ASP.NET Core 2.1

Daniele Bochicchio
Microsoft Regional Director, ASP.NET MVP
daniele@aspitalia.com
@dbochicchio

Matteo Tumiati
Windows Development MVP
matteot@icubed.it
@xTuMiOx

Microsoft
Regional Director

aspitalia.com

icubed

CommunityDays.it

Agenda

- Recap
- Da ASP.NET 2.0 a 2.1
- Performance di .NET Core 2.1
- Performance per HTTPs, IIS e Kestrel
- HttpClientFactory
- Migliorie per WebAPI
- Functional MVC testing
- GDPR e Identity
- Razor UI in Class Library
- SignalR
- Docker e Alpine

CommunityDays.it

Cos'è .NET Core

Piattaforma di sviluppo

Cross-platform

- Windows: x64, x86
- Linux: x64 and ARM32 (nuovo in 2.1, Raspberry)
- macOS: x64

Flexible deployment

Side-by-side o machine-wide

Open source su GitHub

Supportato da Microsoft

Eredita Base Class Library di .NET Framework 4.x

Rivisto: no dipendenze Windows

Compatibile .NET Framework, Xamarin e Mono

CommunityDays.it

ASP.NET Core 2 Recap

ASP.NET Core = MVC + WebAPI

Novità di 2.1: SignalR

Performance migliorate

- Partenza più rapida, maggior numero di richieste al secondo

Partenza più rapida

- Template semplificati

Benefici introdotti da .NET Core 2

- .NET Standard 2, etc

Refactoring di componenti chiave

- Runtime, hosting con IIS, autenticazione, logging

Novità di .NET Core 2.1

- .NET Core Global Tools
  - Tool installabili globalmente e lanciabili da riga di commando
- Socket e SocketsHttpHandler
  - Eliminazione di dipendenze a livello di piattaforma (libcurl per Linux e macOS, WinHTTP per Windows)
  - Comportamenti congruenti a livello di piattaforma e versioni
- Supporto for Span<T>/Memory<T> in Socket/NetworkStream
- Performance
  - https://blogs.msdn.microsoft.com/dotnet/2018/04/18/performance-improvements-in-net-core-2-1/

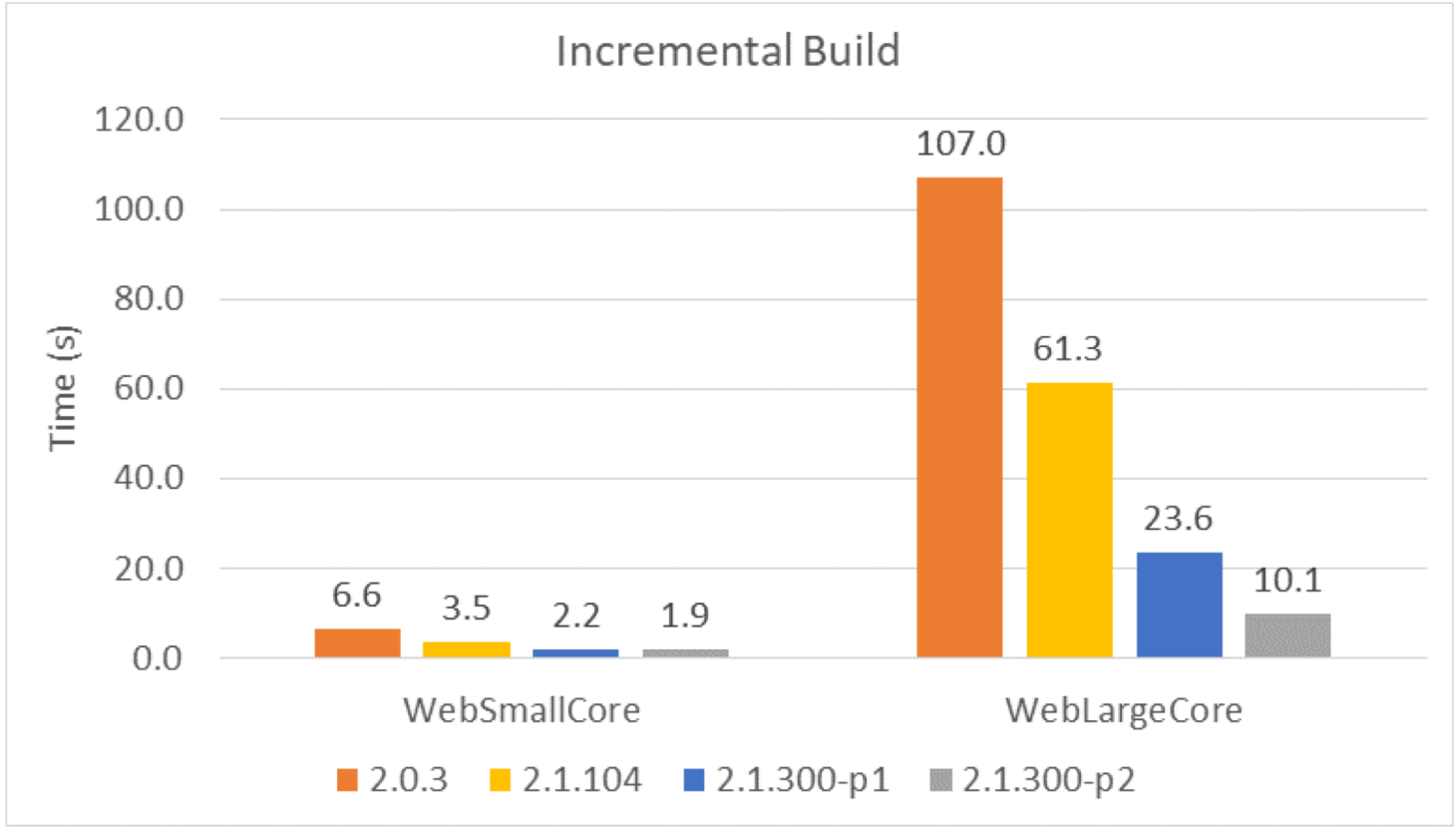
.NET Core 2.1: prestazioni in compilazione

