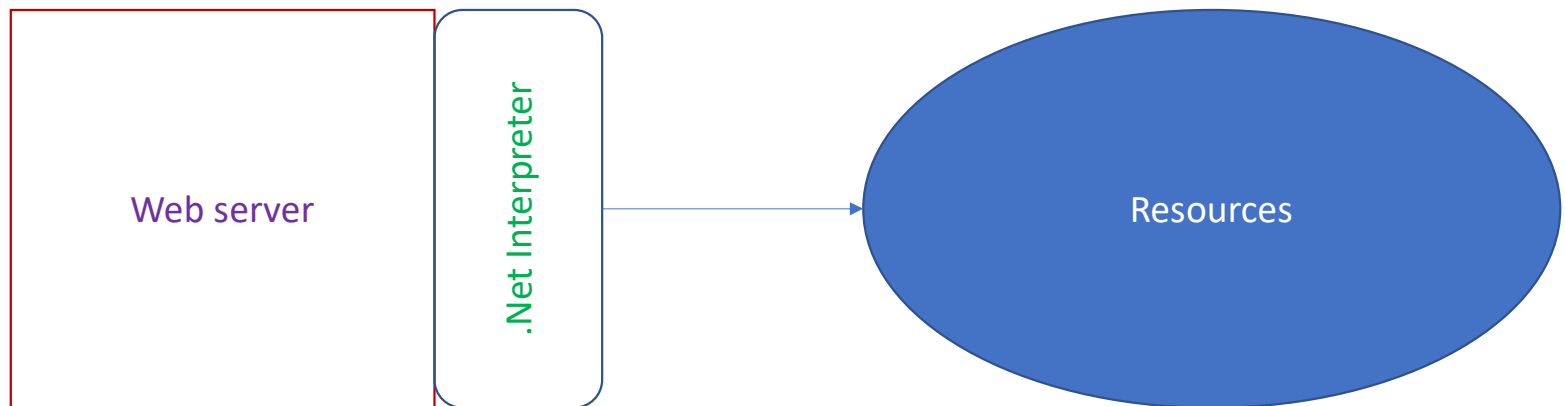
ASP.Net

- Release in 2002 as a successor to ASP
- Normally used c# syntax
- Uses .aspx as an extension
- Tutorial: https://www.w3schools.com/asp/webpages_intro.asp

Example:

```
<!DOCTYPE html>
<html>
<body>
@{
if (IsPost)
{
string companyname = Request["CompanyName"];
string contactname = Request["ContactName"];
<p>You entered: <br>
Company Name: @companyname <br>
Contact Name: @contactname </p>
}
else
{
<form method="post" action="">
Company Name:<br>
<input type="text" name="CompanyName" value=""><br>
Contact Name:<br><br>
<input type="text" name="ContactName" value=""><br><br>
<input type="submit" value="Submit" class="submit">
</form>
}
}
</body>
</html>
```

