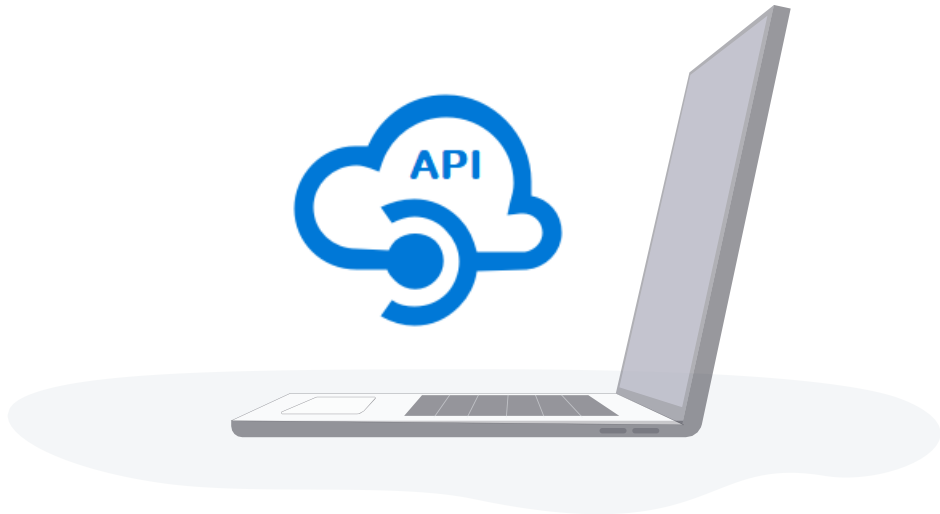
Harmony IT Solution

Web Application Programming Interface (API)

Tahaluf Training Center 2022

Harmony IT Solution

1. Overview Of HTTP Verbs
2. Overview Of HTTP Status Code
3. Controller
4. Upload Image



Overview Of HTTP Verbs

REST APIs enable to development of any kind of web application having all CRUD operations (Create, Retrieve, Update, Delete).

REST guidelines suggest using specific HTTP verbs on a particular type of call made to the server.

HTTP GET

Use GET requests **to retrieve resource represent information only** and not to update it in any way.
As GET requests do not update the state of the resource, these are said to be **safe methods**.

HTTP POST

Use POST APIs **to create new resources**, When talking strictly in terms of REST, POST verbs are used to create a new resource into the collection of resources.

HTTP PUT

Use PUT APIs primarily **to modify existing resource** (if the resource does not exist, then API may decide to create a new resource or not).

HTTP DELETE

DELETE APIs are used **to delete resources** (identified by the Request URI).

