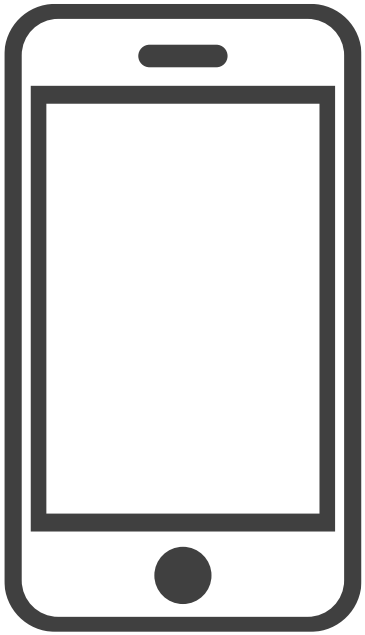
ASP.NET CORE

ACTION RESULTS

issntt@nus.edu.sg

Problem



?

Client

Server

How can our web apps **generate** appropriate **responses** to clients?

Objectives

At the end of this lesson, students will be able to

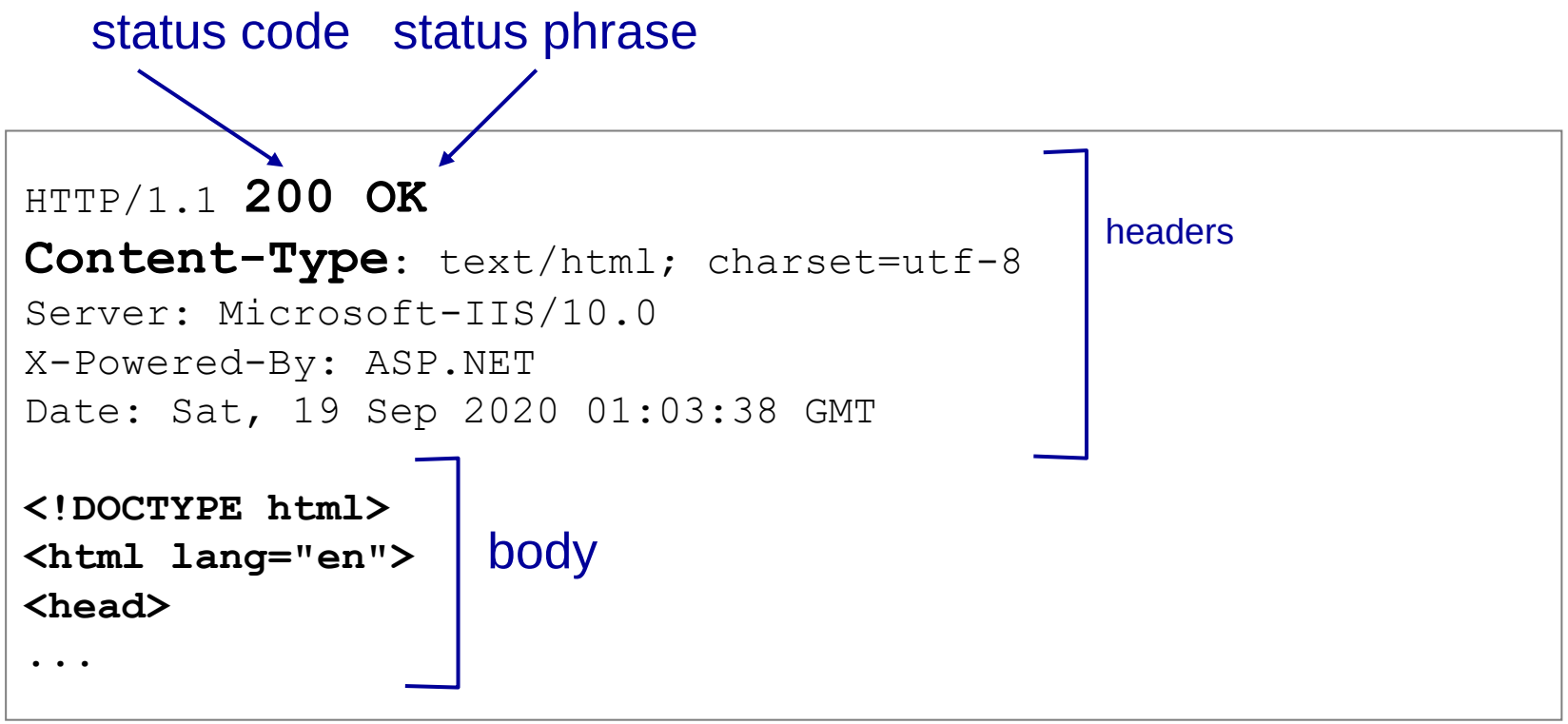
- Describe different parts of an HTTP Response and their functionalities
- Describe the roles of *IActionResult* interface and *ActionResult* class in ASP.NET Core web app development
- Design and implement ASP.NET Core web app that generates HTTP Responses on different scenarios

