Let us understand query string parameters in ASP.NET Web API with an example.  
  
We want to modify the following Get() method in **EmployeesController**so that we can retrieve employees by gender

public class EmployeesController : ApiController

{

    public IEnumerable<Employee> Get()

    {

        using (EmployeeDBEntities entities = new EmployeeDBEntities())

        {

            return entities.Employees.ToList();

        }

    }

}

Depending on the value we specify for query string parameter gender, the Get() method should return the data.

| **Query String** | **Data** |
| --- | --- |
| http://localhost/api/employees?gender=All | All Employees |
| http://localhost/api/employees?gender=Male | Only Male Employees |
| http://localhost/api/employees?gender=Female | Only Female Employees |

If the value for gender is not Male, Female or All, then the service should return status code 400 Bad Request. For example, if we specify **ABC**as the value for gender, then the service should return status code 400 Bad Request with the following message.  
Value for gender must be Male, Female or All. ABC is invalid.  
  
**Below is the modified Get() method**

1. Gender is being passed as a parameter to the Get() method
2. Default value is "All". The default value makes the parameter optional
3. The gender parameter of the Get() method is mapped to the gender parameter sent in the query string

public HttpResponseMessage Get(string gender = "All")

{

    using (EmployeeDBEntities entities = new EmployeeDBEntities())

    {

        switch (gender.ToLower())

        {

            case "all":

                return Request.CreateResponse(HttpStatusCode.OK,

                    entities.Employees.ToList());

            case "male":

                return Request.CreateResponse(HttpStatusCode.OK,

                    entities.Employees.Where(e => e.Gender.ToLower() == "male").ToList());

            case "female":

                return Request.CreateResponse(HttpStatusCode.OK,

                    entities.Employees.Where(e => e.Gender.ToLower() == "female").ToList());

            default:

                return Request.CreateErrorResponse(HttpStatusCode.BadRequest,

                    "Value for gender must be Male, Female or All. " + gender + " is invalid.");

        }

    }

}