Harmony IT Solution

Overview Of HTTP Status Code

Harmony IT Solution

HTTP STATUS CODES

5xx Server Error

- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout

4xx Client Error

- 401 Unauthorized Error
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed

3xx Redirection

- 301 Permanent Redirect
- 302 Temporary Redirect
- 304 Not Modified

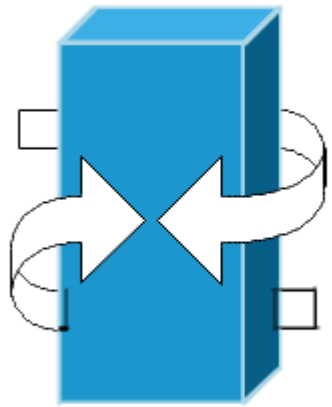
2xx Success

- 200 Success / OK

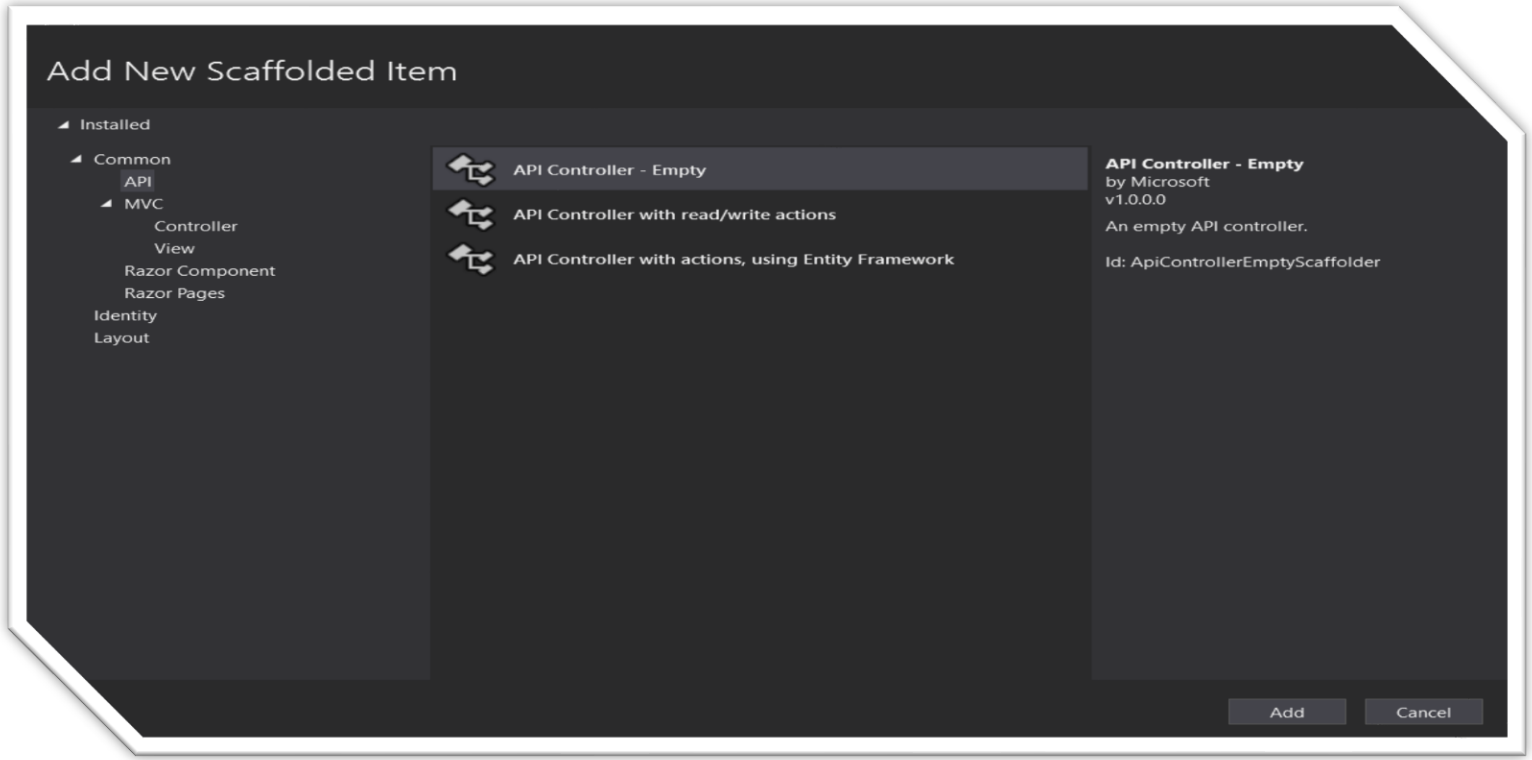
Controller

Web API Controller handles incoming HTTP methods requests and sends a response to the caller.

Web API controller class can be created in the Controllers folder or any other folder in the project's root folder.



Right Click on Controllers => Add => Controller => Choose API Controller – Empty => CourseController.



In TahalufLearn.API => Controller => CourseController add the following :

```
private readonly ICourseService courseService;

        public CourseController(ICourseService
courseService)
        {
            this.courseService = courseService;
        }
```

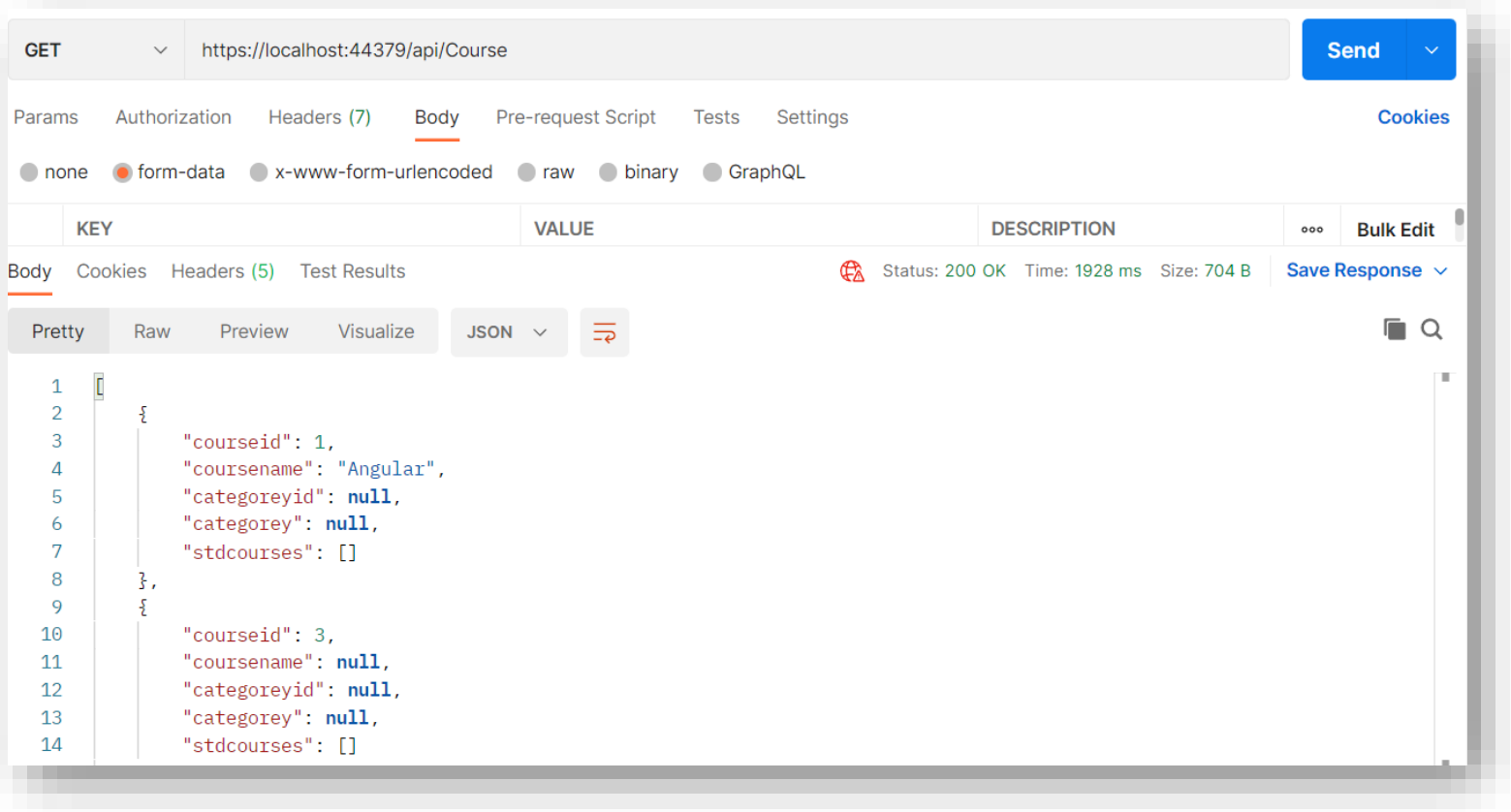
In TahalufLearn.API => Controller => CourseController add the following :

```
[HttpGet]
        public List<Course> GetAllCourse()
        {
            return courseService.GetAllCourse();
        }
```

