MISA®
TIN CẬY - TIỆN ÍCH - TẬN TÌNH

ASP.NET WEB API

Nguyễn Văn Mạnh

Nội dung chính

- **Tìm hiểu về Web API**
- **Xây dựng ứng dụng Web API**
- **Cấu hình Web API**
- **Web API Routing**
- **Web API Controller**
- **Parameter Binding.**

.NET WEB API là gì?

- API là viết tắt của Application Programming Interface, phương thức **kết nối với các thư viện và ứng dụng khác**
- **Web API là** công nghệ khá mới của hãng Microsoft để xây dựng dịch vụ thành phần phân tán.**Web API là** mô hình hỗ trợ MVC: routing, controller, action result, filter, filte.
- ASP.NET Web API là một framework để xây dựng các dịch vụ HTTP có thể được truy cập từ bất kỳ khách hàng nào bao gồm các trình duyệt và các thiết bị di động. Nó là một nền tảng lý tưởng để xây dựng các ứng dụng RESTful trên. NET Framework

Các đặc điểm của ASP.NET WEB API

- Là một nền tảng lý tưởng để xây dựng dịch vụ RESTful
- Được xây dựng trên ASP.NET và hỗ trợ ASP.NET request / response pipeline
- Ánh xạ các hành động HTTP bởi Methods name.
- Hỗ trợ các dữ liệu trả về khác nhau (JSON, XML….)
- Có thể lưu trữ (host) trong IIS , hoặc chính nó hoặc các máy chủ khác có hỗ trợ .NET 4.0+
- ASP.NET Web API framework tích hợp HttpClient để giao tiếp với máy chủ Web API. HttpClient có thể được sử dụng từ phía máy chủ ASP.MVC, ứng dụng Windows Form, Console hoặc các ứng dụng khác.

ASP.NET Web API Versions

Web API Version	Supported .NET Framework	Coincides with	Supported in
Web API 1.0	.NET Framework 4.0	ASP.NET MVC 4	VS 2010
Web API 2 - Current	.NET Framework 4.5	ASP.NET MVC 5	VS 2012, 2013

DEMO tạo Web API Project

- Web API config
- Global.asax
- Web API Controller
- Web API

MISA®
TIN CẬY - TIỆN ÍCH - TẬN TÌNH

Configure Web API

Configure Web API

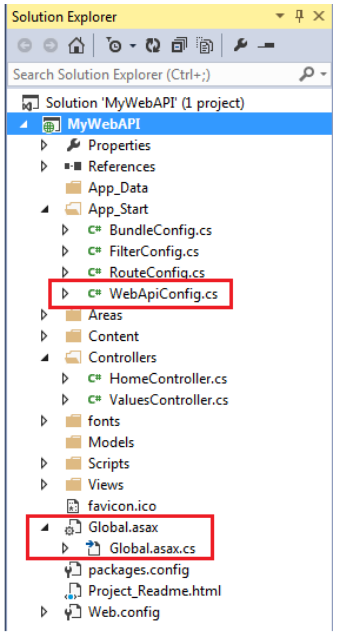
- Project Web API có sẵn class WebApiConfig mặc định trong thư mục App_Start và Global.asax:

Global.asax

```
public class WebAPIApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        GlobalConfiguration.Configure(WebApiConfig.Register);

        //other configuration
    }
}
```

WebApiConfig

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {

        config.MapHttpAttributeRoutes();

        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );

        // configure additional webapi settings here..
    }
}
```



HttpConfiguration

Property	Description
DependencyResolver	Gets or sets the dependency resolver for dependency injection.
Filters	Gets or sets the filters.
Formatters	Gets or sets the media-type formatters.
IncludeErrorDetailPolicy	Gets or sets a value indicating whether error details should be included in error messages.
MessageHandlers	Gets or sets the message handlers.
ParameterBindingRules	Gets the collection of rules for how parameters should be bound.
Properties	Gets the properties associated with this Web API instance.
Routes	Gets the collection of routes configured for the Web API.
Services	Gets the Web API services.

Web API Routing

Web API hỗ trợ 2 kiểu định tuyến:

- Convention-based Routing
- Attribute Routing

Convention-based Routing

Example: WebApiConfig with Default Route

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        // Enable attribute routing
        config.MapHttpAttributeRoutes();

        // Add default route using convention-based routing
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

Tạo route bằng phương pháp thủ công

Example: Add Default Route

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.MapHttpAttributeRoutes();

        // define route
        IHttpRoute defaultRoute = config.Routes.CreateRoute("api/{controller}/{id}",
                                                new { id = RouteParameter.Optional }, null);

        // Add route
        config.Routes.Add("DefaultApi", defaultRoute);

    }
}
```

MapHttpRoute()

