Video 3.3

Implementing Post

|  |
| --- |
| **Metadata**: Spot the problem, highlight it, and design the solution in 3 core steps  (To be covered in the video) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem / Solution (Not more than 50 words)** | **Step 1 (Not more than 10 words)** | **Step 2 (Not more than 10 words)** | **Step 3 (Not more than 10 words)** |
| In this video, we will implement Post verb and see how the data is passed at server and response is sent back to client | First we will create a POST method | We will see how to pass data to Post method | Then see the response returned by the server |

|  |
| --- |
| **Script** the Video – Plan your narration (viewers will see and hear this) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Introduction** | | | |
| **No.** | **Action on Screen  (Code / Screenshots / One line about the action occurring on screen)** | **Narration**  **(The corresponding explanation to the Action on Screen)** | | |
| 1 | Replace the slide with a screenshot of the actual slide  C:\Data\Work\ForPackt\Deliverables\Draft-Scripts\Section 3\Video 3.3\images\331.png | | **Video Introduction**  In the previous video we looked at GET implementation and in this video we will implement Post. |
| 2 |  | |  |

|  |  |  |
| --- | --- | --- |
| **Steps or Tasks**  (Refer to the Writing Guidelines- Script Best Practices) | | |
| **No.** | **Action on Screen  (Code / Screenshots / One line about the action occurring on screen)** | **Narration**  **(The corresponding explanation to the Action on Screen)** |
| 3 | C:\Data\Work\ForPackt\Deliverables\Draft-Scripts\Section 3\Video 3.3\images\Demo.png | So let’s start with creating the Post method.  We will return here IActionResult which is base class. Now where we don’t have something like IHttpActionResult as now both MVC and Web API are same from framework point of view. Let’s decorate it with the HTTPPost attribute. I like to use because it is makes the methods self- descriptive. |
| 4 |  | We need a Book object that would be used for creation. Lets write the logic.  if(newBook == null)  {  return HttpBadRequest("Data is not received in proper format");  }  A bad request with error message will be returned to client.  So now lets call repository to addBook and it returns added book in response.  If addedBook is null  Then return BadRequest  Else  Return we need to return the 201 response code along with the location and/or the added object.  ASP.NET 5 provides a method *CreatedAtRoute* which does all it. It takes route name, controller name, Id and the added object in parameter as  Now we have done with the controller code. Now it’s time to run it and add a book |
| 5 | http://localhost:3795/api/books/4/true | As we discussed earlier that in Post we pass the data via Body. Let’s first get a book via get request say with id 4  So we have got the book with Author. Now let’s modify a bit and create a new book..  Before executing it, we need to add one header content-type to tell that in what format we are sending the data. So lets add it as content-type as application/json  Now lets send the request  Let’s see the book object.. Oh.. it is not populated as per the request. So something is missing |
| 6 |  | As we sent the content in body, how does the web api method know that from where book is to be populated. We need to add one attribute FromBody as  Lets try once more  [Debug] Awesome!! Now we got the initialized object as we passed in the request. Let it get continued |
| 7 |  | Lets check the response. Here the response code we got 201 (created) and in the body we got the created object with the Id. |
| 8 |  | Let’s see the header. Here we got additional header with location of the resource uri. Now we just copy the location and make a get request we will get the same instance. |
| 9 |  | Similarly we can send data in XML format and accordingly we need to add the content type header. |
| 10 |  | Let’s send one more request with wrong data to see the behaviour.  I am just adding some junk characters here  So here we can see bad request and with the message in the body that we sent. |

|  |  |
| --- | --- |
| **Summary** | |
|  | In this video, we have seen how to implement Post and way to pass the data to server. |
| [Mandatory slide] – Next Video  Slide8.PNG | In the next video, we will implement another Verb Put |