In Postman:

1. In URL add => Https://LocalHost:PortNumber/Api/ControllerName/[RouteName]/[Parameters]
2. Select The method => (Get)
3. Send the request

Harmony IT Solution



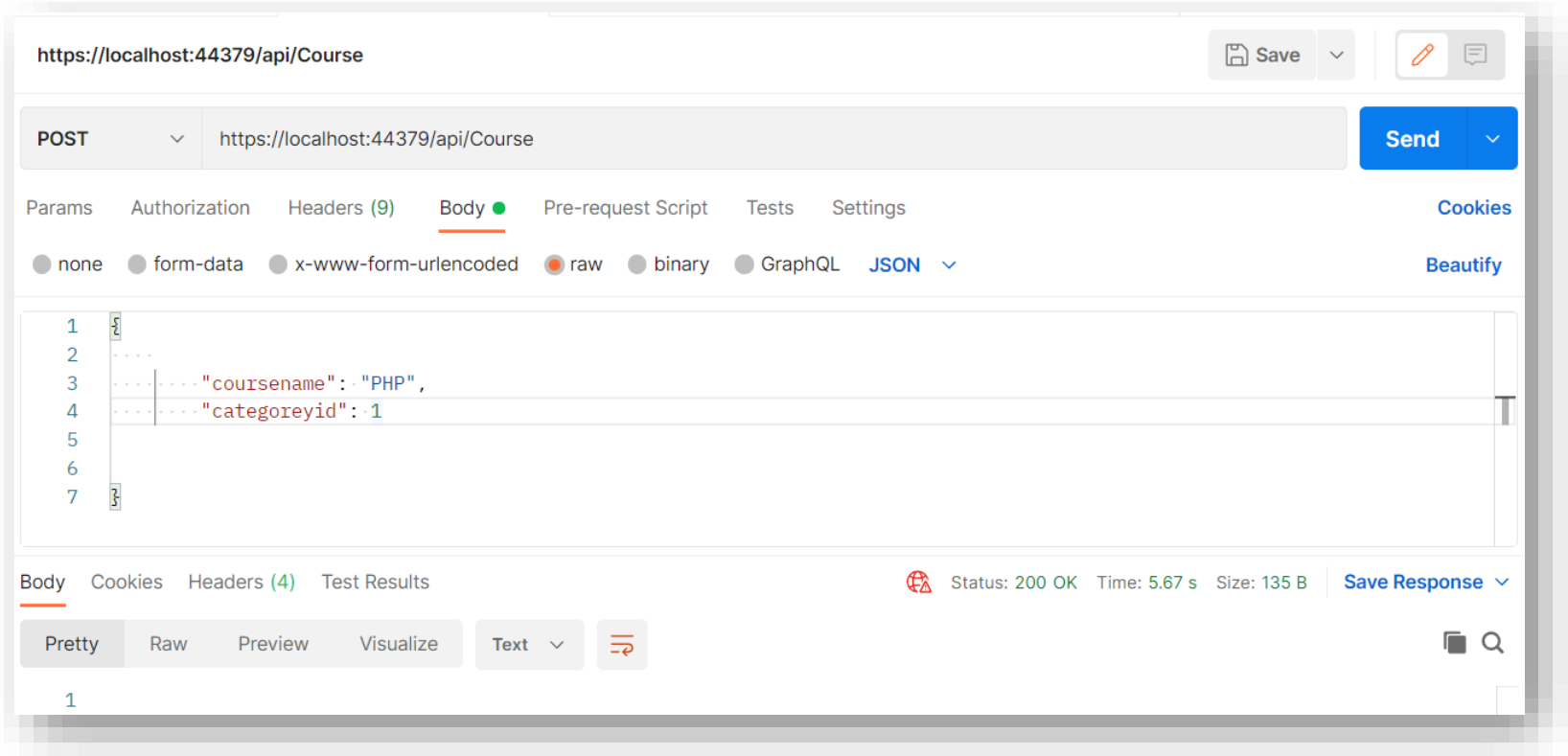
In TahalufLearn.API => Controller => CourseController add the following :

```
[HttpPost]
        public void CreateCourse(Course course)
            {
             courseService.CreateCourse(course);
            }
```