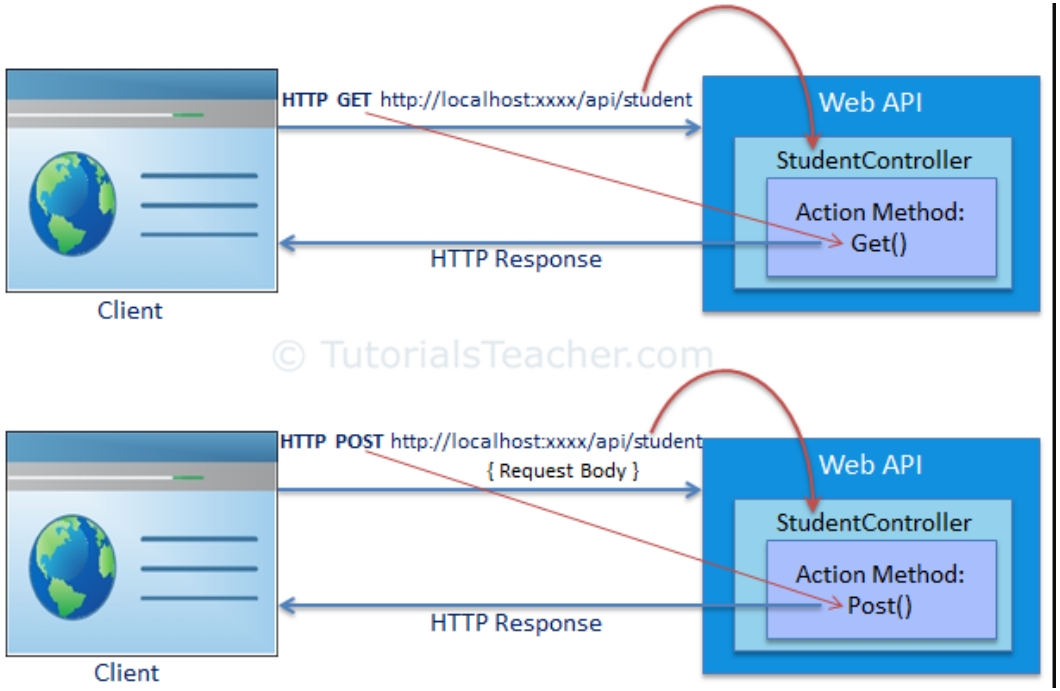
- List các tham số của phương thức MapHttpRouter:

Parameter	Description
name	Name of the route
routeTemplate	URL pattern of the route
defaults	An object parameter that includes default route values
constraints	Regex expression to specify characteristic of route values
handler	The handler to which the request will be dispatched.

Xem cách Web API xử lý các yêu cầu gửi đến và phản hồi của Http

Sample HTTP GET Request

```
GET http://localhost:1234/api/values/ HTTP/1.1
User-Agent: Fiddler
Host: localhost: 60464
Content-Type: application/json
```



Configure Multiple Routes

Example: Multiple Routes

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.MapHttpAttributeRoutes();

        // school route
        config.Routes.MapHttpRoute(
            name: "School",
            routeTemplate: "api/myschool/{id}",
            defaults: new { controller="school", id = RouteParameter.Optional }
            constraints: new { id ="/d+" }
        );

        // default route
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

Attribute Routing

- Attribute sử dụng [Route()] attibute để xác định tuyến đường.
- Có thể áp dụng cho bất cứ một **controller** hoặc phương thức hành động nào.

Attribute Routing

Để sử dụng Attribute routing, nó phải được kích hoạt trong WebApiConfig bằng cách gọi phương thức config.MapHttpAttributeRoutes().

Example: Attribute Routing

```
public class StudentController : ApiController
{
    [Route("api/student/names")]
    public IEnumerable<string> Get()
    {
        return new string[] { "student1", "student2" };
    }
}
```

Web API controller

- Web API controller là các class được tạo ra trong thư mục Controller hoặc thư mục khác trong thư mục gốc của Project.
- Tên phải kết thúc bằng **Controller**.
- **Phải using System.Web.Http.ApiController**
- Tất cả các phương thức public phải được gọi từ các action methods.
- Dựa trên các requests URL và các HTTP actions (**GET/ POST/PUT/PATCH/DELETE**). Web API sẽ quyết định và đưa ra các action methods tương ứng để thực thi.

Ví dụ

```
public class ValuesController : ApiController
{
    // GET api/values
    public IEnumerable<string> Get()
    {
        return new string[] { "value1", "value2" };
    }

    // GET api/values/5
    public string Get(int id)
    {
        return "value";
    }

    // POST api/values
    public void Post([FromBody]string value)
    {
    }

    // PUT api/values/5
    public void Put(int id, [FromBody]string value)
    {
    }

    // DELETE api/values/5
    public void Delete(int id)
    {
    }
}
```