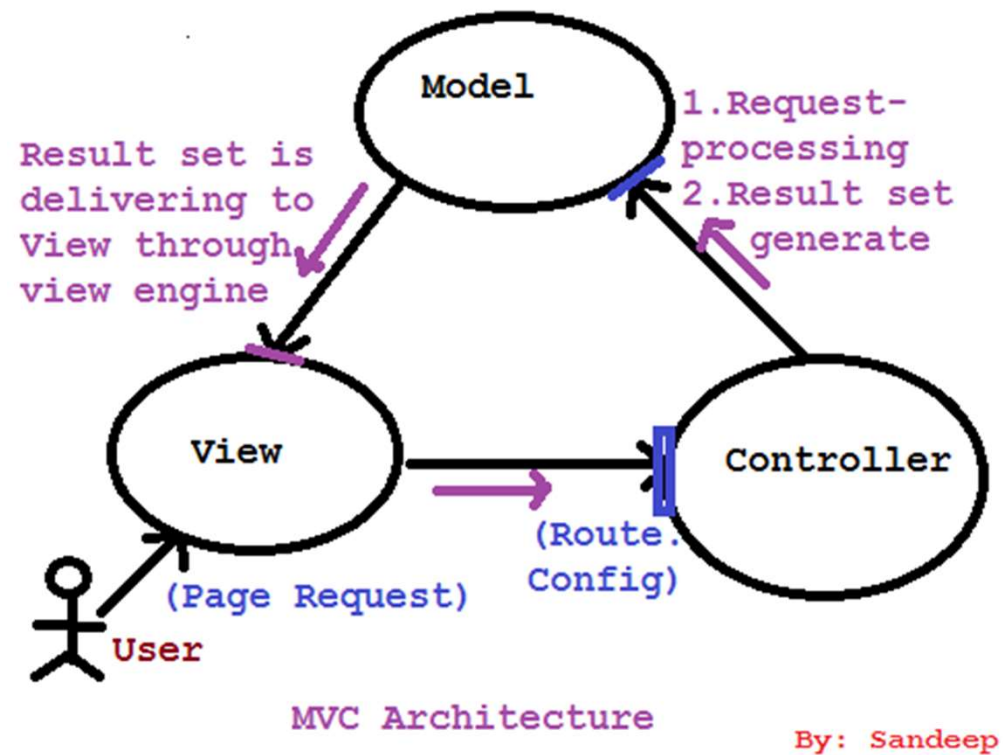
ASP/ASP.Net



ASP.Net MVC → ASP.Net Core MVC (2016)

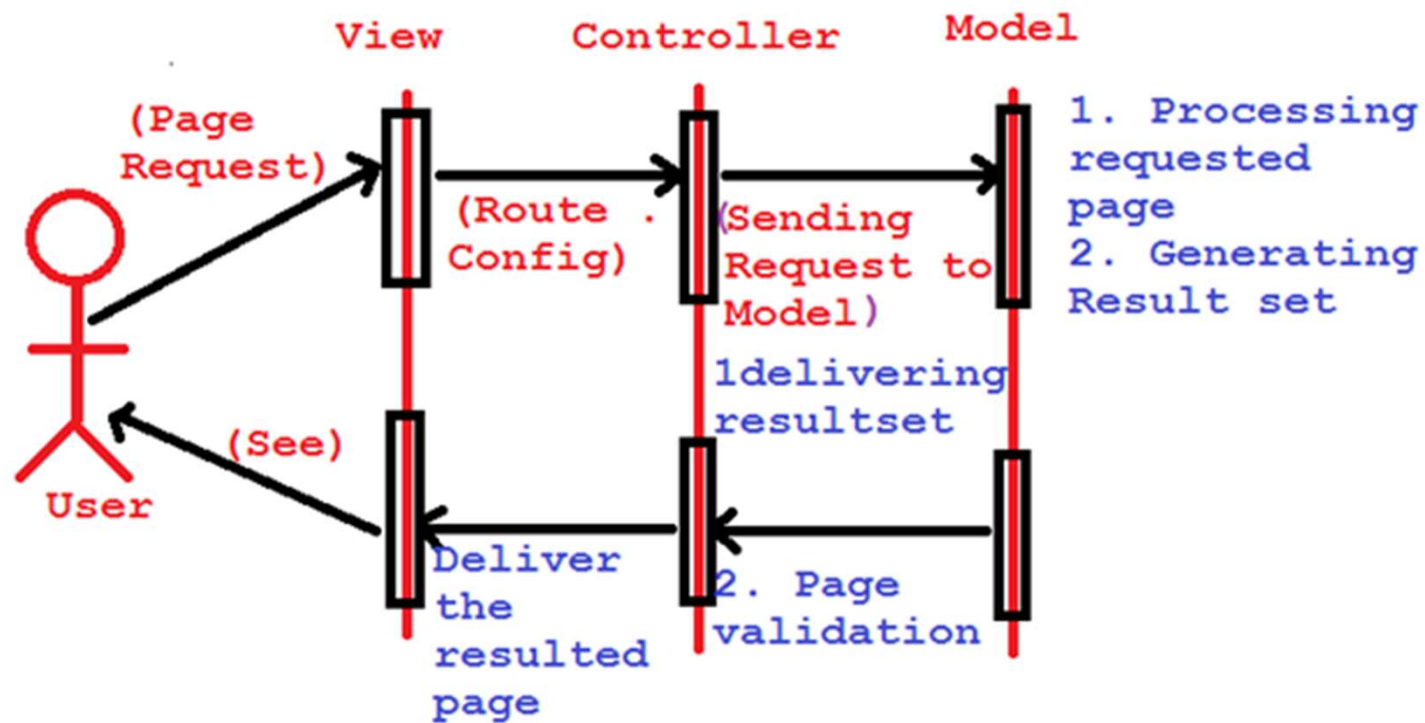
- MVC = Model View Controller
 - Model represent data
 - View is UI
 - Controller is the request handler

