

# Overview



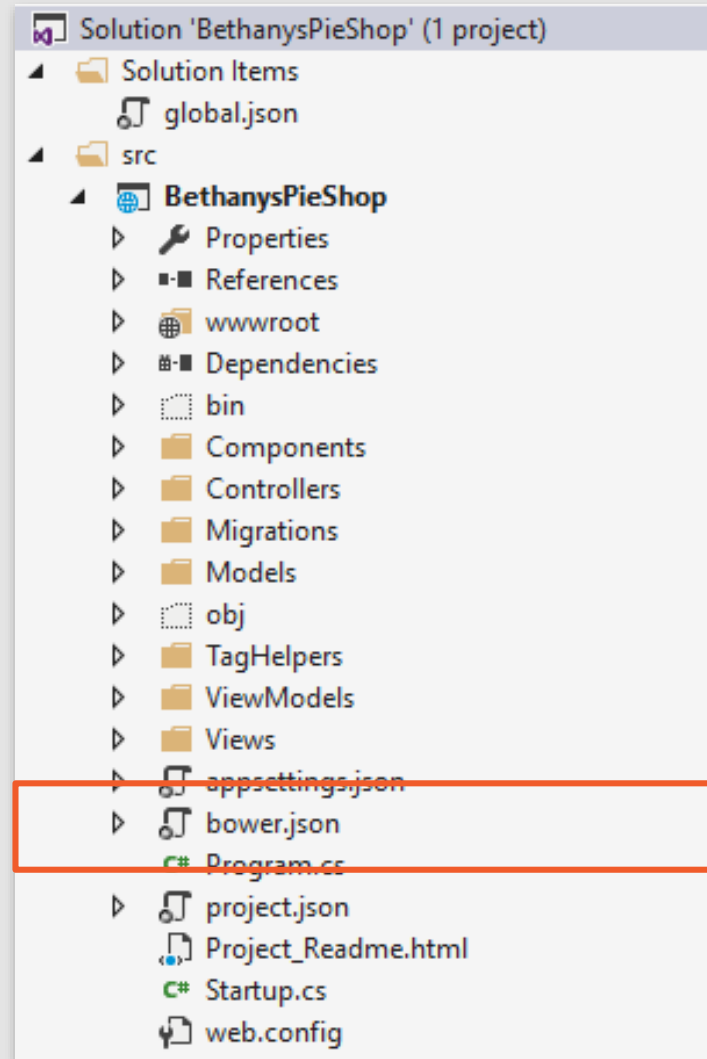
Adding client-side libraries with Bower

Creating an API controller

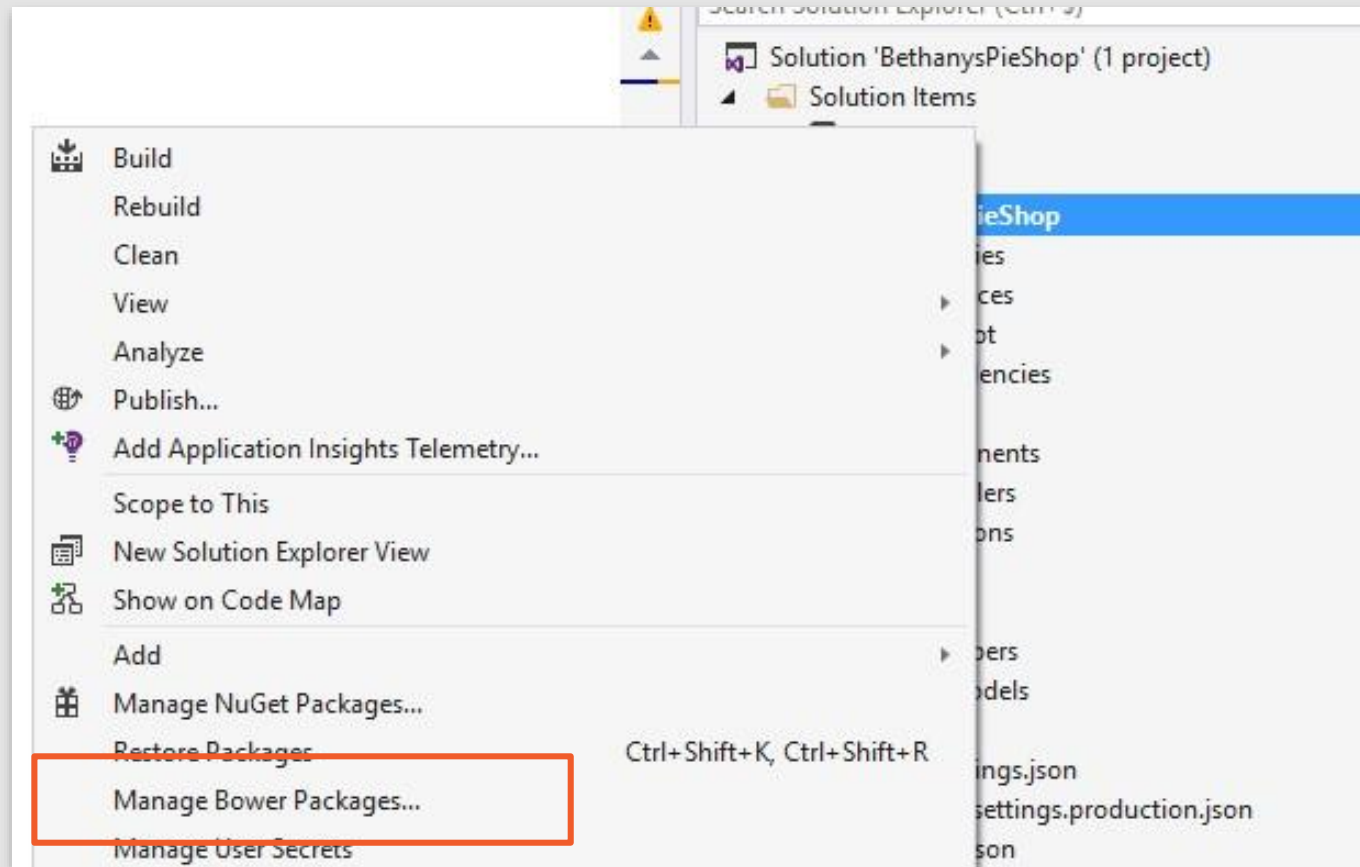
# Adding Client-side Libraries with Bower