Web API controller Base class

Handles Http GET request

http://localhost:1234/api/values

Handles Http GET request with query string

http://localhost:1234/api/values?id=1

Handles Http POST request

http://localhost:1234/api/values

Handles Http Put request

http://localhost:1234/api/values?id=1

Handles Http DELETE request

http://localhost:1234/api/values?id=1

Tạo các phương thức không bắt đầu bằng các HTTP verb

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace MyWebAPI.Controllers
{
    public class ValuesController : ApiController
    {
        [HttpGet]
        public IEnumerable<string> Values()
        {
            return new string[] { "value1", "value2" };
        }

        [HttpGet]
        public string Value(int id)
        {
            return "value";
        }

        [HttpPost]
        public void SaveNewValue([FromBody]string value)
        {
        }

        [HttpPut]
        public void UpdateValue(int id, [FromBody]string value)
        {
        }

        [HttpDelete]
        public void RemoveValue(int id)
        {
        }
    }
}
```

Các đặc điểm của Web API Controller

- Phải using lớp **System.Web.Http.ApiControlle**r
- Phải được tạo trong thư mục gốc của Project. (Nên tạo trong thư mục **Controllers** theo quy ước)
- Tên phương thức có thể giống hoặc bắt đầu với **Http Verb**. Hoặc có thể áp dụng các Http verb attributes cho các phương thức ([HttpGet], [HttpPut], [HttpPost], [HttpDelete])
- Kiểu trả về có thể là bất cứ kiểu nào.

Action Method Naming Conventions

HTTP Request Method	Possible Web API Action Method Name	Usage
GET	Get() get() GET() GetAllStudent() *any name starting with Get *	Retrieves data.
POST	Post() post() POST() PostNewStudent() *any name starting with Post*	Inserts new record.
PUT	Put() put() PUT() PutStudent() *any name starting with Put*	Updates existing record.
PATCH	Patch() patch() PATCH() PatchStudent() *any name starting with Patch*	Updates record partially.
DELETE	Delete() delete() DELETE() DeleteStudent() *any name starting with Delete*	Deletes record.

MISA®
TIN CẬY - TIỆN ÍCH - TẬN TÌNH

Parameter Binding

Parameter Binding

- Các ràng buộc mặc định:

HTTP Method	Query String	Request Body
GET	Primitive Type, Complex Type	NA
POST	Primitive Type	Complex Type
PUT	Primitive Type	Complex Type
PATCH	Primitive Type	Complex Type
DELETE	Primitive Type, Complex Type	NA

Phương thức với các tham số nguyên thủy

Example: Primitive Param

```
public class StudentContro
{
    public Student Get(int
    {

    }
}
```

Example: Multiple Parameters Binding

```
public class StudentController : ApiController
{
    public Student Get(int id, string name)
    {

    }
}
```

POST Action Method with Primitive Parameter

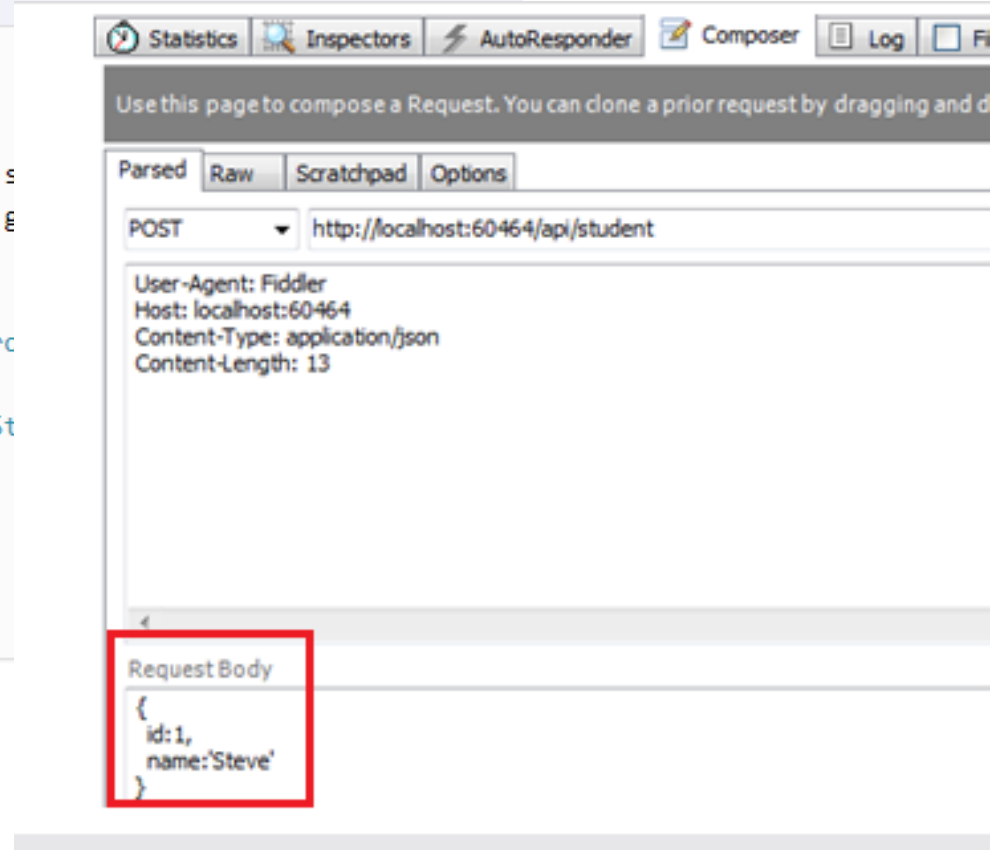
Example: Post Method with Primitive Parameter

