Adding Client-side Functionality

Gill Cleeren
ARCHITECT

@gillcleeren   www.snowball.be

Overview

Adding client-side libraries with Bower
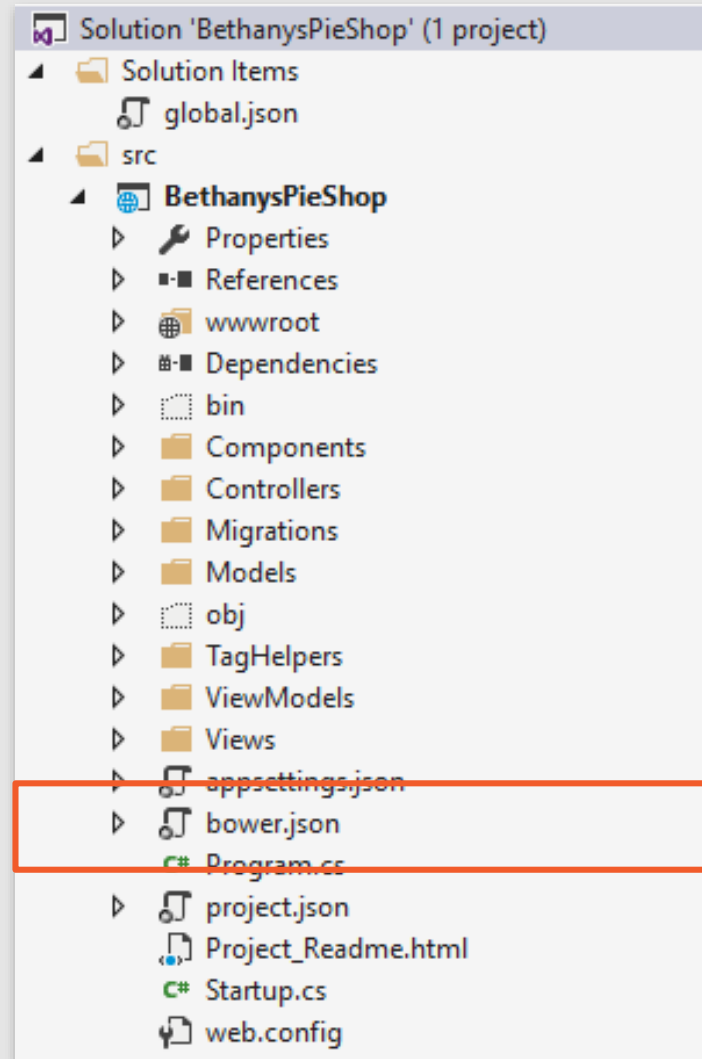
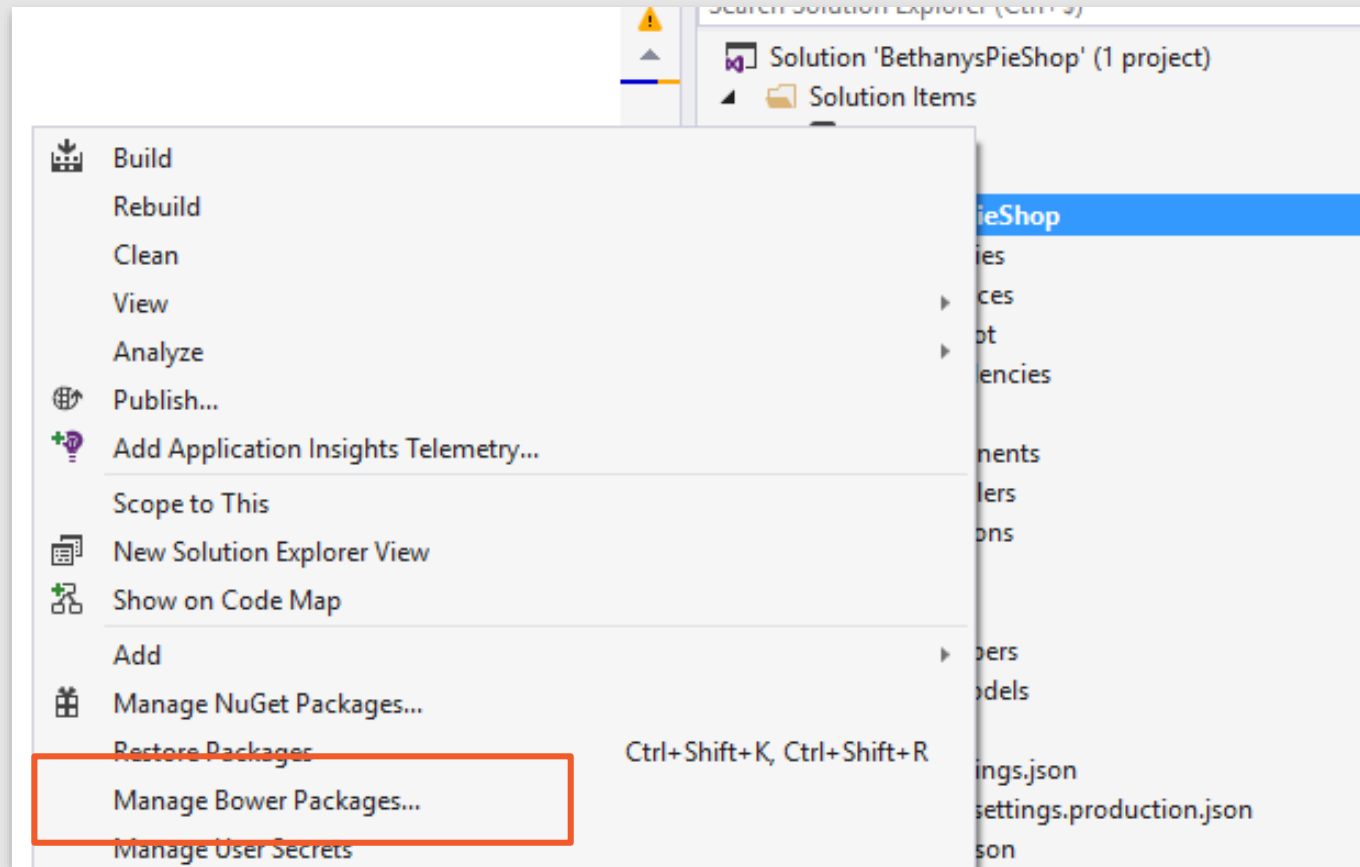Creating an API controller