---

# bower.json



# Managing Bower Packages



# Managing Bower Packages

Bower: BethanysPieShop × bower.json

Manage Bower Packages: 'BethanysPieShop' Project

Browse


Installed

Update Available

Search (Ctrl+E)

🔍

☐ Include Prerelease




jquery

JavaScript library for DOM...


Uninstall

Update




bootstrap

The most popular front-end framework for developing r...




angular

HTML enhanced for web apps




font-awesome

Font Awesome




animate.css

A cross-browser library of CSS animations. As easy to us...




reveal.js

The HTML Presentation Framework



impress.js

It's a presentation framework based on the power of CS...



jquery

Uninstall

Update

2.2.3

Options

☒ Save changes to bower.json

Description

JavaScript library for DOM operations

Author(s): jQuery Foundation and other contributors

License: MIT

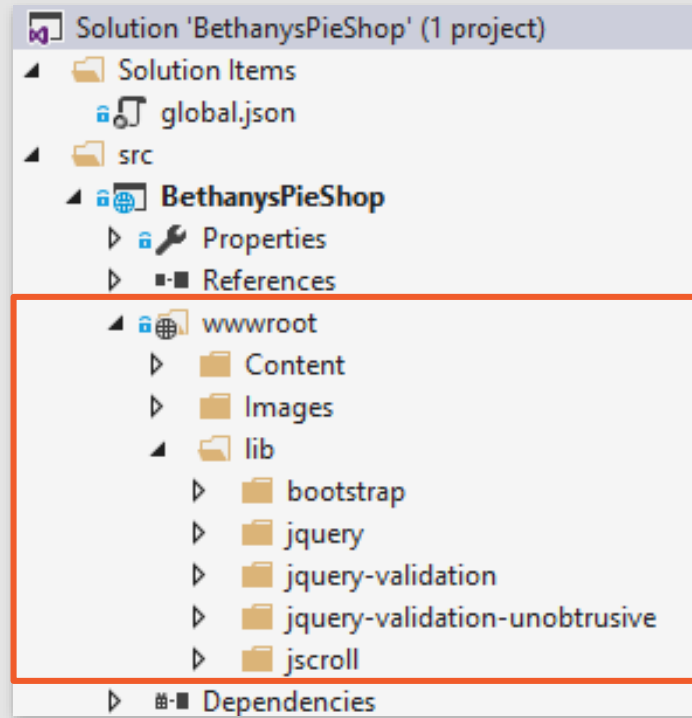
Downloads: 0

Project URL: <https://jquery.com>

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

## Adding Support for jQuery

# wwwroot



# Changes in \_layout.cshtml

```
<html>
```

```
<head>
```

```
    <script src="~/lib/jquery/dist/jquery.js"></script>
```

```
</head>
```

```
<body>
```

```
    ...
```

```
    @RenderSection("scripts", required: false)
```

```
</body>
```

```
</html>
```