```
public class StudentController : ApiController
{
    public Student Post(id id, string name)
    {

    }
}
```

Http Post với kiểu dữ liệu phức tạp

Example: Post Method with Complex Type Parameter

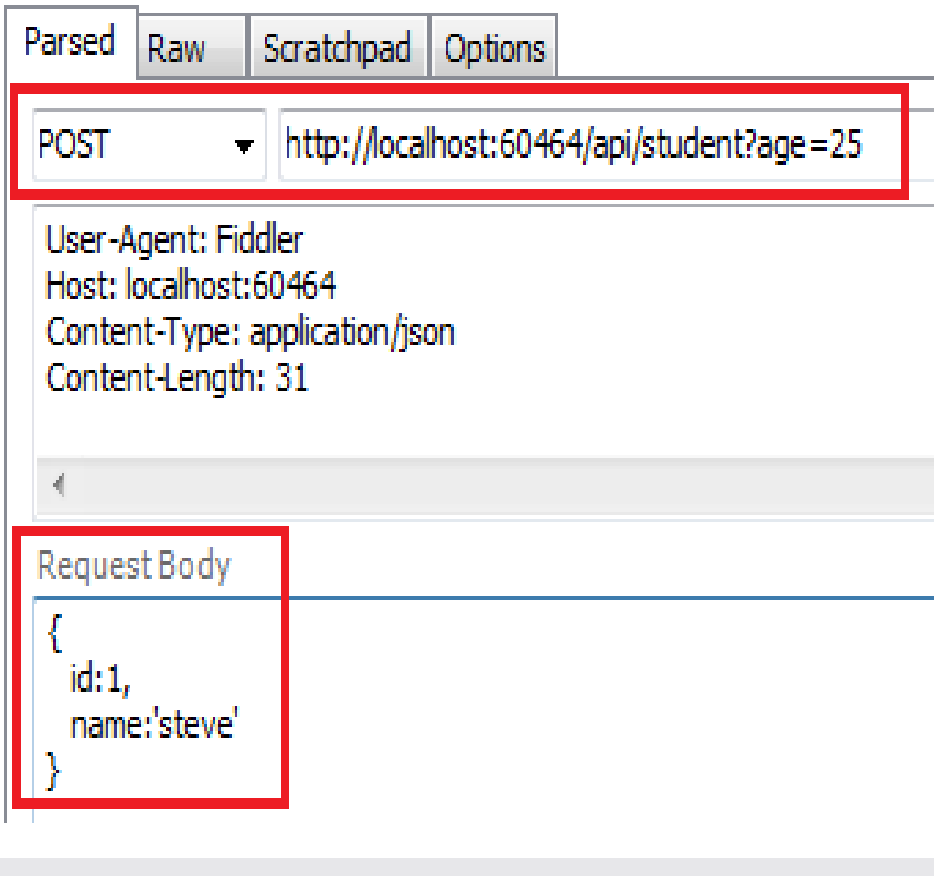
```
public class Student
{
    public int Id { get; s
    public string Name { g
}

public class StudentContro
{
    public Student Post(St
    {

    }
}
```



Phương thức POST với tham số hỗn hợp

Example: Post Method with F

```
public class Student
{
    public int Id { get; set;
    public string Name { get;