In Postman:

1. In URL add => Https://LocalHost:PortNumber/Api/ControllerName/[RouteName]/[Parameters]
2. Select The method => (Post)
3. Choose Body => raw => JSON => then add the course data as JSON object.
4. Send the request

Harmony IT Solution



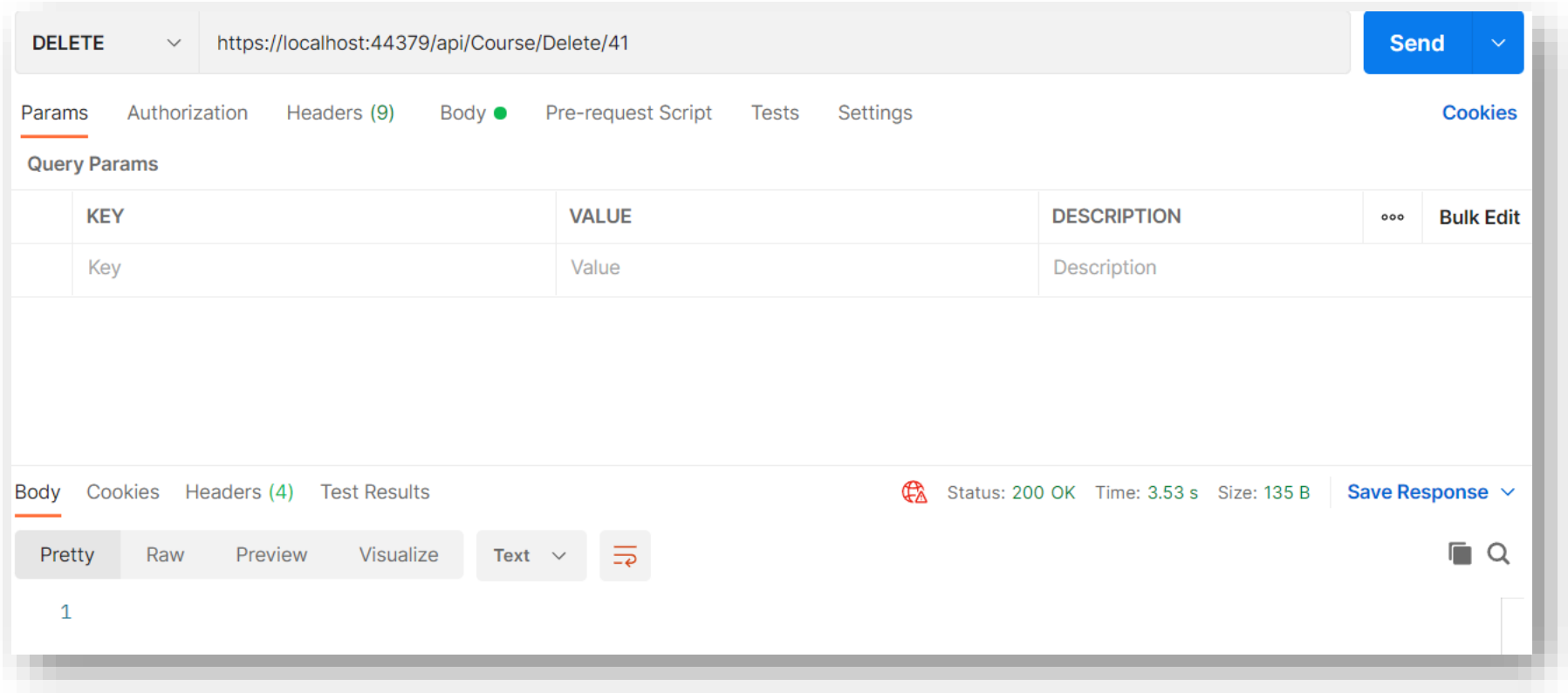
In TahalufLearn.API => Controller => CourseController add the following :

```
[HttpDelete]
        [Route("delete/{id}")]
        public void DeleteCourse(int id)
            {
            courseService.DeleteCourse(id);
            }
```

In Postman:

