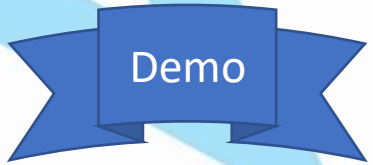
Module 2 Day 12

Consuming APIs - Part 2:
POST, PUT and DELETE

HTTP Methods (Verbs)

Method	Definition
GET	Read and return the specified resource. Requests using GET should only retrieve data, never change server data.
POST	Create a new resource, often causing a change in state on the server. Not idempotent (duplicate calls might cause two new resources).
PUT	Update the specified resource with the request payload. Idempotent (duplicate calls may be made without side effects).
DELETE	Remove the specified resource from the server.



HTTP Methods (Verbs)

Method	URL	Req Body	Resp Body	Example

GET	http://api.com/hotels	None	JSON array of objects	`RestRequest request = new RestRequest("hotels");`
				`IRestResponse<List<Hotel>> response =`
				`client.Get<List<Hotel>>(request);`

GET	http://api.com/hotels/101	None	JSON object	`RestRequest request = new RestRequest($"hotels/{id}");`
				`IRestResponse<Hotel> response =`
				`client.Get<Hotel>(request);`

POST	http://api.com/reservations	JSON object	None or JSON object	`RestRequest request = new RestRequest("reservations");`
				`request.AddJsonBody(newReservation);`
				`IRestResponse<Reservation> response =`
				`client.Post<Reservation>(request);`

PUT	http://api.com/reservations/10	JSON object	None or JSON object	`RestRequest request = new`
				`RestRequest($"reservations/{res.Id}");`
				`request.AddJsonBody(res);`
				`IRestResponse<Reservation> response =`
				`client.Put<Reservation>(request);`

DELETE	http://api.com/reservations/11	None	None	`RestRequest request = new`
				`RestRequest($"reservations/{reservationId}");`
				`IRestResponse response = client.Delete(request);`

Error Handling

- IRestResponse.***ResponseStatus***
  - Tells whether there was a complete request-response cycle
- IRestResponse.***IsSuccessful***
  - Status Code falls between 200-299
  - Status code ranges (w3schools)
- IRestResponse.***StatusCode***
  - Gives the System.Net.HttpStatusCode enum

```
namespace RestSharp
{
    //
    // Summary:
    //     Status for responses (surprised?)
    public enum ResponseStatus
    {
        None = 0,
        Completed = 1,
        Error = 2,
        TimedOut = 3,
        Aborted = 4
    }
}
```

```
if (response.ResponseStatus != ResponseStatus.Completed)
{
    throw new Exception("Error occurred - unable to reach server.");
}
else if (!response.IsSuccessful)
{
    throw new Exception($"Error occurred - received non-success response: {(int)response.StatusCode} {response.StatusCode}");
}
```