Video 3.5

Implementing DELETE

|  |
| --- |
| **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)** |
| This video covers the implementation of Delete verb | Implement Delete verb | Testing the delete verb |  |

|  |
| --- |
| **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 3.5\images\351.png | | **Video Introduction**  In the previous video we learned Put verb and its implementation. Till now we have completed create, read and update a resource. In this video, we will start with Implementing Delete verb. |
| 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 3.5\images\Demo.png | What do we require to delete a resource or here a book?  An Identifier of the resource!! Thats it.  So in this scenario, we just need the id of the book and that should be passed via URL. |
| 4 | [HttpDelete("{id:int}")]  public IActionResult Delete(int id)  {  if (id == 0)  {  return HttpBadRequest("Data is not received in proper format");  }  var book = this.repository.FindBook(id, false);  if(book == null)  {  return new HttpNotFoundResult();  }  bool isDeleted = this.repository.DeleteBook(id);  if (!isDeleted)  {  return HttpBadRequest("Error while deleting the book");  }  return new HttpOkResult();  } | I have already added Delete method and lets have a look on that  So in delete method we decorated it with HttpDelete attribute and provided the Id here as earlier and it will be passed via URL.  First we are checking whether is Id is good then we are checking the book whether this book with this exist in the system or not. In case of no, we are returning NotFound status.  After that, we are deleting the book and if it returns true which means that book is deleted successfully then we are sending Ok response else we are sending 400 |
| 5 |  | Now time to test it.  Here we are going to delete a book with Id .. . Before that let’s first see it existence via get request.  We got the book in response, it means it exists. |
| 6 |  | Let’s send the delete request now.  Here we can see that id is initialized with 10. That’s good. Let’s get it executed  We got the 200 response it means it executed successfully. |
| 7 |  | Now lets run the same request again and if the item would not be found then it should return not found status.  Yeah we are getting 404 not found status. It means that item was deleted in previous request. |
| 8 |  | It means our delete is working as expected. |

|  |  |
| --- | --- |
| **Summary** | |
|  | In this video, we have discussed delete verb |
| [Mandatory slide] – Next Video  C:\Data\Work\ForPackt\Deliverables\Draft-Scripts\Section 3\Video 3.5\images\353.png | In the next video, we will discuss about Patch keyword |