1. In URL add => Https://LocalHost:PortNumber/Api/ControllerName/[RouteName]/[Parameters]
2. Select The method => (Delete)
1. Send the request

Harmony IT Solution



In TahalufLearn.API => Controller => CourseController add the following :

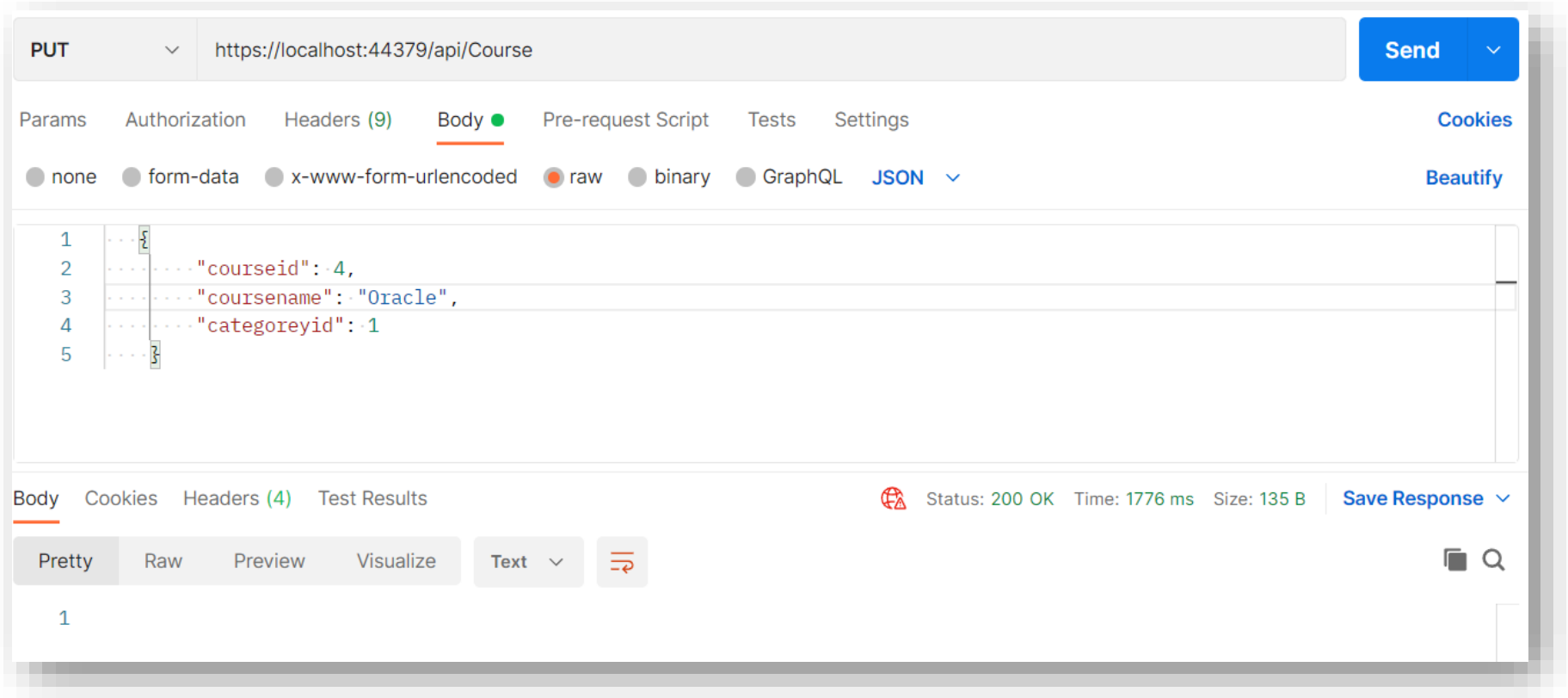
```
[HttpPut]
        public void UpdateCourse(Course course)

            {
             courseService.UpdateCourse(course);
            }
```

In Postman:

1. In URL add => Https://LocalHost:PortNumber/Api/ControllerName/[RouteName]/[Parameters]
2. Select The method => (Put)
3. Choose Body => raw => JSON => then add the course data as JSON object.
4. Send the request