Adding Client-side Libraries with Bower

bower.json

Managing Bower Packages

Managing Bower Packages

```json
{
  "name": "asp.net",
  "private": true,
  "dependencies": {
    "bootstrap": "3.3.7",
    "jquery": "3.1.1",
  }
}
```

Adding Support for jQuery

wwwroot
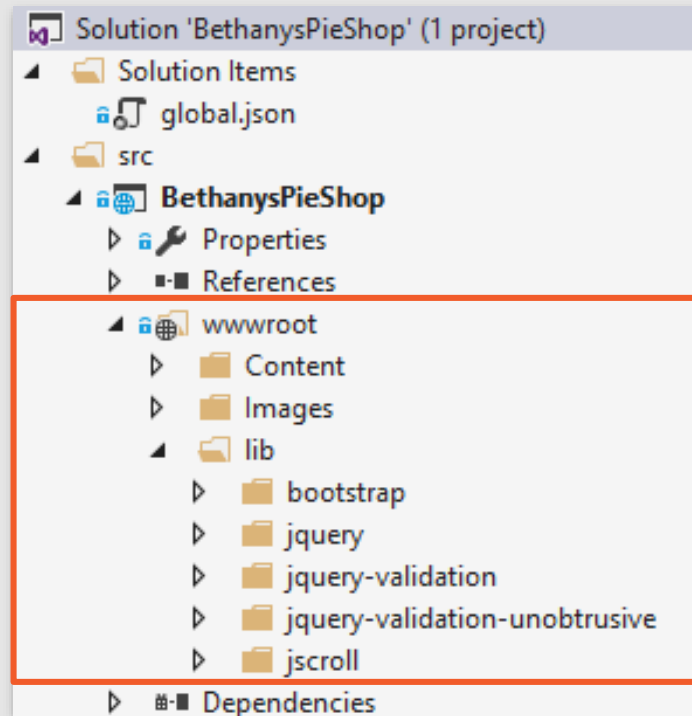
Changes in _layout.cshtml

```html
<html>
<head>
    <script src="~/lib/jquery/dist/jquery.js"></script>
</head>
<body>
  ...
    @RenderSection("scripts", required: false)
</body>
</html>
```

