Workshop 3 – Building a RESTful API

Agenda

- Create a RESTful API backend

Couchbase
DEVELOPER COMMUNITY

©2016 Couchbase Inc.

Platforms

- .NET - ASP.NET WebAPI
- Or .NET Core – ASP.NET Core WebAPI
- Couchbase (document database)

Couchbase
DEVELOPER COMMUNITY

©2016 Couchbase Inc.

ASP.NET

- Relies on IIS
- Controllers contain methods
- Global.asax.cs

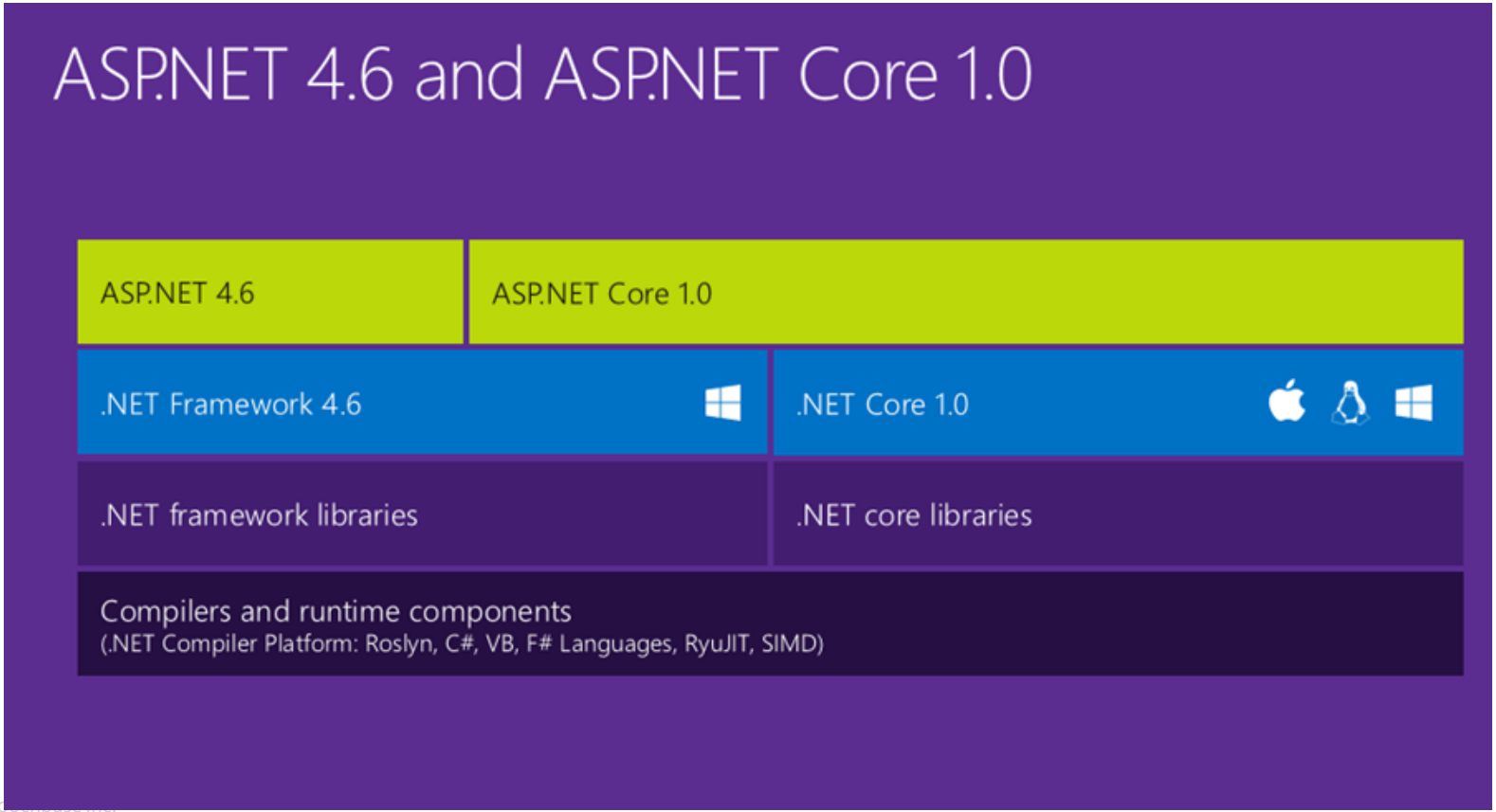
©2016 Couchbase Inc.

ASP.NET Core

- Runs as a "command line" app, uses Kestrel server
- Controllers contain methods
- Can run on .NET or .NET Core

©2016 Couchbase Inc.