MVC Architecture



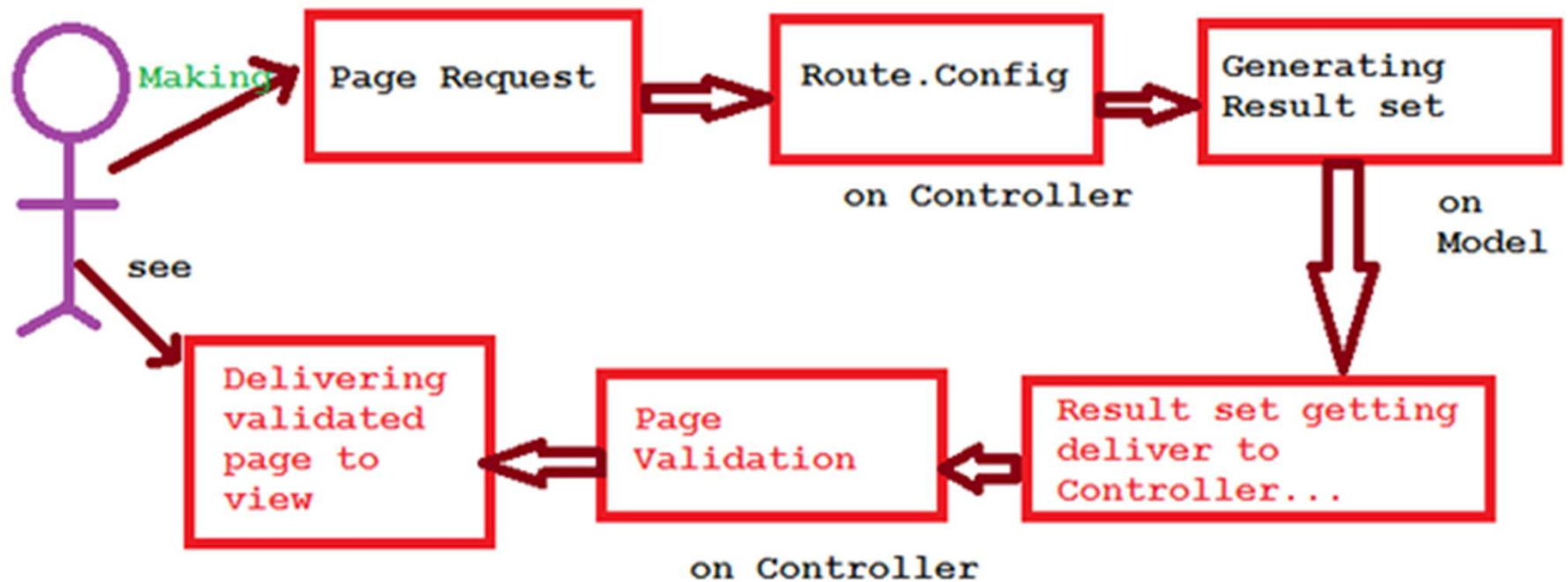
Ref: <https://www.c-sharpcorner.com/UploadFile/18585c/mvc-architecture-mvc-life-cycle/>