}

public class StudentControlle
{
    public Student Post(int a
    {

    }
}
```



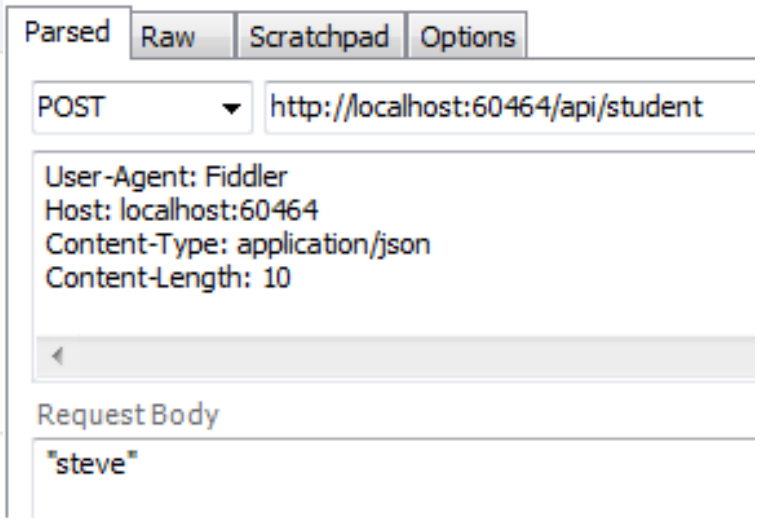
[FromUri] and [FromBody]

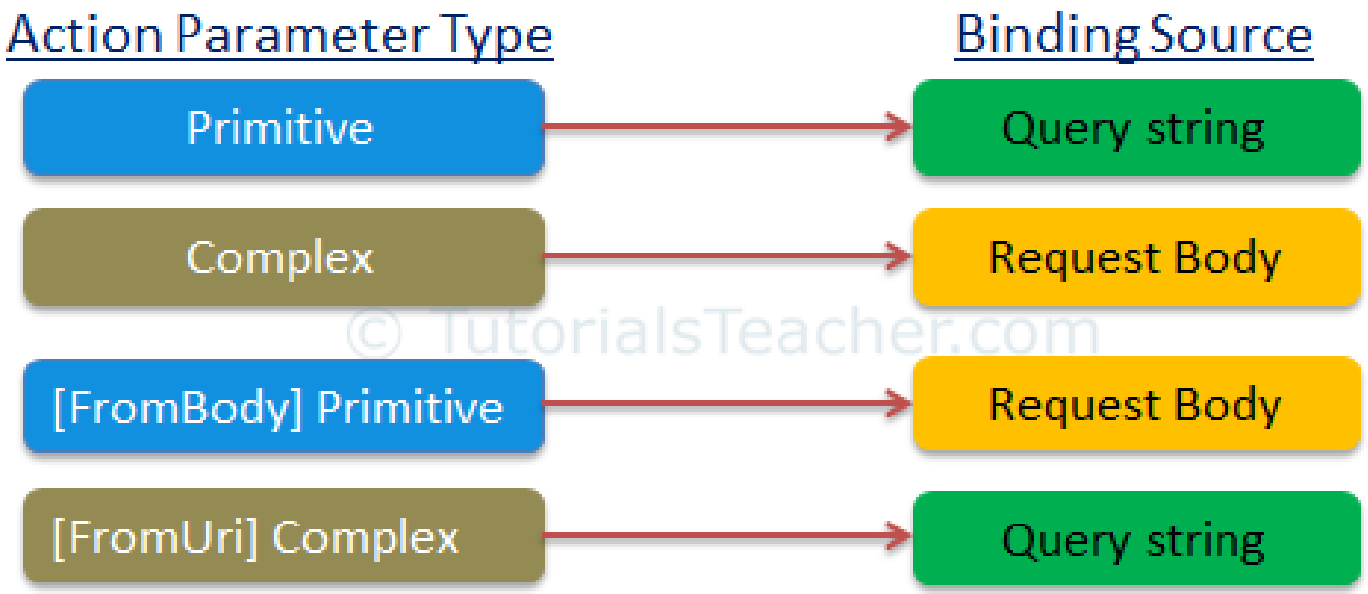
Example: FromUri

```
public class StudentController : ApiController
{
    public Student Post([FromUri]Student stud)
    {

    }
}
```

Example: FromBody

```
public class StudentController : ApiController
{
    public Student Post([FromBody]string name)
    {

    }
}
```





Web API Parameter Bindings

MISA®
TIN CẬY - TIỆN ÍCH - TẬN TÌNH

Action Method Return Type

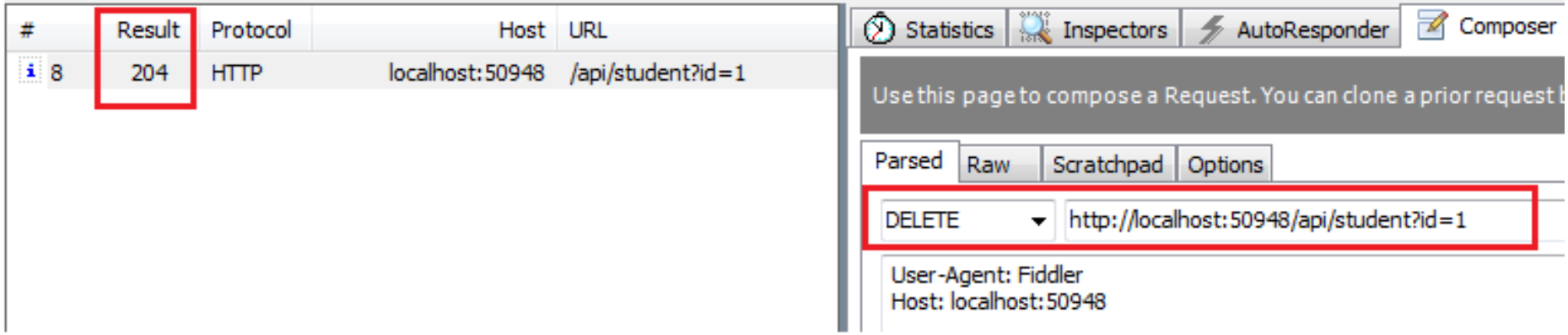
Action Method Return Type

- Void
- Primitive type or Complex type
- HttpResponseMessage
- IHttpActionResult

Void

Example: Void Return Type