Harmony IT Solution



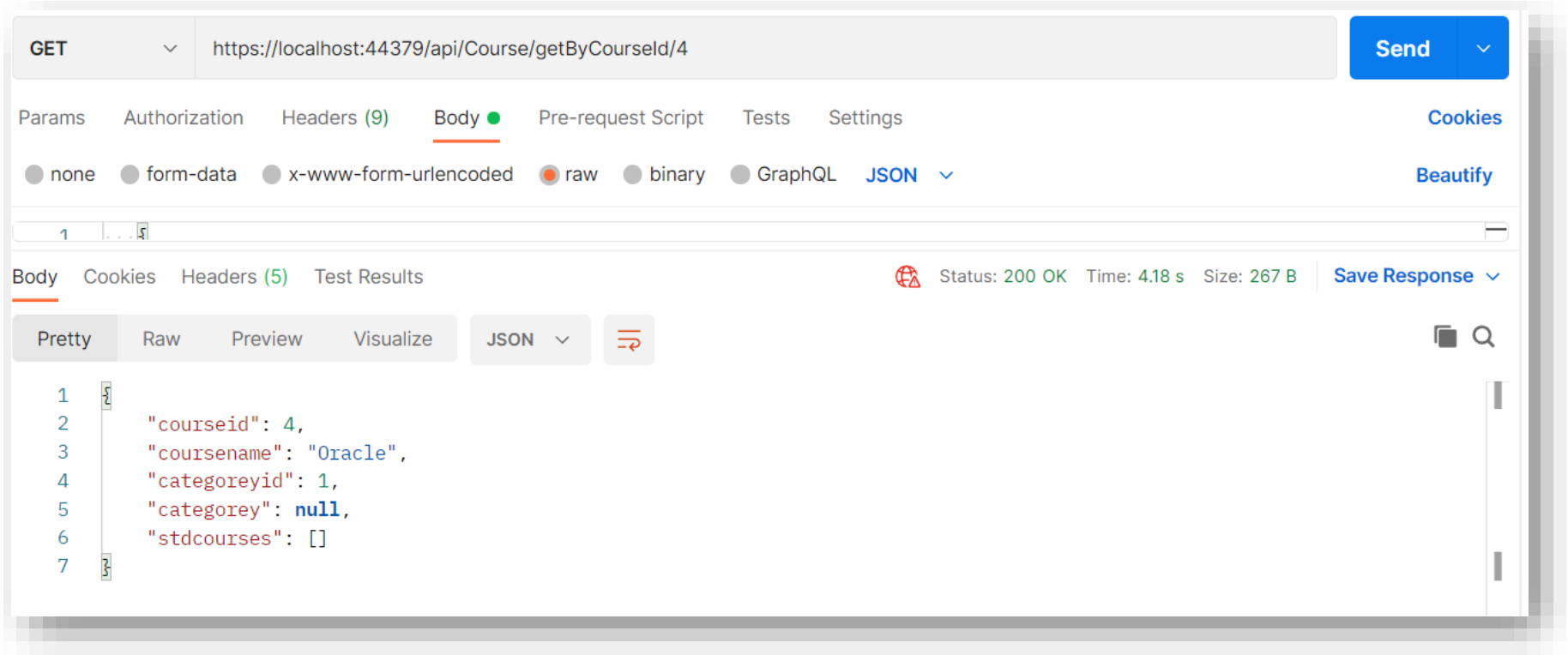
Harmony IT Solution

In TahalufLearn.API => Controller => CourseController add the following :

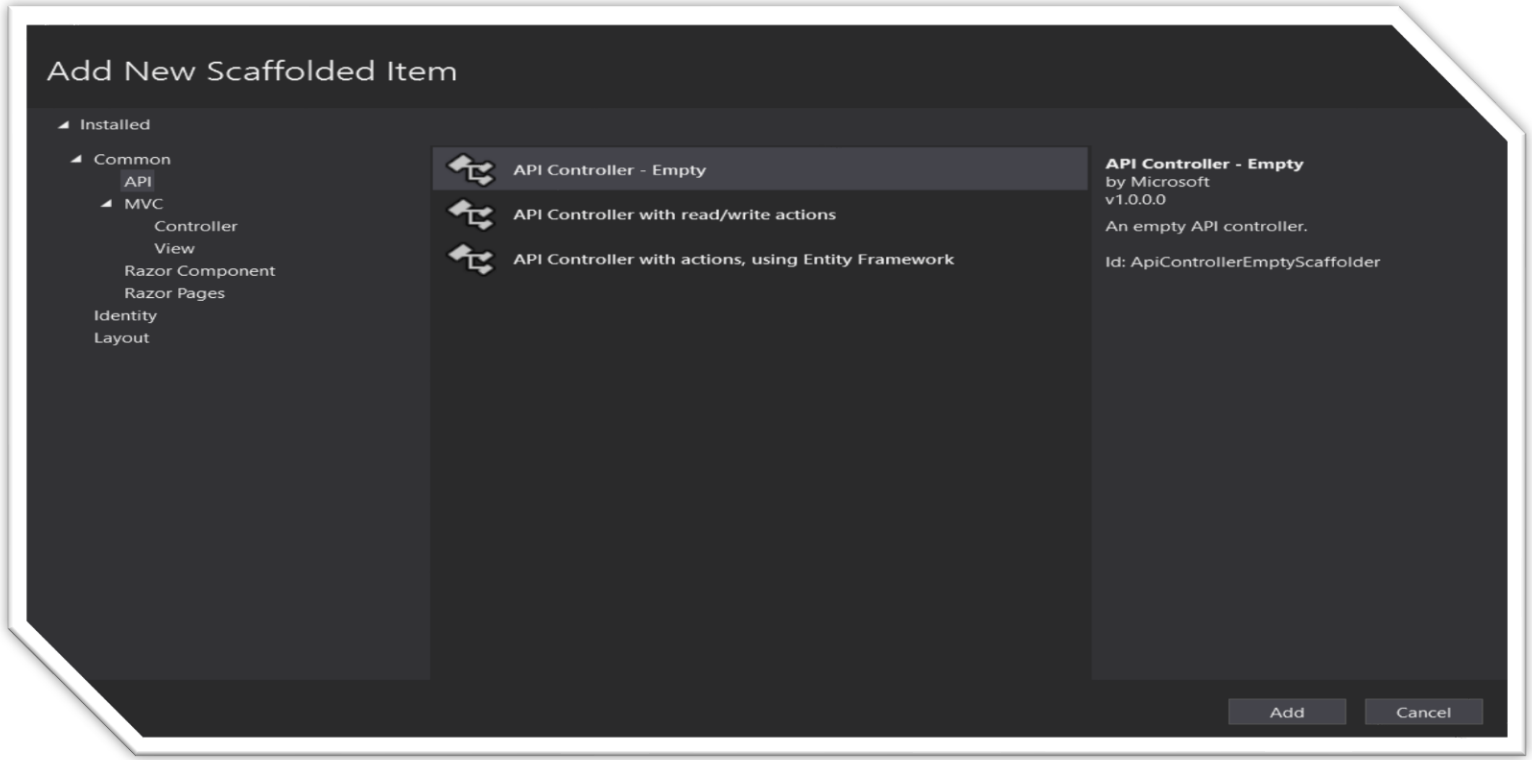
```
[HttpGet]
        [Route("getByCourseId/{id}")]
        public Course GetByCourseId(int id)
            {
            return courseService.GetByCourseId(id);
            }
```