Life cycle of MVC



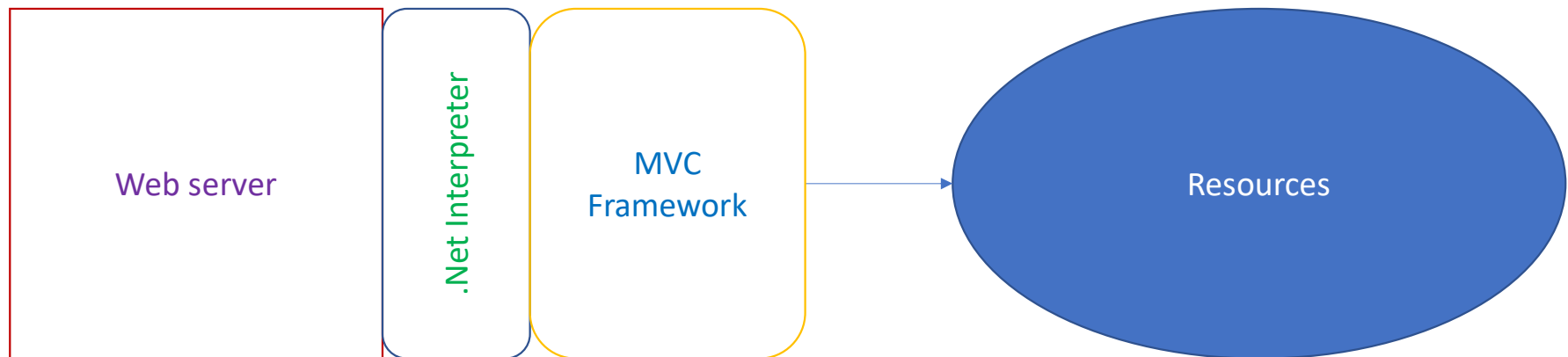
Ref: <https://www.c-sharpcorner.com/UploadFile/18585c/mvc-architecture-mvc-life-cycle/>

Live cycle of MVC



Ref: <https://www.c-sharpcorner.com/UploadFile/18585c/mvc-architecture-mvc-life-cycle/>