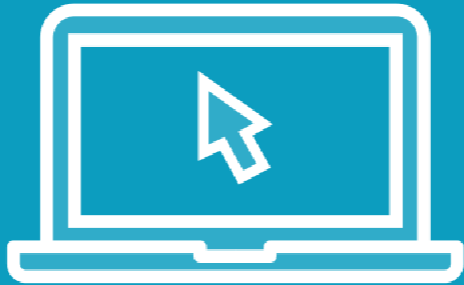


# Adding a Scripts Section

```
@section scripts{  
    <script>  
        $(function () {  
            ...  
        });  
    </script>  
}
```



# Demo

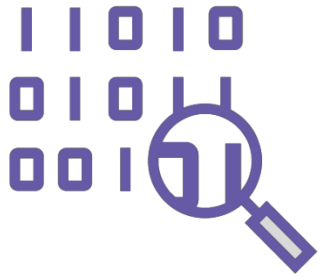


## Adding jQuery to our site

# Creating an API Controller

---

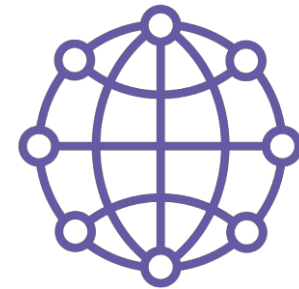
# API Controllers



Data without HTML



REST-based



HTTP

# API Controllers



# A Basic API Controller

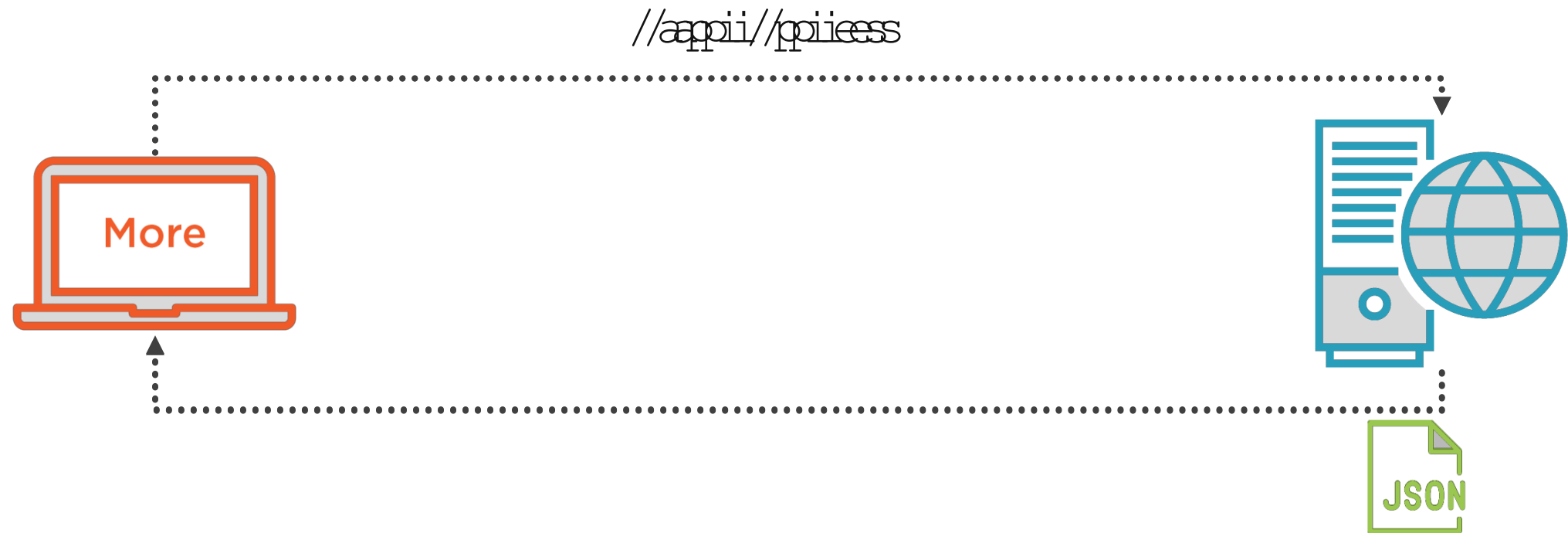
```
[Route("api/[controller]")]
public class PieDataController : Controller
{
    private IPieRepository _pieRepository;

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

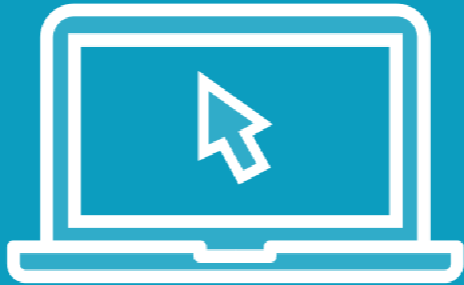
    [HttpGet("{id}")]
    public PieViewModel LoadPieById(int id) //api/PieData/id
    { ... }
}
```



# Async Operations



# Demo



Creating the API controller

Creating a Load More function with  
jQuery and Ajax



# 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**