Topics

- **HTTP Response**
  - **Status code and status phrase**
  - **Body**
  - **Content Type**
- Generating HTTP Response with ASP.NET Core
- Action Results

HTTP Response

Following is a sample response when Browsers access http://localhost:53440/Home/Index



https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview#HTTP_Messages

Status Code

Status codes **generally** indicate if the request was **successful**, and/or **why**

Status Code	Outcomes
200 (OK)	Operation is **successful**
201 (Created)	Resource has been **created successfully**
302 (Found)	**Redirection** (**new URL** is in Location header)
400 (Bad Request)	Server **does not understand** client's request
401 (Unauthorized)	Server does not know **who you are**
403 (Forbidden)	Server knows you but you are not **allowed** to access the resource
404 (Not Found)	Resource is **not found** in the server
500 (Internal Server Error)	Generic response when your **server encounters** an **exception**

Body

A body is **optional**, containing the **data** of the respective fetched resource

http://localhost:53440/sample.html

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/10.0
Content-Type: text/html

<html>
<body>
<h1>Hello, DipSA!</h1>
<p>How are you today?</p>
</body>
</html>
```

http://localhost:53440/sample.css

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/10.0
Content-Type: text/css

body {
  background-color: white;
}
h1 {
  font-size: 18px;
}
```

In what situations may server **not** need to send response body?

