Logols Learning

WEEKEND WEB DEVELOPMENT BOOT CAMP

TRAINING: WEB API

REST

- Representational state transfer
- REST is protocol independent
- Architectural Style
  - Client-Server Communication
  - Stateless
  - Cacheable
  - Uniform Interface
    - Addressable Resources

HTTP

- Many REST services now use HTTP
- This is Hypertext Transfer Protocol
- What web pages use
- Request and Response
- Stateless
- Requests: GET, POST, PUT, DELETE

Controller

- The request initially goes to the controller
- Based upon the route a method is run.
- The JSON body of the request can be bound to a model.
- Whatever is returned from the method is returned in JSON.

Attributes

- Allow for declarations on classes, methods, or properties.
- Give special behavior or properties.
- [HttpGet], [HttpPost], [HttpPut], [HttpDelete]
- Example:

```
[HttpPost]
public void Post([FromBody]Answer answer)
{
   _service.Insert(answer);
}
```

Routing

- Tells ASP.Net which controller should receive the request
- Based on the pattern of the request
- Default Route: “http://domain/api/{controller}/{action}/{id}”

URL	Http	Controller	Method	Parameter
http://www.google.com/Employee/	Get	Employee	Get	None
http://www.google.com/Employee/1	Get	Employee	Get	id = 1
http://www.google.com/Employee/	Post	Employee	Post	In Request Body
http://www.google.com/Employee/	Put	Employee	Put	In Request Body
http://www.google.com/Employee/1	Delete	Employee	Delete	id = 1

Class Attribute Route

- Controller Names End With “Controller”
  - This is left off of URL for routing
- Class Attribute Route Ex:

```
[Route("api/[controller]")]
public class TimeTravelerController : Controller
```

JSON

- JavaScript Object Notation
- How an Object is represented in JavaScript.

Example:

```
{
   ‘studentID’: 0
   ‘firstName’: ‘Kathy’,
   ‘lastName’: ‘Smith’
}
```

Response Binding

- Return value to JSON Data
- Example:

```
[HttpGet]
public IEnumerable<TimeTraveler> Get()
{
    return timeTravelerRepository.GetAll();
}
```

Parameter Binding

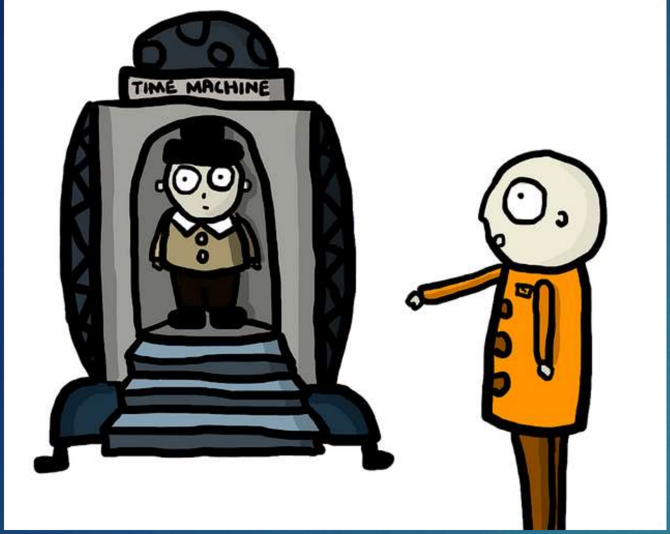
- JSON in Request Body converted to Class
- Example:

```
[HttpPost]
public void Post([FromBody]Answer answer)
{
   _service.Insert(answer);
}
```

CLI Commands

- mkdir – Create Directory
- cd – Change Directory
- Add project:
  - dotnet new classlib
  - dotnet new webapi
- Add reference:
  - dotnet add reference [path]/[name.csproj]
  - dotnet add package Dapper
  - dotnet add package MySql.Data

Postman

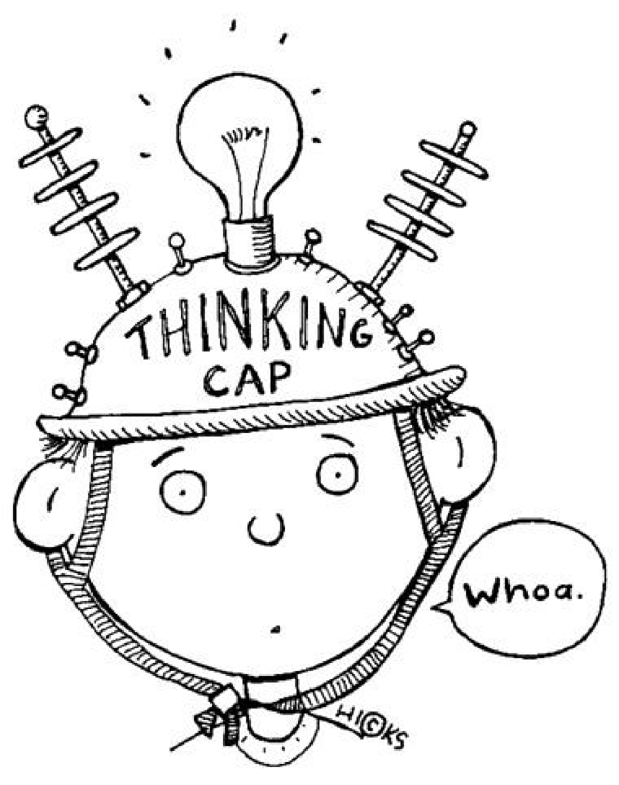


EXAMPLE

ASP.NET WEB API

ASSESSMENT

ASP.NET WEB API



Assignment

- A small internet has been brought back online. Everyone wants your status report.
- They also want to be able to insert, update, and delete data.
- Add a web api to get, insert, update, and delete person status.

QUICK REVIEW

WEB API

Not really a sign you'd want to see whilst driving through an eerily quiet neighbourhood...

Additional Resources

- Microsoft Page
  - https://www.asp.net/web-api
- Microsoft Tutorial
  - https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api
- PluralSight
  - https://app.pluralsight.com/player?author=jon-flanders&name=aspnetwebapi-m1-introduction&mode=live&clip=0&course=aspnetwebapi

Keep Practicing!

- Try creating different web api’s.
- Use the different http actions.
- Hook it up to the database.