In Postman:

1. In URL add => Https://LocalHost:PortNumber/Api/ControllerName/[RouteName]/[Parameters]
2. Select The method => (Get)
1. Send the request



Right Click on Controllers => Add => Controller => Choose API Controller – Empty => StudentController.



In TahalufLearn.API => Controller => StudentController add the following :

```
private readonly IStudentService _studentService;

        public StudentController(IStudentService Istdservice)
        {
            this._studentService = Istdservice;
        }
```

In TahalufLearn.API => Controller => StudentController add the following :

```
[HttpGet]
        public List<Student> GetAllStudent()
            {
            return _studentService.GetAllStudent();
            }
```

Harmony IT Solution

In TahalufLearn.API => Controller => StudentController add the following :

```
[HttpPost]
        public void CreateStudent(Student Student)
            {
             _studentService.CreateStudent(Student);
            }
```

In TahalufLearn.API => Controller => StudentController add the following :

```
[HttpPut]
        public void UpdateStudent(Student Student)
            {
            _studentService.UpdateStudent(Student);
            }
```

Harmony IT Solution

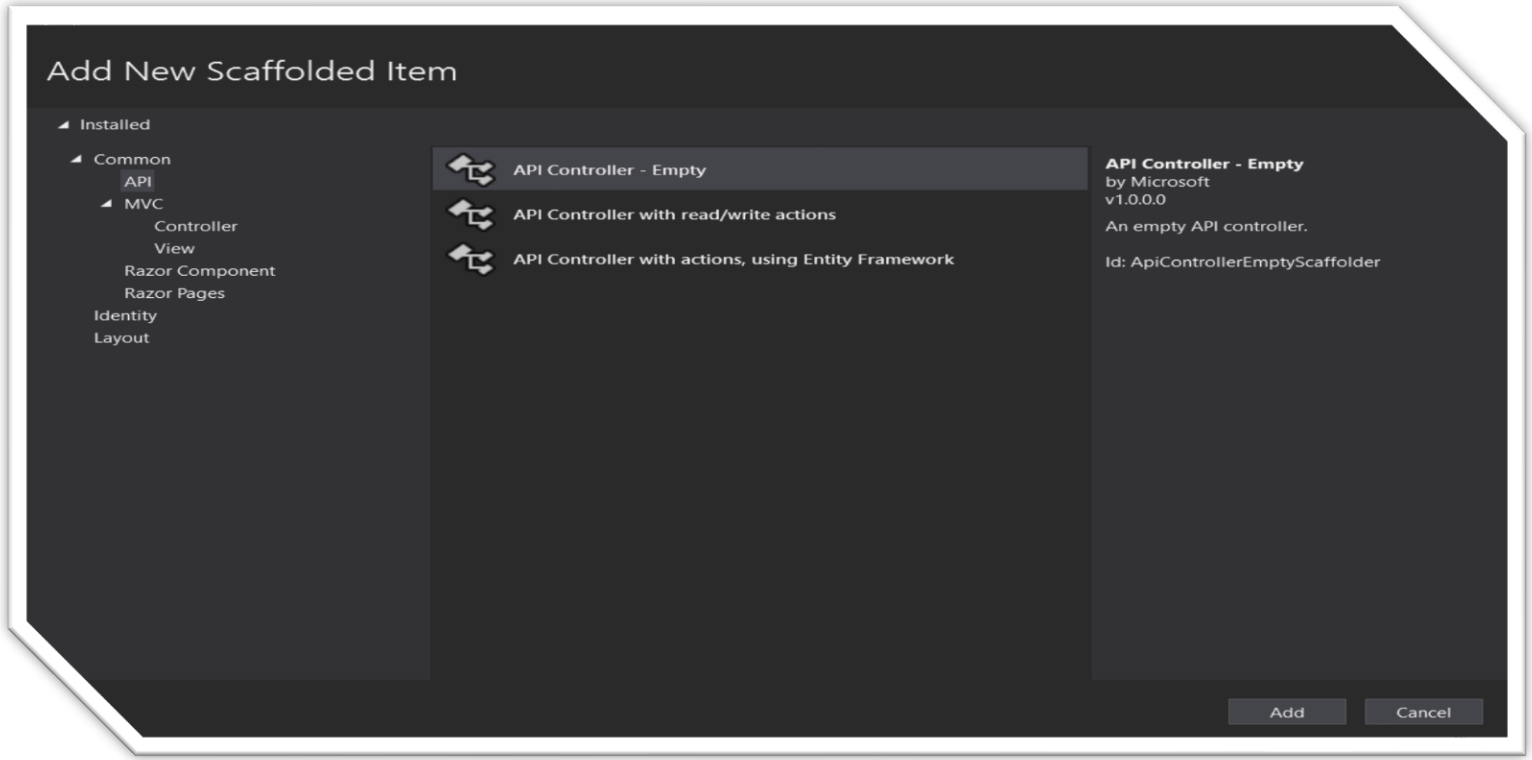
In TahalufLearn.API => Controller => StudentController add the following :

```
[HttpDelete]
        [Route("delete/{id}")]
        public void DeleteStudent(int id)
            {
            _studentService.DeleteStudent(id);
            }
```

