

Writing Integration Tests for ASP.NET Core Web APIs: Part 1



Steve Gordon

MICROSOFT DEVELOPER TECHNOLOGIES MVP

@stevejgordon www.stevejgordon.co.uk



Overview



Testing GET requests

Asserting on JSON responses

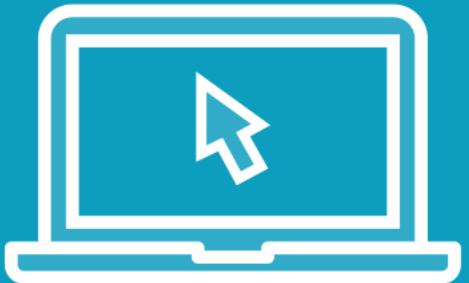
Creating test clients

Simplifying test code

Testing response headers



Demo



Test a GET request

- Test routing
- Test business logic
- Assert on HTTP responses



Achievements



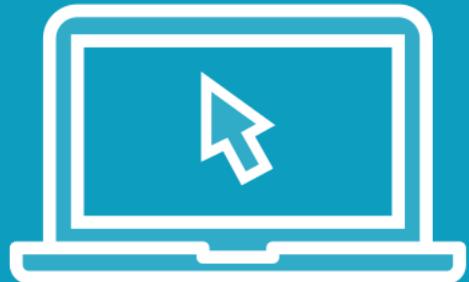
Tested configuration and use of ASP.NET Core features

Tested an action method and business logic within its dependency

We have tests which will warn us of breaking changes in our API



Demo

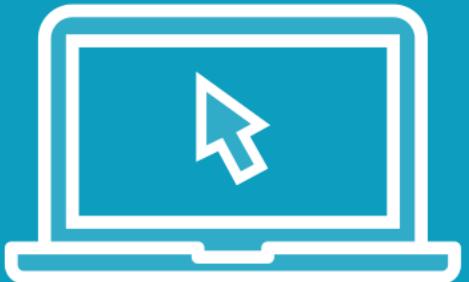


Discuss why the current response content test is brittle

Refactor the test to improve JSON content assertions



Demo



Using CreateDefaultClient

Using CreateClient

Configuring the test client



CreateDefaultClient

Creates a new instance of HttpClient that can be used to send a HttpRequestMessage to the server. By default, the base address of the HttpClient instance will be set to 'http://localhost'.



CreateClient

Creates an instance of HttpClient that automatically follows redirects and handles cookies. The HttpClient instance is configured using values configured in WebApplicationFactory.ClientOptions.



Demo



Simplifying response assertions

- Reducing code verbosity



{JSON}

GetFromJsonAsync

- Uses case-insensitive deserialization
- Validates response has success status code
- Validates Content-Type header
- Validates response includes content



`application/json`

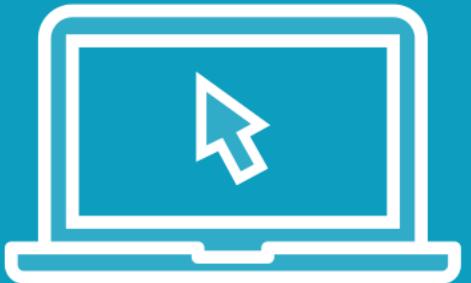
`application/problem+json`

JSON Media Types

GetFromJsonAsync supports additional JSON media types.



Demo



Validating response headers

- Updating the API to send a cache header
- Writing a test to cover the requirement



Summary



- Wrote tests for a GET endpoint**
- Used integration tests to confirm components work together**
- Covered techniques to assert expectations for the response content**
- Learned about test client factory methods**
- Wrote a test for the presence of a Cache-Control header**



Up Next:
Applying Test Driven Development (TDD)
with Integration Tests

