We want to add a new Employee to the Employees table. Include the following Post() method in EmployeesController. Notice the Employee object is being passed as parameter to the Post method. The Employee parameter is decorated with [FromBody] attribute. This tells Web API to get employee data from the request body.

public void Post([FromBody] Employee employee)

{

    using (EmployeeDBEntities entities = new EmployeeDBEntities())

    {

        entities.Employees.Add(employee);

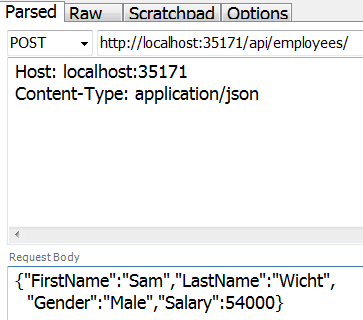
        entities.SaveChanges();

    }

}

**At this point build the solution. Fireup Fiddler and issue a Post request**

1. Set the HTTP verb to POST
2. Content-Type: application/json. This tells that we are sending JSON formatted data to the server
3. In the Request Body, include the employee object that you want to add to the database in JSON format
4. Finally click execute

   
  
This works fine and adds the employee to the database as expected. The problem here is that since the return type of the Post() method is void, we get status code of 204 No Content. When a new item is created, we should actually be returning status code 201 Item Created. With 201 status code we should also include the location i.e URI of the newly created item. To achieve this, change the implementation of the Post() method as shown below. 

public HttpResponseMessage Post([FromBody] Employee employee)

{

    try

    {

        using (EmployeeDBEntities entities = new EmployeeDBEntities())

        {

            entities.Employees.Add(employee);

            entities.SaveChanges();

            var message = Request.CreateResponse(HttpStatusCode.Created, employee);

            message.Headers.Location = new Uri(Request.RequestUri +

                employee.ID.ToString());

            return message;

        }

    }

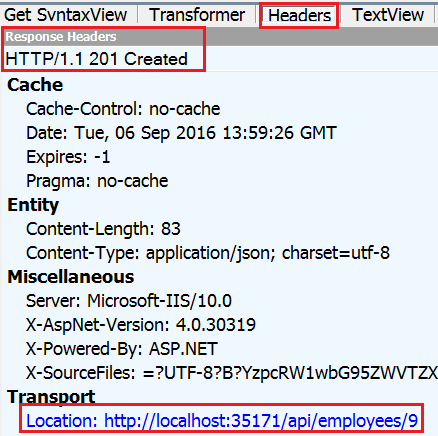
    catch (Exception ex)

    {

        return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ex);

    }

}

At this point, issue another Post request from fiddler. Notice in the response header we have status code 201 Created and also the location i.e the URI of the newly created item.   
   
  
Let's also modify Get(int id) method. At the moment when we issue a request for an employee id that does not exist, we get null back and the status code is 200 OK. According to the standards of REST when an item is not found we should be returning 204 Not Found. To achieve this modify the **Get(int id)** method as shown below. 

public HttpResponseMessage Get(int id)

{

    using (EmployeeDBEntities entities = new EmployeeDBEntities())

    {

        var entity = entities.Employees.FirstOrDefault(e => e.ID == id);

        if (entity != null)

        {

            return Request.CreateResponse(HttpStatusCode.OK, entity);

        }

        else

        {

            return Request.CreateErrorResponse(HttpStatusCode.NotFound,

                "Employee with Id " + id.ToString() + " not found");

        }

    }

}

At this point, when we issue a request for an employee with ID = 101 which does not exist we get a 404 along with the message "Employee with Id 101 not found"  
  
**Here are the important points to remember**

* If a method return type is void, by default status code 204 No Content is returned.
* When a new item is created, we should be returning status code 201 Item Created.
* With 201 status code we should also include the location i.e URI of the newly created item.
* When an item is not found, instead of returning NULL and status code 200 OK, return 404 Not Found status code along with a meaningful message such as "Employee with Id = 101 not found"