

## RESTful Convention

Request	Description
<b>GET</b> /api/customers	Get all customers
<b>GET</b> /api/customers/1	Get customer with ID 1
<b>POST</b> /api/customers	Add a new customer (customer data in the request body)
<b>PUT</b> /api/customers/1	Update customer with ID 1 (customer data in the request body)
<b>DELETE</b> /api/customers/1	Delete customer with ID 1

## Building an API

```
public IHttpActionResult GetCustomers() {}
```

```
[HttpPost]
```

```
public IHttpActionResult CreateCustomer(CustomerDto customer) {}
```

```
[HttpPut]
```

```
public IHttpActionResult UpdateCustomer(int id, CustomerDto  
customer) {}
```

```
[HttpDelete]
```

```
public IHttpActionResult DeleteCustomer(int id) {}
```

## Helper methods

- NotFound()
- Ok()
- Created()
- Unauthorized()

## AutoMapper

Create a mapping profile first:

```
public class MappingProfile : Profile
{
    public MappingProfile()
    {
        Mapper.CreateMap<Customer, CustomerDto>();
    }
}
```

Load the mapping profile during application startup (in global.asax.cs):

```
protected void Application_Start()
{
    Mapper.Initialize(c => c.AddProfile<MappingProfile>());
}
```

To map objects:

```
var customerDto = Mapper.Map<Customer, CustomerDto>(customer);
```

# Building Web APIs

By: Mosh Hamedani

Or to map to an existing object:

```
Mapper.Map(customer, customerDto);
```

## Enabling camel casing

In WebApiConfig:

```
public static void Register(HttpConfiguration config)
{
    var settings =
config.Formatters.JsonFormatter.SerializerSettings;

    settings.ContractResolver = new
CamelCasePropertyNamesContractResolver();

    settings.Formatting = Formatting.Indented;
}
```