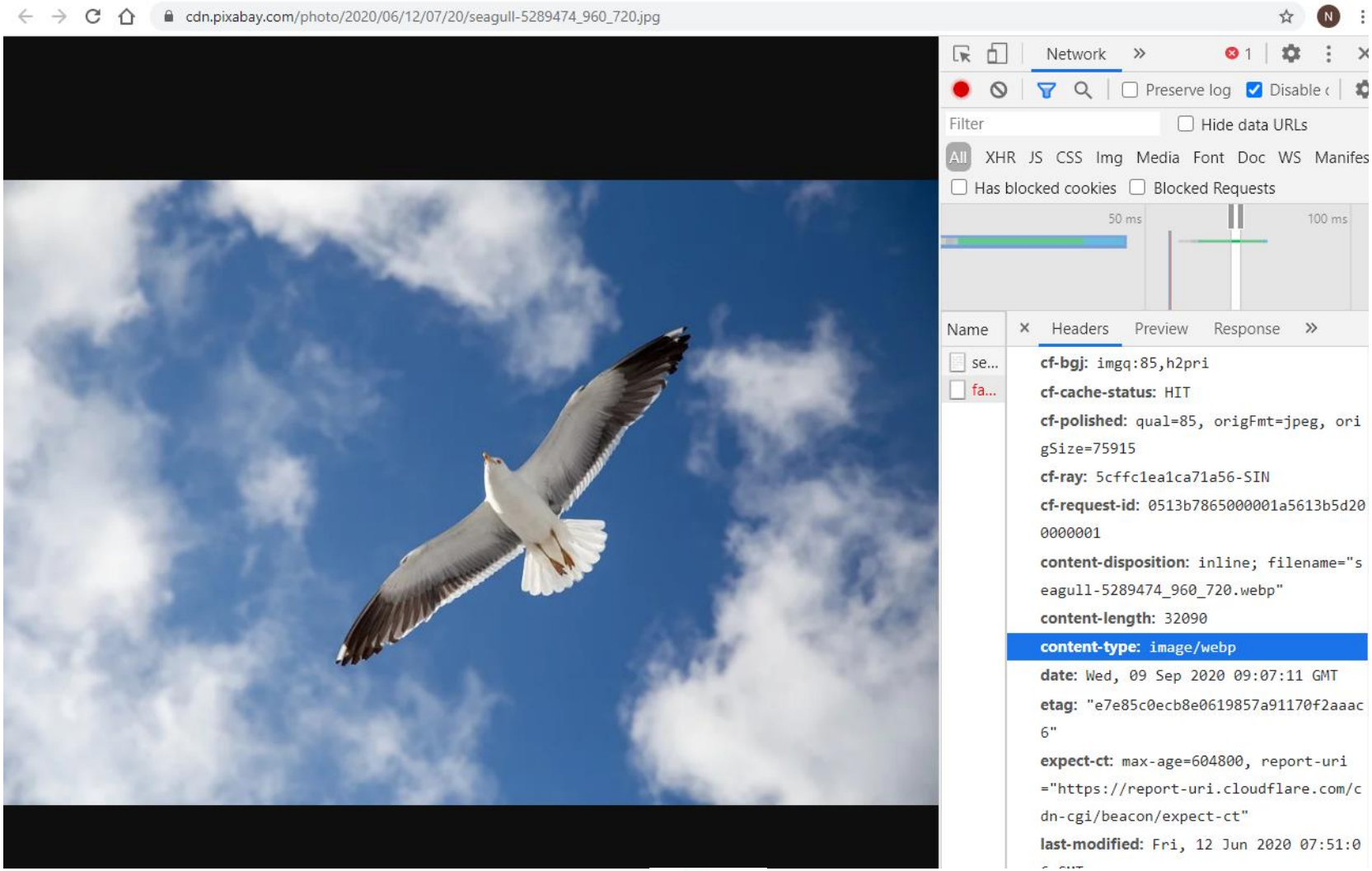
Content Type

Content type is among the **headers,** and it helps clients know **how to display** the data in the body

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/10.0
Content-Type: text/html

<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Common Content Type
text/plain
text/html
image/gif
image/jpeg
audio/mpeg
video/mpeg
application/msword
application/pdf

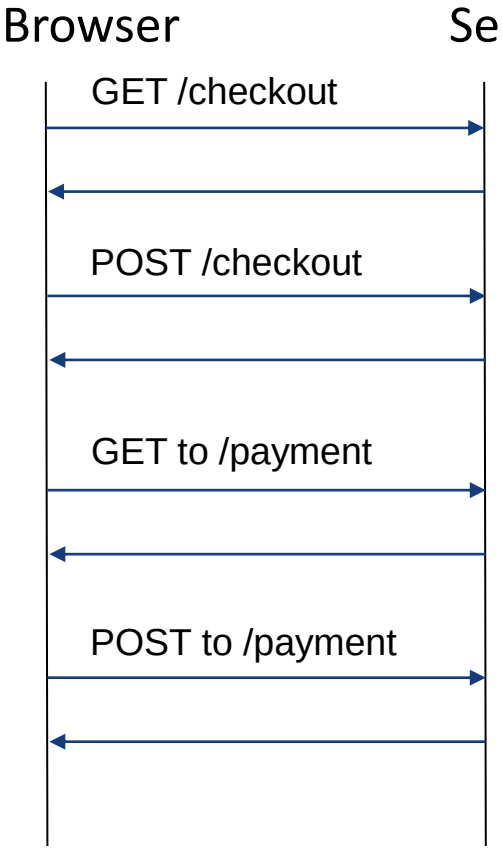
Content Type



What if the server sends the **incorrect** content type?

Quiz

Consider a common scenario in ecommerce web apps. For each HTTP Response, what will be **status code**, **body**, and **content type?**



1. Users navigate to checkout page

1. Generate an **HTML checkout page** and return it to the browser

2. Users click Buy button

2. Begin the checkout process by sending a **redirect response** to the payment page

3. Browser automatically redirect to the payment page

3. Generate an **HTML page for payment** and return it to the browser

4. Users fills in and submit the payment form

4. Process the payment and send a **redirect response** the order complete page

Next

Our server needs to **generate** an **HTTP Response** with following requirements:

- **Status code and phrase**: 200 OK
- **Content type**: text/plain
- **Body**: Hello World

How can we do that with ASP.NET Core?