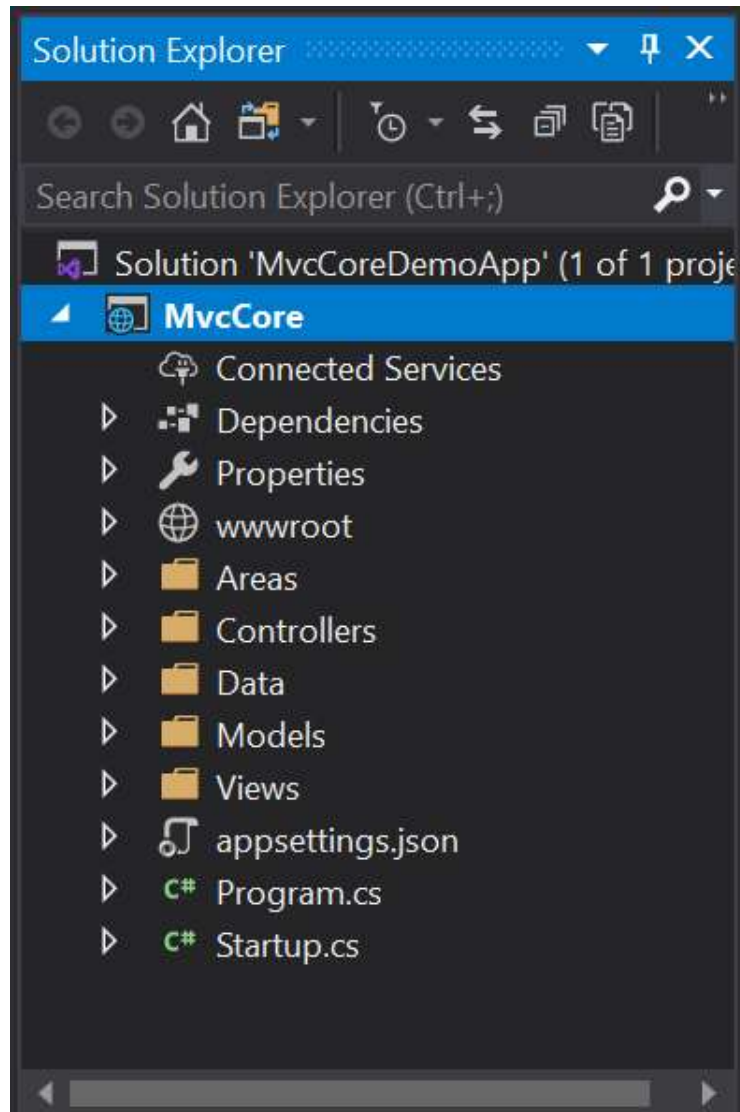
ASP.Net Core MVC



ASP.Net Core MVC Tutorial

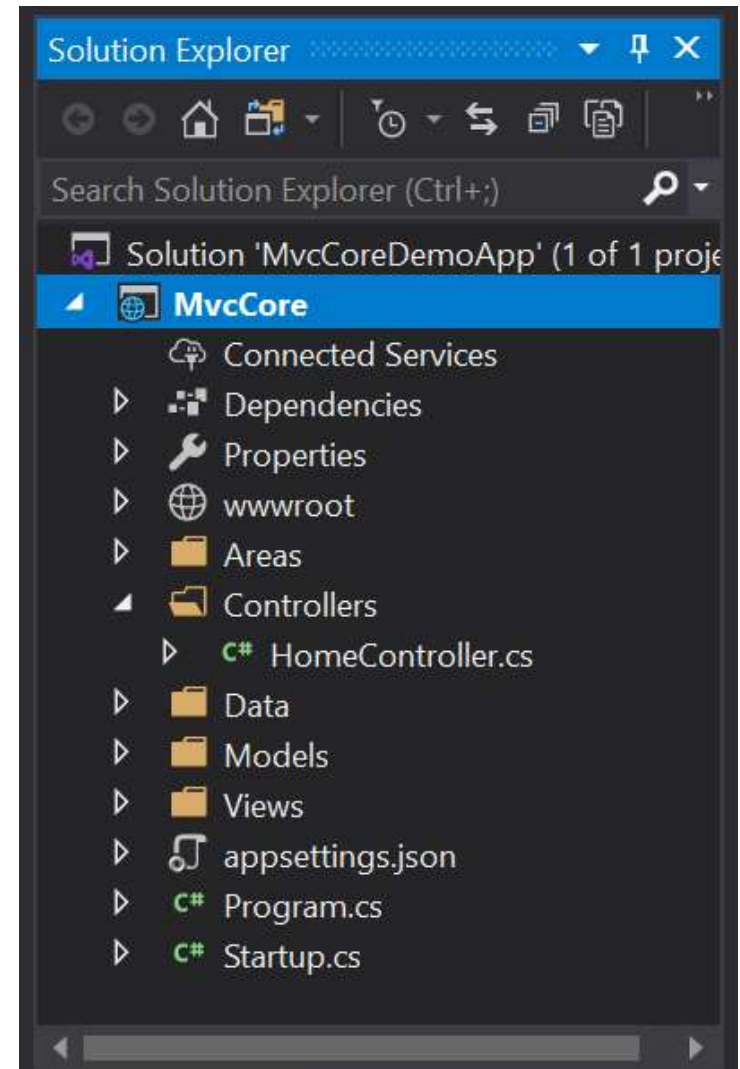
- VDO Clip: ASP.Net Core Introduction:
<https://www.youtube.com/watch?v=1ck9LIBxO14>
- ASP.Net Core Web Tutorial:
 - <https://asp.mvc-tutorial.com/>
 - <https://www.tektutorialshub.com/asp-net-core-tutorial/>

Project Structure



Controller

- Routing to resource
 - From picture there is 1 (virtual) directory in this web site
 - Named: Home
 - `http://host/home`
- In `HomeController.cs` contain methods for rendering pages
 - Each method correspond to 1 page