Adding a Scripts Section

```
@section scripts{

    <script>

        $(function () {

            ...

        });

    </script>

}
```

Demo

Adding jQuery to our site

Creating an API Controller

API Controllers

Data without HTML   **REST-based**   **HTTP**

API Controllers

`/api/pies/22`
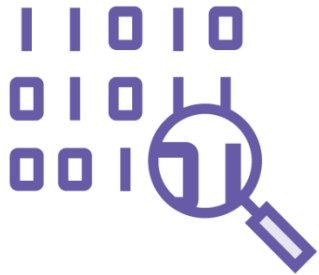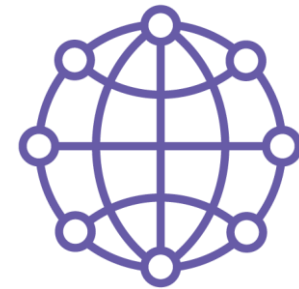
GET
POST
PUT
DELETE

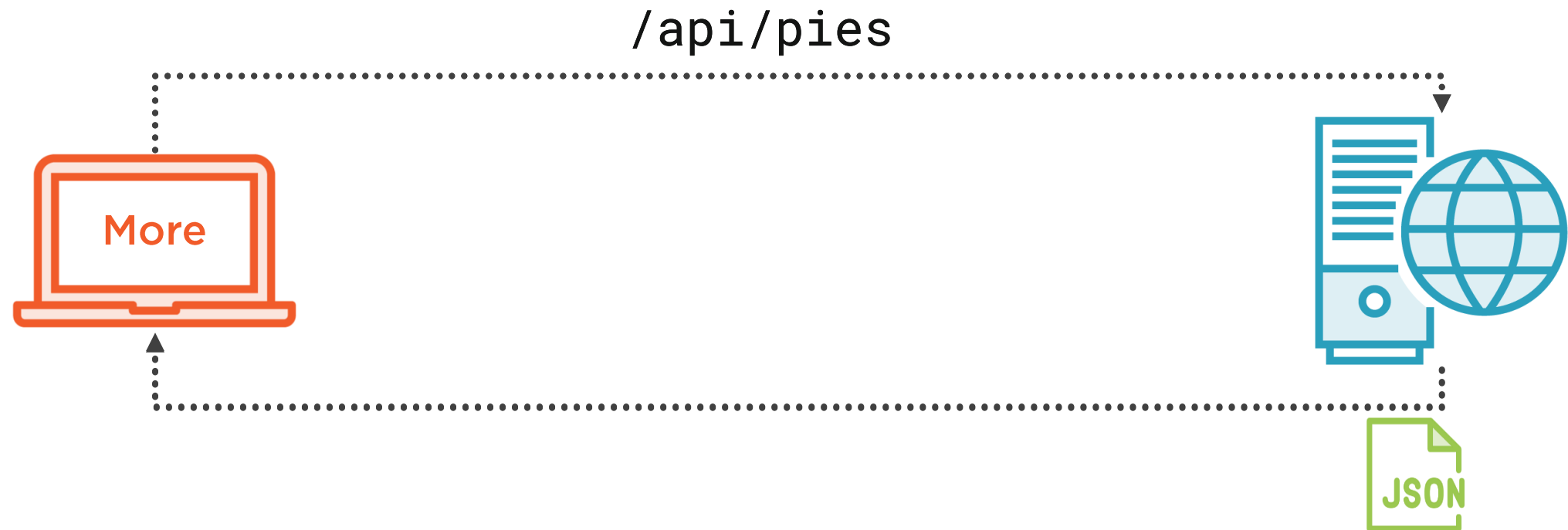# A Basic API Controller

```csharp
[Route("api/[controller]")]
public class PieDataController : Controller
{

    private IPieRepository _pieRepository;


    [HttpGet]
    public IEnumerable<PieViewModel> LoadMorePies() //api/PieData
    { ... }

    [HttpGet("{id}")]
    public PieViewModel LoadPieById(int id) //api/PieData/id
    { ... }

}
```

Async Operations

`/api/pies`

Demo

Creating the API controller

Creating a Load More function with jQuery and Ajax

https://www.pluralsight.com/courses/ asp-dotnet-core-api-building-first

Summary

Bower is the way forward to manage client-side libraries

Web API now based on same controller type

Calls can be made asynchronously from client-side script