```
public class StudentController : ApiController
{
```

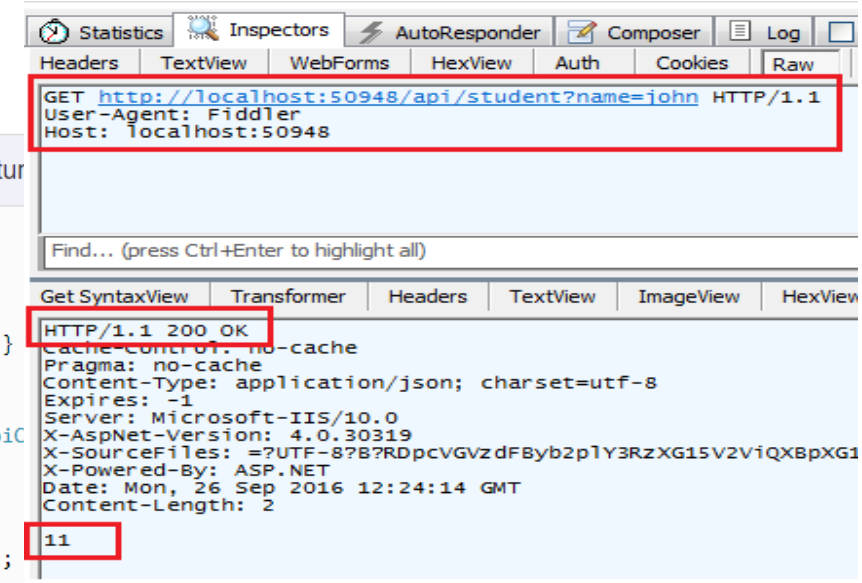


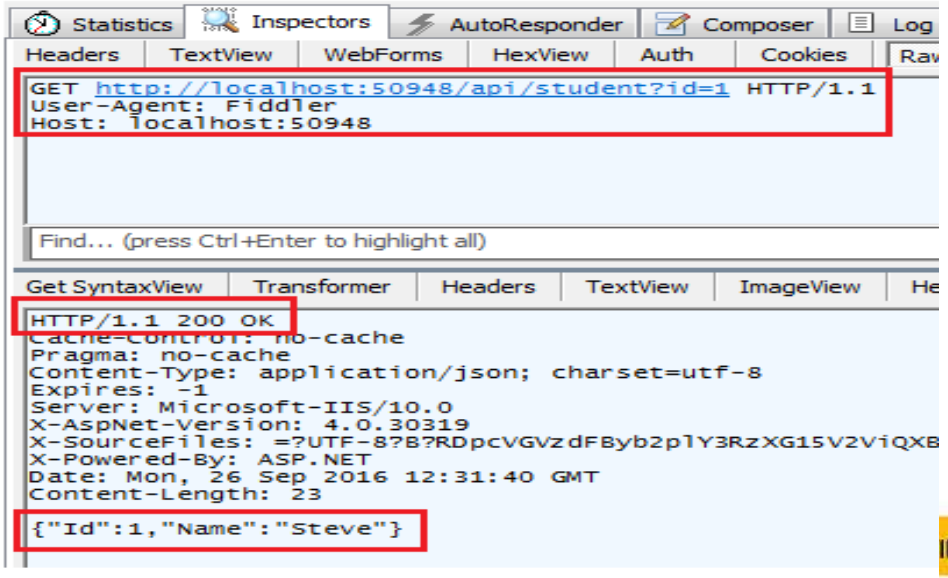
Void Response Status

Primitive or Complex Type

Example: Primitive or Complex Retur

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
}

public class StudentController : ApiC
{
    public int GetId(string name)
    {
        int id = GetStudentId(name);

        return id;
    }

    public Student
    {
        var studen

        return stu
    }
}
```





HttpResponseMessage

Web API Request Pipeline

HttpResponseMessage

Example: Return HttpResponseMessage

```
public HttpResponseMessage Get(int id)
{
    Student stud = GetStudentFromDB(id);

    if (stud == null) {
```

Statistics | Inspectors | AutoResponder | Composer | Log

Headers | TextView | WebForms | HexView | Auth | Cookies | Raw

```
GET http://localhost:50948/api/student?id=1 HTTP/1.1
User-Agent: Fiddler
Host: localhost:50948
```

Find... (press Ctrl+Enter to highlight all)

Get SyntaxView | Transformer | Headers | TextView | ImageView | He

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?RDpcVGVzdFByb2plY3RzXG15V2ViQXB
X-Powered-By: ASP.NET
Date: Mon, 26 Sep 2016 12:45:31 GMT
Content-Length: 23