HomeController.cs

0 references

```
public IActionResult Index()  
{  
    return View();  
}
```

<http://host/Home> or <http://host/Home/index>

0 references

```
public IActionResult Test()  
{  
    return View();  
}
```

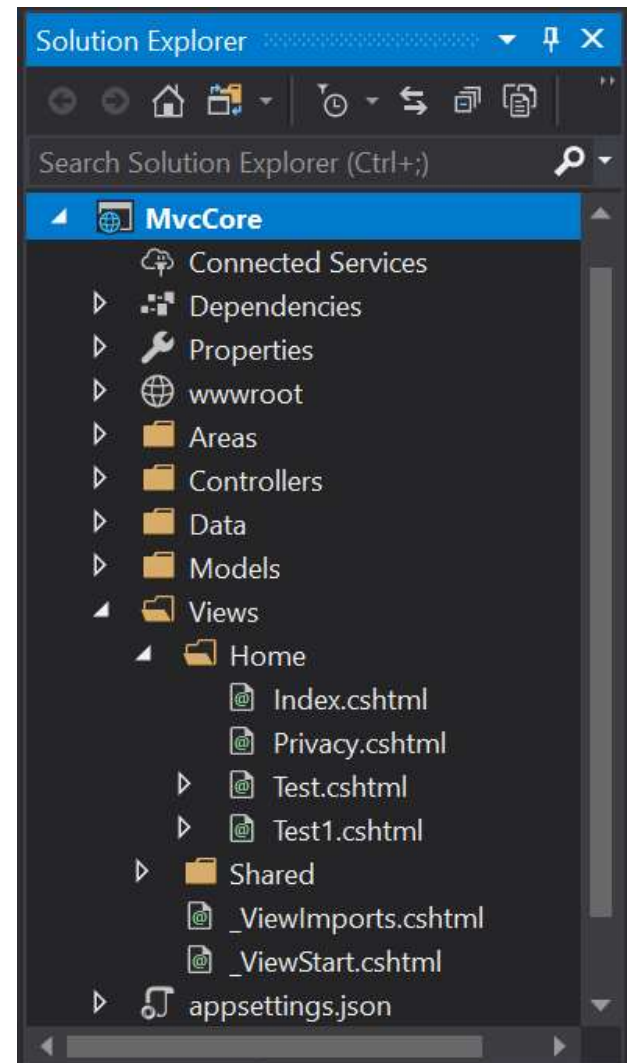
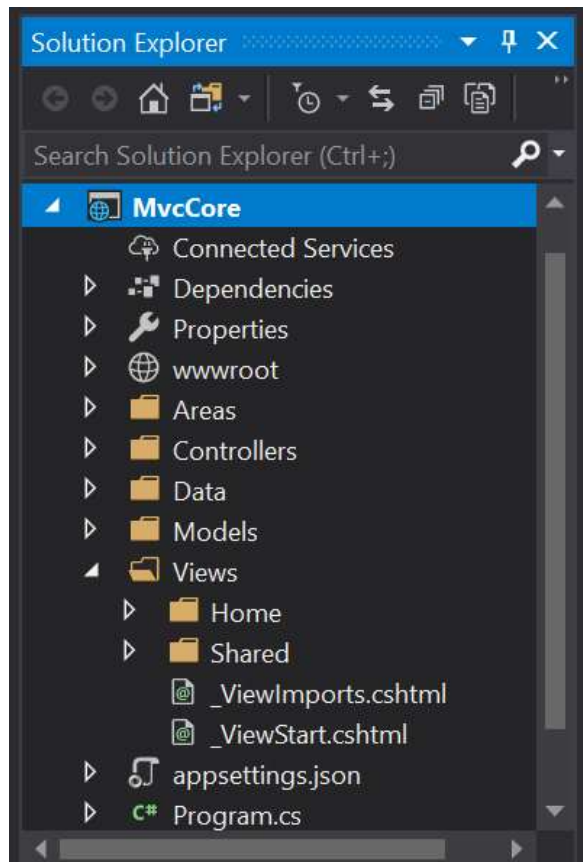
<http://host/Home/Test>

