In Web API 1, we have HttpResponseMessage type that a controller action method returns. A new type called "IHttpActionResult" is introduced in Web API 2 that can be returned from a controller action method. Instead of returning HttpResponseMessage from a controller action, we can now return IHttpActionResult. There are 2 main advantages of using the IHttpActionResult interface.

1. The code is cleaner and easier to read
2. Unit testing controller action methods is much simpler. We will discuss, how easy it is to unit test a method that returns IHttpActionResult instead of HttpResponseMessage in a later video.

Consider the following StudentsController. Notice both the Get() methods return HttpResponseMessage. To create the HttpResponseMessage, we either use CreateResponse() or CreateErrorResponse() methods of the Request object. 

public class StudentsController : ApiController

{

    static List<Student> students = new List<Student>()

    {

        new Student() { Id = 1, Name = "Tom" },

        new Student() { Id = 2, Name = "Sam" },

        new Student() { Id = 3, Name = "John" }

    };

    public HttpResponseMessage Get()

    {

        return Request.CreateResponse(students);

    }

    public HttpResponseMessage Get(int id)

    {

        var student = students.FirstOrDefault(s => s.Id == id);

        if (student == null)

        {

            return Request.CreateErrorResponse(HttpStatusCode.NotFound,

                "Student not found");

        }

        return Request.CreateResponse(student);

    }

}

In the following example, we have replaced both instances of HttpResponseMessage with IHttpActionResult. To return status code 200, we used Ok() helper method and to return status code 404, we used NotFound() method. To the Ok() method we have passed the type we want to return from the action method. Also notice, the code is now much cleaner and simpler to read. 

public class StudentsController : ApiController

{

    static List<Student> students = new List<Student>()

    {

        new Student() { Id = 1, Name = "Tom" },

        new Student() { Id = 2, Name = "Sam" },

        new Student() { Id = 3, Name = "John" }

    };

    public IHttpActionResult Get()

    {

        return Ok(students);

    }

    public IHttpActionResult Get(int id)

    {

        var student = students.FirstOrDefault(s => s.Id == id);

        if (student == null)

        {

            //return NotFound();

            return Content(HttpStatusCode.NotFound, "Student not found");

        }

        return Ok(student);

    }

}

**In addition to Ok() and NotFound() helper methods**, we have the following methods that we can use depending on what we want to return from our controller action method. All these methods return a type, that implements IHttpActionResult interface.

* BadRequest()
* Conflict()
* Created()
* InternalServerError()
* Redirect()
* Unauthorized()