{"Id":1,"Name":"Steve"}
```

```
StatusCode.OK, stud);
```

Statistics | Inspectors | AutoResponder | Composer | Log

Headers | TextView | WebForms | HexView | Auth | Cookies | Raw

```
GET http://localhost:50948/api/student?id=100 HTTP/1.1
User-Agent: Fiddler
Host: localhost:50948
```

Find... (press Ctrl+Enter to highlight all)

Get SyntaxView | Transformer | Headers | TextView | ImageView | Hex

```
HTTP/1.1 404 Not Found
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-SourceFiles: =?UTF-8?B?RDpcVGVzdFByb2plY3RzXG15V2ViQXB
X-Powered-By: ASP.NET
Date: Mon, 26 Sep 2016 12:42:02 GMT
Content-Length: 3

100
```

IHttpActionResult

Example: Return IHttpActionResult Type using Ok() and NotFound() Methods

```
public IHttpActionResult Get(int id)
{
    Student stud = GetStudentFromDB(id);

    if (stud == null)
    {
        return NotFound();
    }

    return Ok(stud);
}
```

IHttpActionResult

ApiController Method	Description
BadRequest()	Creates a BadRequestResult object with status code 400.
Conflict()	Creates a ConflictResult object with status code 409.
Content()	Creates a NegotiatedContentResult with the specified status code and data.
Created()	Creates a CreatedNegotiatedContentResult with status code 201 Created.
CreatedAtRoute()	Creates a CreatedAtRouteNegotiatedContentResult with status code 201 created.
InternalServerError()	Creates an InternalServerErrorResult with status code 500 Internal server error.
NotFound()	Creates a NotFoundResult with status code404.
Ok()	Creates an OkResult with status code 200.
Redirect()	Creates a RedirectResult with status code 302.
RedirectToRoute()	Creates a RedirectToRouteResult with status code 302.
ResponseMessage()	Creates a ResponseMessageResult with the specified HttpResponseMessage.
StatusCode()	Creates a StatusCodeResult with the specified http status code.
Unauthorized()	Creates an UnauthorizedResult with status code 401.

Create Custom Result Type

Example: Create Custom Result Type

```
public class TextResult : IHttpActionResult
{
    string _value;
    HttpRequestMessage _request;

    public TextResult(string value, HttpRequestMessage request)
    {
        _value = value;
        _request = request;
    }

    public Task<HttpResponseMessage> ExecuteAsyn
    {
        var response = new HttpResponseMessage()
        {
            Content = new StringContent(_value),
            RequestMessage = _request
        };
        return Task.FromResult(response);
    }
}
```

Example: Return Custom Result Type

```
public IHttpActionResult GetName(int id)
{
    string name = GetStudentName(id);