Topics

- HTTP Response
- **Generating HTTP Response with ASP.NET Core**
- Action Results

First Attempt

"**Manually**" generate the **whole** HTTP Response

```
public static string GenerateResponse()
{
    return @"
        HTTP/1.1 200 OK
        Content-Length: 11
        Content-Type: text/plain; charset=utf-8
        Server: Microsoft-IIS/10.0
        X-Powered-By: ASP.NET
        Date: Sat, 19 Sep 2020 13:58:01 GMT

        Hello World
    ";
}
```

This code was used before .NET Core and therefore NOT our focus! You don't need to understand the details of the code

What can be **improved** from this approach?

Second Attempt

ASP.NET provides the **OOP way** to set the **HTTP Status**, **Content Type**, custom **Headers** and **Body** of the response

```
public class ManualController
{
   HttpContext ctx;
   public ManualController(
           IHttpContextAccessor _ctx) {
      ctx = _ctx.HttpContext;
   }

   public void Index() {
      ctx.Response.StatusCode = 200;
      ctx.Response.ContentType = "text/plain";

      byte[] content =
         Encoding.ASCII.GetBytes("Hello World");
      ctx.Response.Body.WriteAsync(
         content, 0, content.Length);
   }
}
```

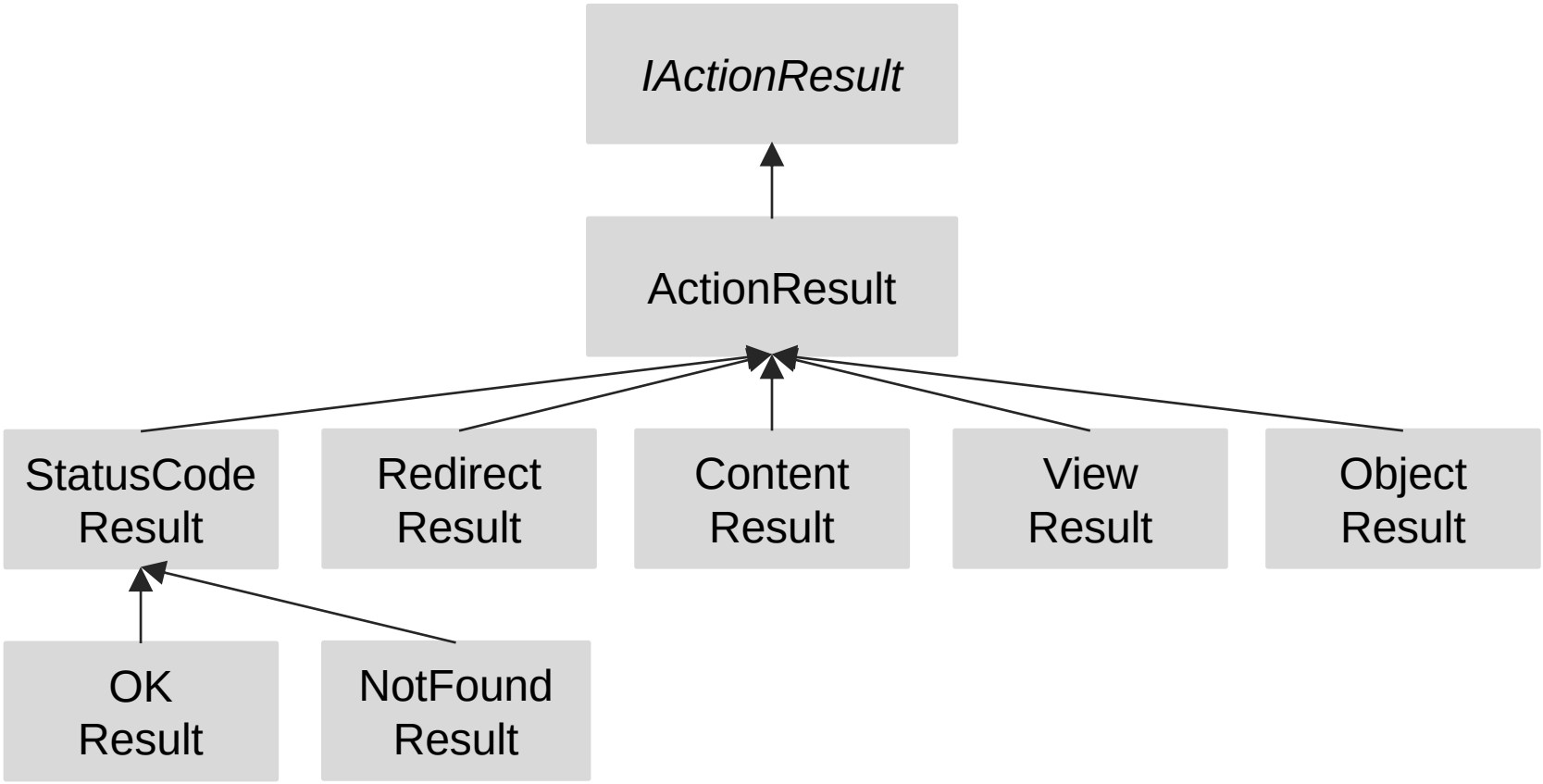
```
HTTP/1.1 200 OK
Content-Type: text/html
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2020
09:34:54 GMT

Hello World
```