In TahalufLearn.API => Controller => StudentController add the following :

```
[HttpGet]
        [Route("getByStudentId/{id}")]
        public Student GetStudentById(int id)
            {
            return _studentService.GetStudentById(id);
            }
```

Right Click on Controllers => Add => Controller => Choose API Controller – Empty => StudentCourseController.



In TahalufLearn.API => Controller => StudentCourseController add the following :

```
private readonly IStudentCourseService _studentCourseService;

        public StudentCourseController(IStudentCourseService
studentCourseService)
            {
            _studentCourseService = studentCourseService;
            }
```

In TahalufLearn.API => Controller => StudentCourseController add the following :

```
[HttpPost]
        public void CreateStudentCourse(StdCourse studentCourse)
            {
            _studentCourseService.CreateStudentCourse(studentCourse);
            }
```

In TahalufLearn.API => Controller => StudentCourseController add the following :

```
[HttpDelete]
        [Route("delete/{id}")]
        public void DeleteStudentCourse(int id)
            {
            _studentCourseService.DeleteStudentCourse(id);
            }
```

In TahalufLearn.API => Controller => StudentCourseController add the following :

```
[HttpGet]
        public List<StdCourse> GetAllStudentCourse()
            {
            return _studentCourseService.GetAllStudentCourse();
            }
```

In TahalufLearn.API => Controller => StudentCourseController add the following :

```
[HttpGet]
        [Route("getStudentCourseById/{id}")]
        public StdCourse GetStudentCourseById(int id)
            {
            return _studentCourseService.GetStudentCourseById(id);
            }
```

In TahalufLearn.API => Controller => StudentCourseController add the following :

```
[HttpPut]
public void UpdateStudentCourse(StdCourse studentCourse)
    {
    _studentCourseService.UpdateStudentCourse(studentCourse);
    }
```

Exercise

- ✓ Create a function to display FirstName and LastName from table student.
- ✓ Create a function to display students by firstName.
- ✓ Create a function to display students by BirthOfDate.
- ✓ Create a function to display a student by BirthOfDate interval.
- ✓ Create a function to display the student name with the highest n(2,3,…) marks

Harmony IT Solution

In TahalufLearn.API => Right Click => Add New Folder (Images):

Create upload image function in Course Controller:

```
[Route("uploadImage")]
        [HttpPost]
        public Course UploadIMage()
            {
            var file = Request.Form.Files[0];
            var fileName = Guid.NewGuid().ToString() + "_" + file.FileName;
            var fullPath = Path.Combine("Images", fileName);
```

Create upload image function in Course Controller:

```
using (var stream = new FileStream(fullPath, FileMode.Create))
                {
                file.CopyTo(stream);
                }
            Course item = new Course();
            item.ImageName = fileName;
            return item;
            }
```

```
[Route("uploadImage")]
[HttpPost]
0 references
public Course UploadIMage()
    {
    var file = Request.Form.Files[0];
    var fileName = Guid.NewGuid().ToString() + "_" + file.FileName;
    var fullPath = Path.Combine("Images", fileName);
    using (var stream = new FileStream(fullPath, FileMode.Create))
        {
        file.CopyTo(stream);
        }
    Course item = new Course();
    item.ImageName = fileName;
    return item;
    }
```