0 references

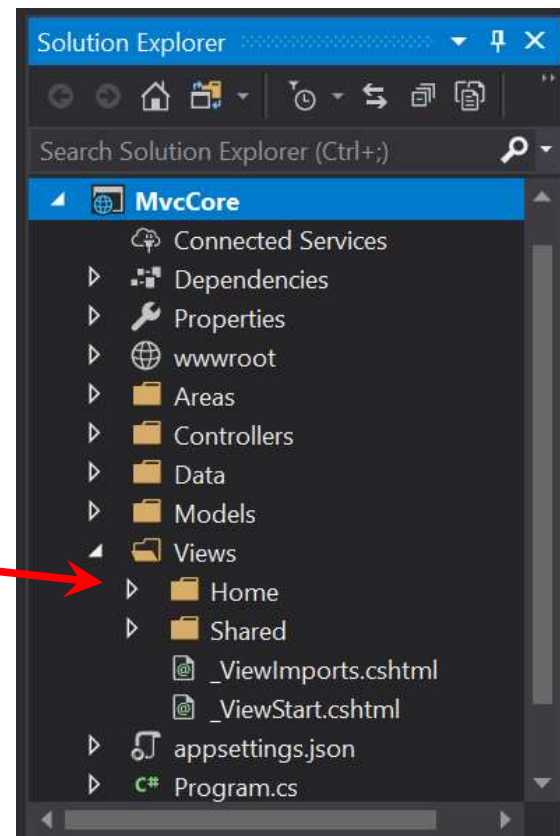
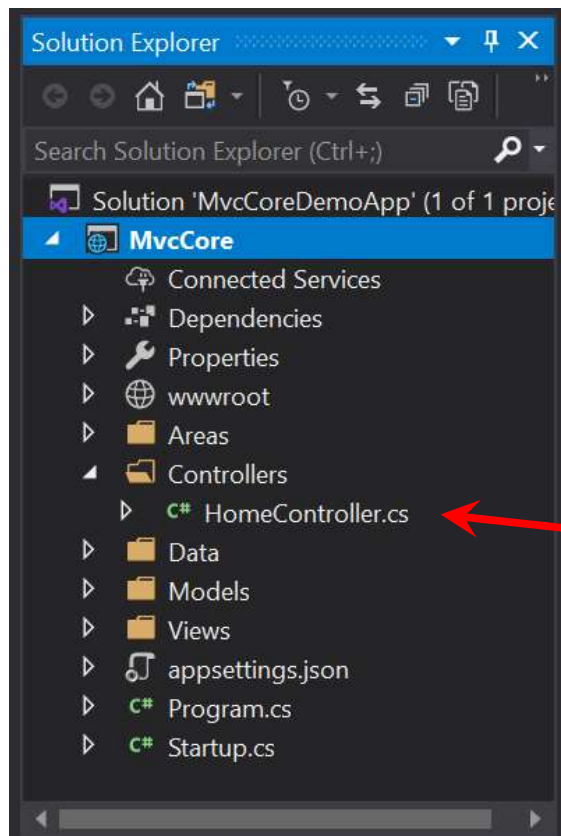
```
public IActionResult Test1()  
{  
    return View();  
}
```