What can be **improved** from this approach?

This code was used before .NET Core and therefore NOT our focus! You don't need to understand the details of the code

Better: Action Results

Action Results **encapsulate** all **low-level details**, making it **even easier** to build responses



Some Action Result classes

`ActionResult` classes end with suffix ***`Result`***

Action Results

When an **action method returns** an **Action Result**, ASP.NET Core **automatically generates** the respective HTTP Response

```
public ContentResult
            ContentResultExample()
{
 1 ContentResult contentRes =
          new ContentResult();
 2 contentRes.Content = "Hello World";

 3 return contentRes;
}
```

```
HTTP/1.1 200 OK
Content-Length: 11
Content-Type: text/plain;
charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2021 13:58:01
GMT

Hello World
```

ContentResult is one of the Action Results. Its **string content** will become the HTTP Response **body**. To create a response:

1. Instantiate a *ContentResult* object
2. Assign the necessary string to its *Content* property
3. Return the object

Action Results

When an action method returns an **Action Result**, ASP.NET Core **automatically generates** the respective HTTP Response

```
public NotFoundResult
            NotFoundResultExample()
{
 1 NotFound notFoundRes =
          new NotFound();

 2 return notFoundRes;
}
```

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 21 Sep 2021 01:44:24
GMT
```

NotFoundResult is one of the Action Results which just generates a 404 responses. To create a response:

1. Instantiate a *ContentResult* object
2. Return the object

Action Results

For some Action Results types, the Controller base class provides **respective helper methods**

```
public ContentResult ContentResultExample()
{
   ContentResult contentRes =
         new ContentResult();
   contentRes.Content = "Hello World";

   return contentRes;
}
```

```
HTTP/1.1 200 OK
Content-Length: 11
Content-Type: text/plain;
charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2021
13:58:01 GMT

Hello World
```

=

```
public ContentResult ContentResultExample()
{
   return Content("Hello World");
}
```

Similarly, *`OK()`* method returns *`OKResult`* objects;
`NotFound()` method returns *`NotFoundResult`* objects;
`Redirect()` and *`RedirectToAction()`* methods return *`RedirectResult`* objects; etc.

Action Results

Returning the *IActionResult* **instead** of the **actual type** helps us to return **any** of the Action Results

```
public IActionResult Index (int id)
{
    if (id == 0)
    {
        return NotFound();
    }
    else
    {
        return Content(
            "Hello World");
    }
}
```

```
HTTP/1.1 200 OK
Content-Length: 11
Content-Type: text/plain;
charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2021 13:58:01
GMT

Hello World
```

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 21 Sep 2021 01:44:24
GMT
```

What is the **language feature** that helps us achieve that?

Action Result type categories

.NET Core provides **many** action result types. Following are **categories** by the **responses' purposes**

Only status, no-body responses

Redirection responses

OK responses with string content

OK responses with HTML content using View

OK responses with object content

Topics

- HTTP Response
- Generating HTTP Response with ASP.NET Core
- **Action Results**
  - **Responses with only status code (Self-Study)**
  - **Redirect responses (Self-Study)**
  - **OK responses with string content (Self-Study)**
  - **OK responses with HTML**
  - **OK responses with object content (Self-Study)**

Responses with only status code

Self study

Merely return an HTTP status code, **no body**

```
public class DemoController :
                        Controller
{
  public IActionResult
              OnlyStatusCode()
   {
      // Other logic
      return NotFound();
   }
}
```

