**FOLLOW THIS LINK (IMP)**

[**https://www.dotnettricks.com/learn/webapi/model-binding-model-binder-example**](https://www.dotnettricks.com/learn/webapi/model-binding-model-binder-example)

**Model Binder**

<https://localhost:44383/api/Student?id=2>

<https://localhost:44383/api/Student/2>

These two calls will invoke same method

**FromBody and FromUri attributes**

Web Api Default convention for binding parameters

1. If the parameter is a simple type like int, bool, double, etc. Web Api tries to get the value from the URI (either from Route Data or QueryString)
2. If the parameter is a complex type like Customer, Employee, etc. Web Api tries to get value from the request body;

To change the default parameter binding process, we can use [FromBody] and [FromUri] attributes

public IHttpActionResult PutStudent([FromUri] int id, [FromBody] Student student)

{

}

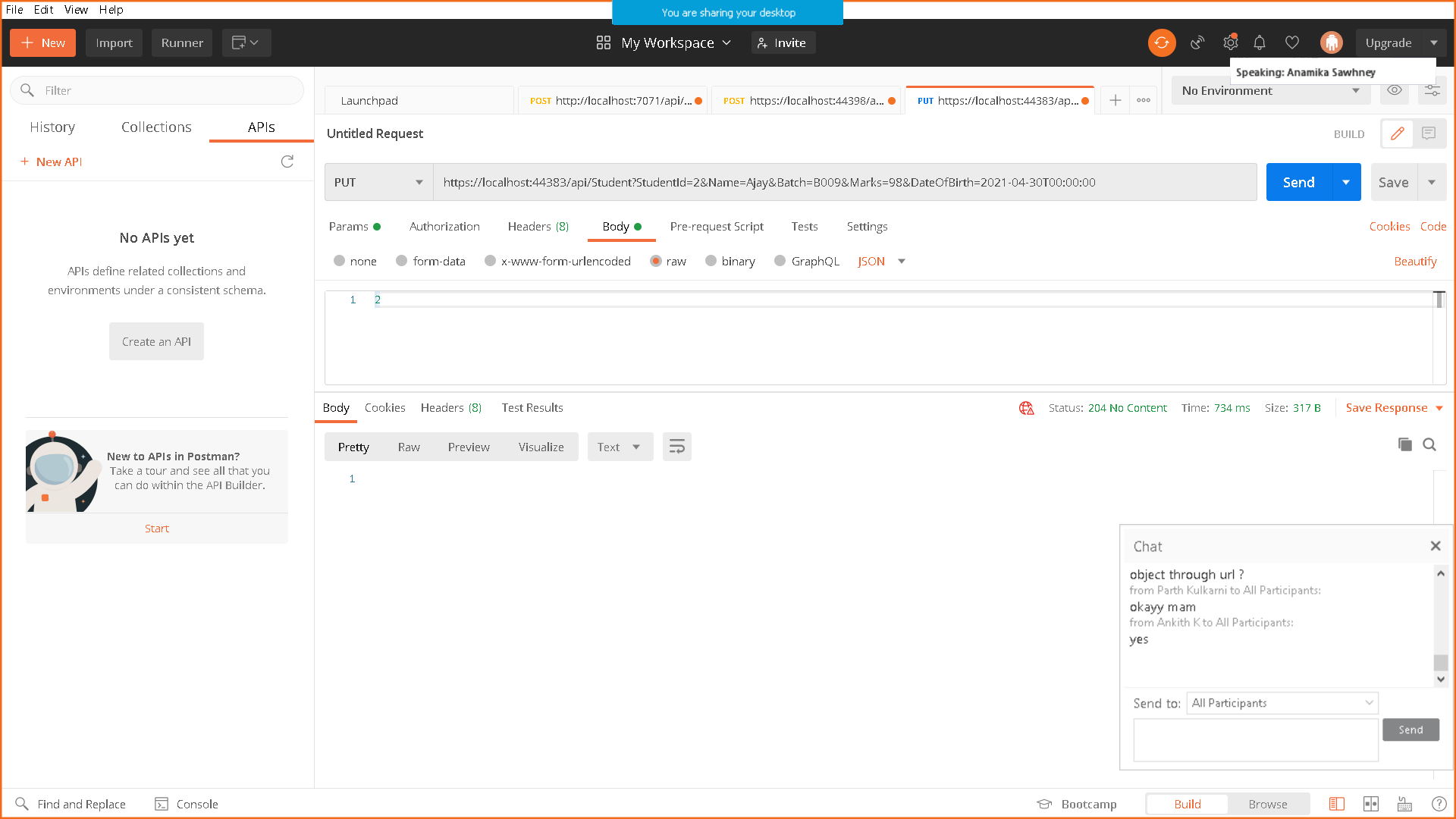
public IHttpActionResult PutStudent([FromBody] int id, [FromUri] Student student)

{

}

public IHttpActionResult PutStudent([FromBody] int id, [FromUri] Student student)

<https://localhost:44383/api/Student?StudentId=2&Name=Ajay&Batch=B009&Marks=98&DateOfBirth=2021-04-30T00:00:00>



public IHttpActionResult PutStudent(int id, [FromUri] Student student)

{

<https://localhost:44383/api/Student/2?StudentId=2&Name=Ajay&Batch=B009&Marks=98&DateOfBirth=2021-04-30T00:00:00>

<https://localhost:44383/api/Student?id=2&StudentId=2&Name=Ajay&Batch=B009&Marks=98&DateOfBirth=2021-04-30T00:00:00>

Use [FromBody] to force Web Api to get simple types from the request body;

Use [FromUri] to force Web Api to get complex types data from the URI (i.e. RouteData or queryString)

[ResponseType(typeof(void))]

public IHttpActionResult PutStudent(int id, [FromUri] Student student)

{

Test in Postman

https://localhost:44383/api/Student?id=3&StudentId=3&Name=New Name&Batch=New Batch&Marks=100&DateOfBirth=2020-12-12T00:00:00

[ResponseType(typeof(void))]

public IHttpActionResult PutStudent([FromBody] int id, [FromUri] Student student)

{

Test in Postman

https://localhost:44383/api/Student?StudentId=3&Name=New Name&Batch=New111 Batch&Marks=100&DateOfBirth=2020-12-12T00:00:00

in Request, raw, json

Type

1

What is MediaTypeFormatter

MediaTypeFormatter is an abstract class from which JsonMediaTypeFormatter and XmlMediaTypeFormatter classed inherits from

How to return only Json from Asp.Net Web Api irrespective of Accept-Header value

public static void Register(HttpConfiguration config)

{

EnableCorsAttribute cors = new EnableCorsAttribute("\*", "\*", "GET,POST,PUT,DELETE");

config.EnableCors(cors);

**config.Formatters.Remove(config.Formatters.XmlFormatter);**

// Web API configuration and services

What is **ASP**.**NET WEB API TypeConverter**? **TypeConverter** is the **converter** class which would be used to **convert** a single query string parameter to the class object that we want to map to. **TypeConverter** can be created by inheriting from the class **TypeConverter** and override its two methods CanConvertFrom, and ConvertFrom

<https://nikhildoomra.wordpress.com/2018/06/08/asp-net-web-api-typeconverter-and-modelbinders/#:~:text=What%20is%20ASP.NET%20WEB,two%20methods%20CanConvertFrom%2C%20and%20ConvertFrom>.