ASP.NET vs ASP.NET Core



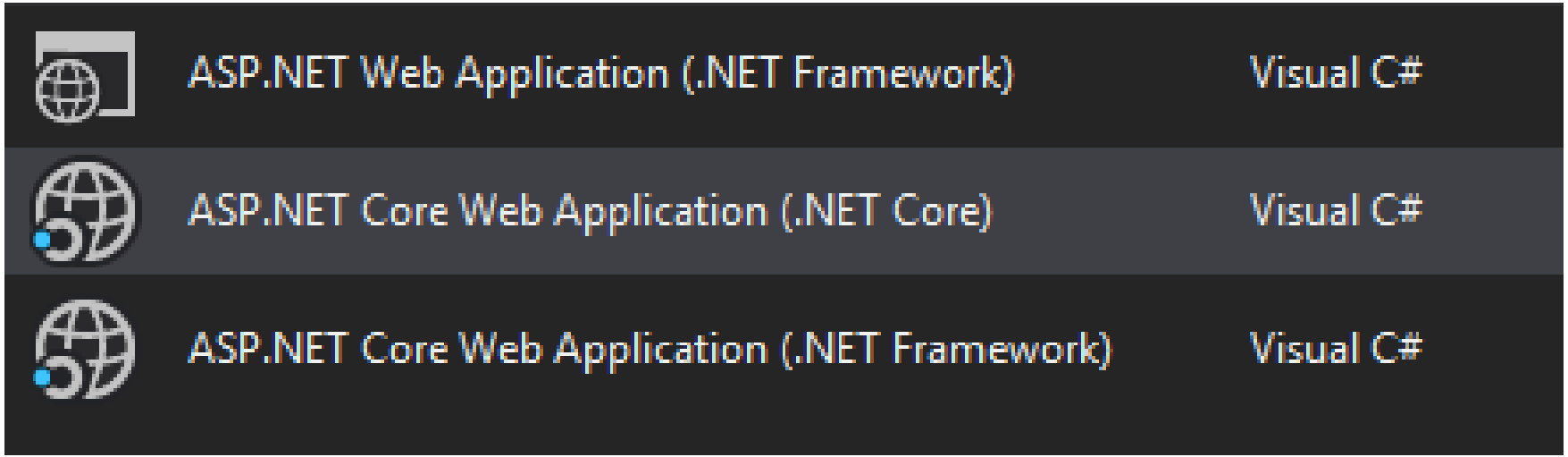
©2016 Co

ASP.NET vs ASP.NET Core

	ASP.NET	ASP.NET Core
Cross platform?	No	Yes
Can use Kestrel?	No	Yes
Can use IIS?	Yes (required)	Yes
Works on Full .NET?	Yes	Yes
Works on .NET Core?	No	Yes

©2016 Couchbase Inc.

ASP.NET vs ASP.NET Core



©2016 Couchbase Inc.

WebAPI

- There are a lot of similarities between ASP.NET and ASP.NET Core
- Similar to MVC:
  - Controllers: classes that inherit from a special base class
  - Methods: endpoints
  - Optional async/await

©2016 Couchbase Inc.

ASP.NET WebAPI vs ASP.NET Core WebAPI

- WebAPI/MVC convergence
- CORS
- Configuration
- Startup
- Static files
- Casing / JsonProperty

©2016 Couchbase Inc.

Core WebAPI Convergence

- In ASP.NET Core, MVC and WebAPI converge
- Instead of Controller and ApiController, there is just Controller
- Instead of IHttpActionResult (WebAPI) there is just IActionResult

©2016 Couchbase Inc.

CORS

- Cross-Origin Resource Sharing

- ASP.NET
  - `[EnableCors(origins: "*", headers: "*", methods: "*")]`

- ASP.NET Core
  -
```
app.UseCors(builder => builder
                .AllowAnyHeader()
                .AllowAnyMethod()
                .AllowAnyOrigin());
```

©2016 Couchbase Inc.

Startup

- ASP.NET
  - Global.asax.cs
  - `System.Web.HttpApplication`
  - `protected void Application_Start()`

- ASP.NET Core
  - Program.cs – console application
  - Startup.cs – used by Program.cs
  - `public void ConfigureServices(IServiceCollection services)`
  - `public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)`

©2016 Couchbase Inc.

Configuration

- ASP.NET
  - Web.config (XML)
  - ConfigurationManager.AppSettings
- ASP.NET Core
  - appsettings.json (JSON)

```
IConfigurationSection settingsSection = Configuration.GetSection("MySettings");
MySettings settings = settingsSection.Get<MySettings>();
services.Configure<MySettings>(settingsSection);
```

©2016 Couchbase Inc.

Static Files

- ASP.NET
  - Put static files wherever
- ASP.NET Core
  - Put static files in wwwroot folder

©2016 Couchbase Inc.

Casing / JsonProperty

- ASP.NET
  - `return Ok(result.Value);`
  - Result object serialized
  - Resultant JSON is PascalCased

- ASP.NET Core
  - `return Ok(result.Value);`
  - Result object serialized
  - Resultant JSON is camelCased
  - Use `[JsonProperty("FirstName")]` to PascalCase

©2016 Couchbase Inc.

Questions

Exercise: Fill in the blanks

- Fill in the blanks to make the application work
  - dotnet_workshop or dotnetcore_workshop
  - Completed versions are in dotnet, dotnetcore
- Checkout the code from Github
  - https://github.com/couchbaselabs/aspnet-nosql-workshop/tree/master/03
- The source code is also available on USB sticks
- You can test with Postman / Fiddler / curl

©2016 Couchbase Inc.

ASP.NET

- dotnet_workshop
- Look for TODOs
- Files included:
  - CouchbaseConfig.cs
  - PersonController.cs
  - Web.config

©2016 Couchbase Inc.

ASP.NET Core

- dotnetcore_workshop
- Look for TODOs
- Files included:
  - appsettings.json
  - PersonController.cs
  - Startup.cs

©2016 Couchbase Inc.

How to execute

Execute RESTful API backend:

- Visual Studio: F5 (or ctrl+f5)
- `dotnet run` from command line

©2016 Couchbase Inc.

Exercise: Getting Started

At the end of the lab, your app should be able to <u>list, add, edit, and delete.</u>

If you have questions or are running into a problem, I'll be walking around helping you individually.

©2016 Couchbase Inc.