/Demo/OnlyStatusCode

```
HTTP/1.1 404 Not Found
Transfer-Encoding: chunked
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2020 03:32:59
GMT
```

Other Action Result classes that generate only status code are *OkResult*, *CreatedResult*, *BadRequestResult*… (following HTTP Status Code list)

Redirect Responses

Self study

Servers can use HTTP response to **direct** users to a **different web page**

```
public class DemoController
                          : Controller {
  public IActionResult ToWebpage() {
    return Redirect(
      "https://www.google.com");
  }
}
```

```
HTTP/1.1 302 Found
Transfer-Encoding: chunked
Location:
https://www.google.com
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2020 07:43:26
GMT
```

Redirect Responses

Self study

Server can redirect to **another action method** and ASP.NET Core helps generate the **target location**

```
public class DemoController
                            : Controller {
  public IActionResult ToAction()
  {
    return
      RedirectToAction("Privacy");
  }

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

```
HTTP/1.1 302 Found
Transfer-Encoding: chunked
Location: /Demo/Privacy
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2020 07:52:13
GMT
```

Redirect Responses

Self study

Server can also redirect to **another action method** in a **different controller**

```
public class DemoController
                          : Controller
{
  public IActionResult
                ToControllerAction()
   {
      return RedirectToAction(
                    "Index",
                    "Home");
   }
}
```

```
HTTP/1.1 302 Found
Transfer-Encoding: chunked
Location: /
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 19 Sep 2020 07:52:13
GMT
```

Can you guess why the generated **location** is only slash */*?

Redirect Responses

Self study

Data can be **passed** when returning a redirection, using **anonymous objects**

```
public class DemoController : Controller {
  public IActionResult
           RedirectToActionWithData() {
    var o = new {
       msg = "Hi DipSA",
       times = 4
    };
    return RedirectToAction("Say", o);
  }

  public IActionResult Say(
         string msg, int times) {
    string content = "";
    for (int i = 0; i < times; i++) {
      content += msg + "</p>";
    }
    return Content($"<html><body>{content}" +
           $"</body></html>", "text/html");
  }
}
```

```
HTTP/1.1 302 Found
Transfer-Encoding: chunked
Location:
/Demo/Say?msg=Hi%20DipSA
&times=4
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 7 Apr 2021 02:37:07
GMT
```

Redirect Responses

Self study

To redirect to an **attribute-based route**, specify the **Controller** and the **Method,** along with **data** (if any)

```
public IActionResult ToAttributeRoute() {
  return RedirectToRoute(new {
     controller = "AttributeBased",
     action = "Index",
     firstParam = 1,
     secondParam = 2
  });
}
```

```
HTTP/1.1 302 Found
Transfer-Encoding: chunked
Location:
/MyIndex?firstParam=1
&secondParam=2
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 21 Sep 2020
03:09:33 GMT
```

```
public class AttributeBasedController
                                  : Controller {
  [Route("MyIndex")]
  public IActionResult Index(
      int firstParam, int? secondParam) {
      // method implementation
   }
}
```

Questions

What are browsers usually do if it receives HTTP Responses in each of following cases:

1. A 404 response **with no body**
2. A 404 response **with body**
3. A 302 response with location: ***https://www.google.com***
4. A 302 response with location: **`/RedirectExample/Say?msg=Hi%20DipSA&times=3`**

Image by Gerd Altmann from Pixabay

OK Responses with String Content

Self study

String content can be returned with *Content()*, which can also specify the **returned MIME type**

```
public IActionResult Index() {
   return Content(
     "<html><body>Hello
World</body></html>", "text/html");
}
```

```
HTTP/1.1 200 OK
Content-Type: text/html
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 21 Sep 2020 06:00:11
GMT
Content-Length: 37

<html><body>Hello
World</body></html>
```

OK Responses with HTML Content

When **returning HTML** content, a better option is to **ask a View** to generate the HTML

```
public class HomeController :
                          Controller
{
    public IActionResult Index() {
        return View();
    }
}
```

```
@{
    Layout = null;
}

<p>Hi, I'm a dumb View</p>
```

View file

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/html;
charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 21 Sep 2020 06:10:10
GMT

<p>Hi, I'm a dumb
View</p>
```

What is the **name** of the **view file**? Where is it **located**?