<http://host/Home/Test1>

View

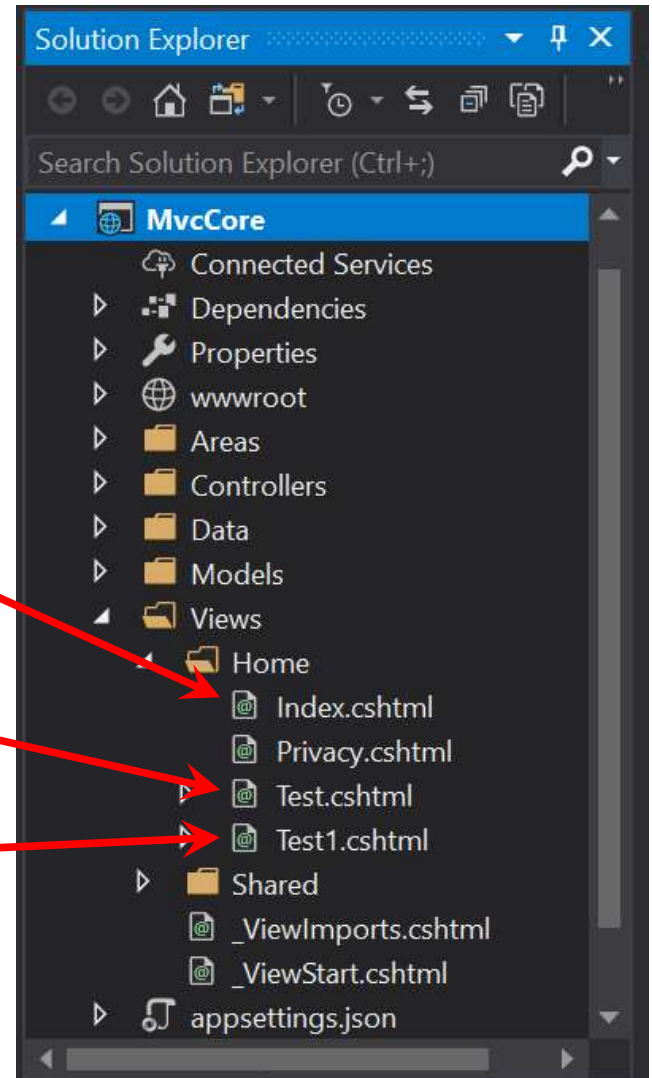


Controller - View



Controller - View

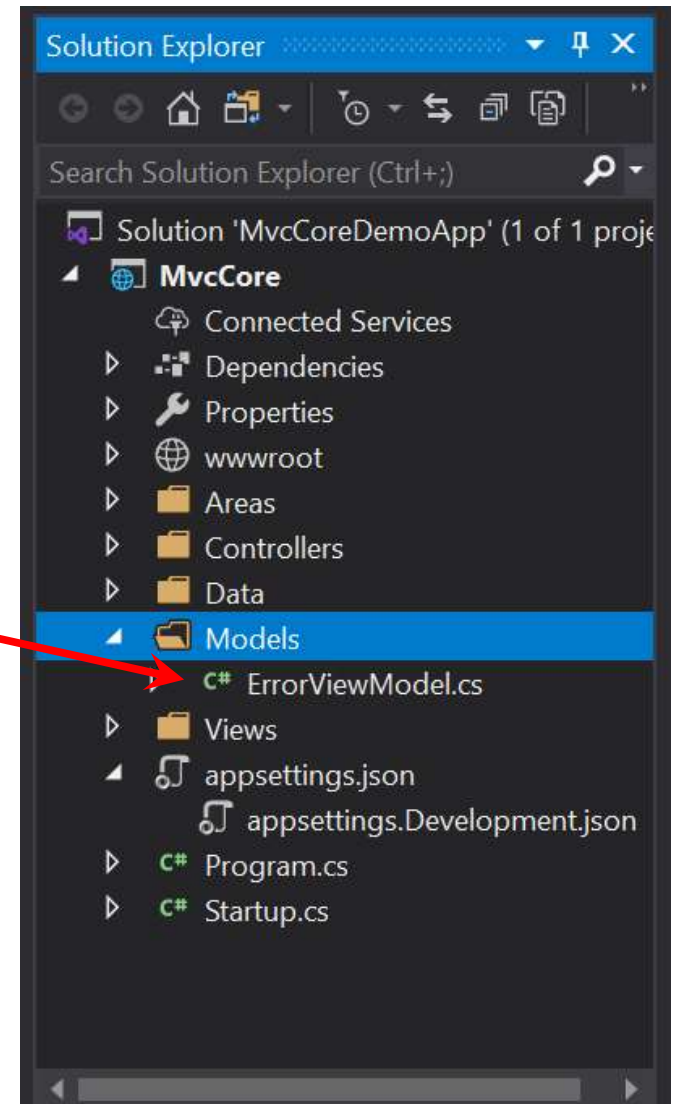
```
0 references  
public IActionResult Index()  
{  
    return View();  
}  
  
0 references  
public IActionResult Test()  
{  
    return View();  
}  
  
0 references  
public IActionResult Test1()  
{  
    return View();  
}
```



Model

```
namespace MvcCore.Models
{
    4 references
    public class ErrorViewModel
    {
        3 references
        public string RequestId { get; set; }

        1 reference
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}
```



Controller-Model-View

```
public IActionResult Error()  
{  
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });  
}
```

HomeController.cs (Controller)

```
<h1 class="text-danger">Error.</h1>  
<h2 class="text-danger">An error occurred while processing your request.</h2>  
  
<@if (Model.ShowRequestId)>  
<div>  
    <p>  
        <strong>Request ID:</strong> <code>@Model.RequestId</code>  
    </p>  
</div>  
</@if>
```

Error.cshtml (View)

```
namespace MvcCore.Models  
{  
    4 references  
    public class ErrorViewModel  
    {  
        3 references  
        public string RequestId { get; set; }  
  
        1 reference  
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);  
    }  
}
```

ErrorViewModel.cs (Model)