    if (String.IsNullOrEmpty(name))
    {
        return NotFound();
    }

    return new TextResult(name, Request);
}
```

MISA®
TIN CẬY - TIỆN ÍCH - TẬN TÌNH

Web API Request/Response Data Formats

Media Type

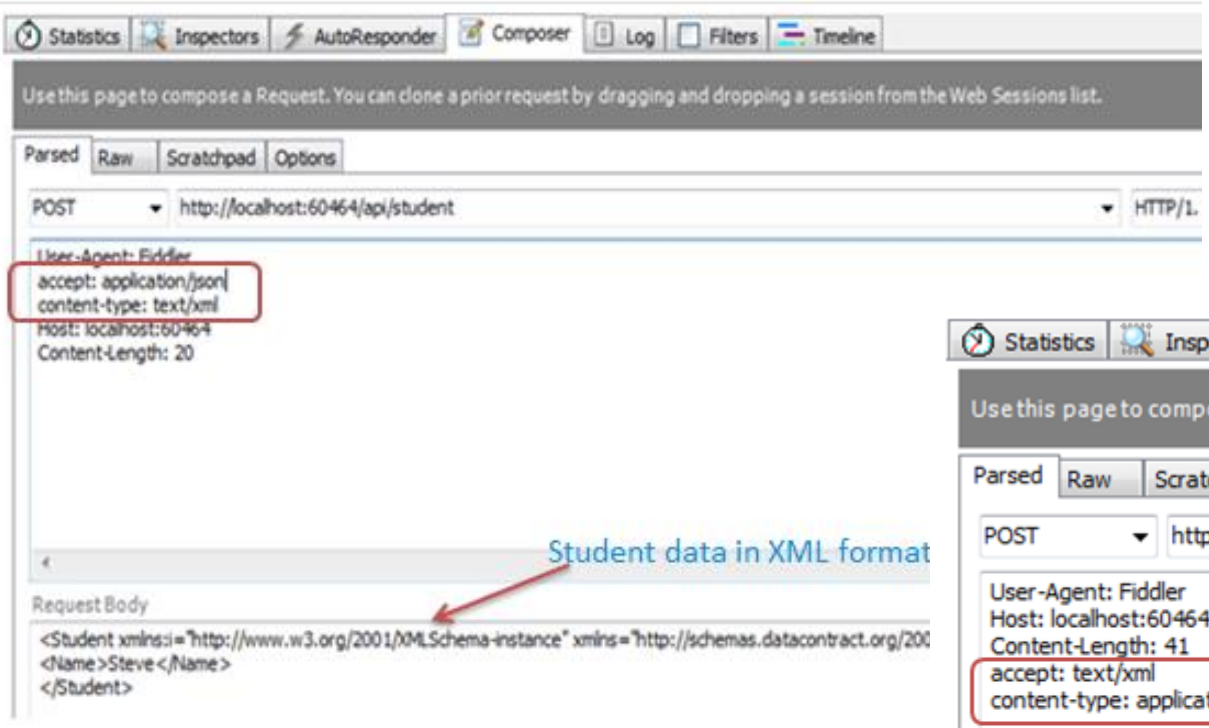
Chỉ định định dạng dữ liệu dưới dạng type/subtype:

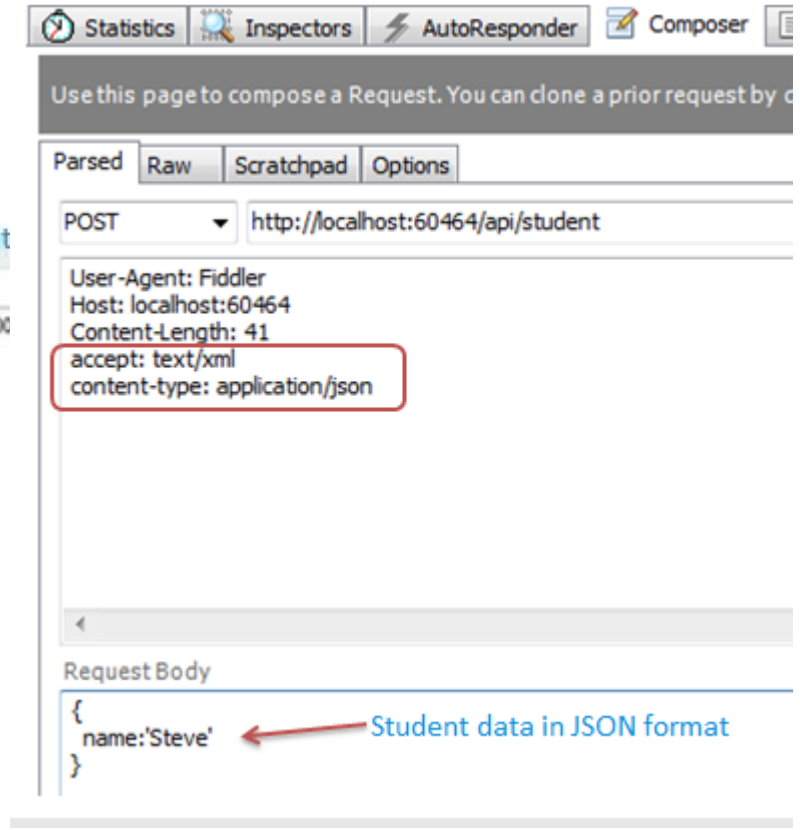
- text/html,
- ext/xml,
- application/json,
- image/jpeg
- ...

Media Type

Example: Post Action Method

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
}

public class StudentController : ApiController
{
    public Student Post(Student student)
    {
        // save student into db
        var insertedStudent = SaveStudent(student);

        return insertedStudent;
    }
}
```


Statistics
Inspectors
AutoResponder
Composer
Log
Filters
Timeline
Use this page to compose a Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.
Parsed
Raw
Scratchpad
Options
POST
http://localhost:60464/api/student
HTTP/1.
User-Agent: Fiddler
accept: application/json
content-type: text/xml
Host: localhost:60464
Content-Length: 20
Student data in XML format
Request Body
<Student xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/200
<Name>Steve</Name>
</Student>


Statistics
Inspectors
AutoResponder
Composer
Use this page to compose a Request. You can clone a prior request by
Parsed
Raw
Scratchpad
Options
POST
http://localhost:60464/api/student
User-Agent: Fiddler
Host: localhost:60464
Content-Length: 41
accept: text/xml
content-type: application/json
Request Body
{
name:'Steve'
}
Student data in JSON format

Media-Type Formatters

Media Type Formatter Class	MIME Type	Description
JsonMediaTypeFormatter	application/json, text/json	Handles JSON format
XmlMediaTypeFormatter	application/xml, text/json	Handles XML format
FormUrlEncodedMediaTypeFormatter	application/x-www-form-urlencoded	Handles HTML form URL-encoded data
JQueryMvcFormUrlEncodedFormatter	application/x-www-form-urlencoded	Handles model-bound HTML form URL-encoded data


THANK YOU