OK Responses with HTML Content

We can ask a **non-default** View by providing the **target view name** in method *View()*

```
public class HomeController :
                          Controller
{
   public IActionResult Index() {
      return View("Smart");
   }
}
```

```
@{
   Layout = null;
}

<p>In fact, I'm even dumber</p>
```

Smart.cshtml

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/html;
charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 21 Sep 2020 06:10:10
GMT

<p>Hi, I'm even
dumber</p>
```

OK Responses with Object Content

Self study

Object content can be returned via an *ObjectResult* object.
Content-Type of the respective HTTP Response is **usually** JSON

```
public class ObjectController :
                          Controller
{
  public IActionResult
                ActionMethod3()
  {
    Person p = new Person()
    {
      Name = "John",
      Age = 20
    };

    return new ObjectResult(p);
  }
}
```

```
HTTP/1.1 200 OK
Content-Type:
application/json;
charset=utf-8
...

{"name":"John","age":20}
```

JSON https://www.youtube.com/watch?v=iiADhChRriM

OK Responses with Object Content

Self study

Content of an **anonymous object** or a **collection** of objects is also allowed

```
public IActionResult ActionMethod4() {
  var anonObj = new
  {
    Name = "Liz",
    School = "NUS"
  };

  return new ObjectResult(anonObj);
}

public IActionResult ActionMethod5() {
  List<Person> persons =
                        new List<Person>()
  {
    new Person { Name = "Dee", Age = 25 },
    new Person { Name = "Joy", Age = 28 }
  };

  return new ObjectResult(persons);
}
```

```
HTTP/1.1 200 OK
Content-Type:
application/json;
charset=utf-8
...

{"name":"Liz","school":"
NUS"}
```

```
HTTP/1.1 200 OK
Content-Type: application/json;
charset=utf-8
...

[{"name":"Alex","age":25
},{"name":"Mary","age":2
8}]
```

OK Responses with Object Content

Self study

Of course, content of a more **basic-type object** is also allowed

```
public IActionResult ActionMethod1()
{
   return new ObjectResult(3);
}

public IActionResult ActionMethod2()
{
   return new ObjectResult(
                     "Hello World");
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json;
charset=utf-8
...

3
```

```
HTTP/1.1 200 OK
Content-Type: text/plain;
charset=utf-8
...

Hello World
```

OK Responses with Object Content

Self study

Alternatively, object content can be returned **directly** by the **object type** itself

```
public class DatatypeController :
                                Controller
{
    public int ActionMethod1() {
        return 1 + 2;
    }

    public string ActionMethod2() {
        return "Hello World";
    }

    public Person ActionMethod3() {
        return new Person() {
            Name = "John",
            Age = 20
        };
    }
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json;
charset=utf-8
...

3
```

```
HTTP/1.1 200 OK
Content-Type: text/plain;
charset=utf-8
...

Hello World
```

```
HTTP/1.1 200 OK
Content-Type: application/json;
charset=utf-8
...

{"name":"John","age":20}
```

Here, no need to use Action Results, but we can't return different data types in the same method

OK Responses with Object Content

Self study

Alternatively, object content can be returned **directly** by the **object type** itself

```
public class DatatypeController :
                                Controller
{
  public object ActionMethod4() {
    return new
    {
      Name = "Liz",
      School = "NUS"
    };
  }

  public List<Person> ActionMethod5() {
    return new List<Person>() {
      new Person {Name = "Dee", Age = 25},
      new Person {Name = "Joy", Age = 28}
    };
  }
}
```

```
HTTP/1.1 200 OK
Content-Type:
application/json;
charset=utf-8
...

{"name":"Liz","school":"
NUS"}
```

```
HTTP/1.1 200 OK
Content-Type: application/json;
charset=utf-8
...

[{"name":"Alex","age":25
},{"name":"Mary","age":2
8}]
```

How popular is this option compared to Action Results?

Readings

- ActionResults in ASP.NET Core https://www.tektutorialshub.com/asp-net-core/asp-net-core-action-results/
- IActionResult and ActionResult demystified https://exceptionnotfound.net/asp-net-core-demystified-action-results/
- An overview of HTTP https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
- Action return types https://docs.microsoft.com/en-us/aspnet/core/web-api/action-return-types?view=aspnetcore-6.0
- JSON syntax https://www.w3schools.com/js/js_json_syntax.asp