Prestazioni migliorate grazie alla compliazione

Build incrementali

Il compilatore capisce quali parti sono effettivamente cambiate

- Speed up package asset resolution
- Speed up incremental package asset resolution
- MSBuild Node Reuse
- MSBuild ResolveAssemblyReferences cache



CommunityDays.it

Novità del runtime di ASP.NET Core 2.1

Su Windows, ASP.NET Core 2.1 gira in-process a wp3wp.exe di IIS

- più richieste al secondo

Su Linux

- comunicazione via Unix Socket per reverse proxy su NGINX
- rimossa la dipendenza di Kestrel da libuv (nuove Socket API)

Benchmark su https://github.com/aspnet/benchmarks

"Migrazione" a .NET Core 2.1

Nuovo meccanismo di **roll-forward on minor version**

Un’app compilata con una major version funziona con una minor uguale o successiva

.NET Core 2 -> .NET Core 2.1

Per migrarle esplicitamente bisogna modificare il .csproj

```
<TargetFramework>netcoreapp2.1</TargetFramework>
```

CommunityDays.it

MVC Compatibility Level

ASP.NET Core MVC 2.1 introduce diversi nuovi comportamenti

- Combinazione di Authorization Filter
- Nuovo exception handler per formatter
- Nuovo validatore per enum
- Migliore gestione del JSON formatter in caso di errori
- Aree per le Razor Pages

Questi nuovi comportamenti vanno abilitati

```
services.AddMvc()
        .SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
```

CommunityDays.it

Demo

File -> New Project

CommunityDays.it

HttpClientFactory

Sistema più semplice per gestire centralmente le istanze di HttpClient

non andrebbe mai creata/distrutta una nuova istanza ogni volta

ma neanche creato un singleton…

Possibilità di creare client personalizzati

Gestione smart del ciclo di vita di HttpClient

Possibilità di integrarlo con Polly

Libreria per gestire il Retry Pattern in maniera automatica

https://github.com/App-vNext/Polly/wiki/Polly-and-HttpClientFactory

HttpClientFactory

Esempio di Policy per Transient Error

```
services.AddHttpClient("GitHub", client =>
{
  client.BaseAddress =
          new Uri("https://api.github.com/");
  client.DefaultRequestHeaders
          .Add("Accept",
             "application/vnd.github.v3+json");
})
.AddTransientHttpErrorPolicy(builder =>
  builder.WaitAndRetryAsync(new[]
  {
    TimeSpan.FromSeconds(1),
    TimeSpan.FromSeconds(5),
    TimeSpan.FromSeconds(10)
  }));
```

```
public class MyController : Controller
{
  private readonly IHttpClientFactory
_httpClientFactory;

  public MyController(IHttpClientFactory
httpClientFactory)
  {
    _httpClientFactory = httpClientFactory;
  }

  public Task<IActionResult> SomeAction()
  {
    var client =
_httpClientFactory.CreateClient("GitHub");

    return Ok(
     await client.GetStringAsync("/someapi"));
  }
}
```

CommunityDays.it

MVC functional test

- Simulazione di test funzionali
- Supporto per più richieste
- Simulazione navigazione utente

```
public class TestingMvcFunctionalTests :
IClassFixture<WebApplicationFactory<Startup>>
{
  public TestingMvcFunctionalTests(WebApplicationFactory<Startup>
fixture)
  {
    Client = fixture.CreateClient();
  }

  public HttpClient Client { get; }

  [Fact]
  public async Task GetHomePage()
  {
    // Inviamo una richiesta all'homepage
    var response = await Client.GetAsync("/");

    // Ci aspettiamo che lo status code restituito sia 200 (OK)
    // L'oggetto response ci permette ovviamente di verificare che
    // il corpo della risposta sia conforme alle aspettative
    Assert.Equal(HttpStatusCode.OK, response.StatusCode);
  }
}
```

CommunityDays.it

Web API conventions

Nuovo attributo APIController

- ActionResult<T> per aiutare Swagger
- ModelState.IsValid chiamato in automatico, se false risposta 400 Bad Request

```
[ApiController, Route("api/[controller]")]
public class ProductsController : Controller {
  [HttpPost, Route("")]
  public ActionResult<Product> Get(int id, [FromBody] Product request)
  {
    var result = repo.GetProduct(id);
    //…
    return Ok(result);
  }
}
```

CommunityDays.it

Sicurezza e GDPR

Supporto a HSTS

Gestione sicura dello switching a HTTPS

```
services.AddHsts(options => {
    options.IncludeSubDomains = true;
    options.Preload = true;
    options.MaxAge = TimeSpan.FromDays(365);
});
```

Cookie consent

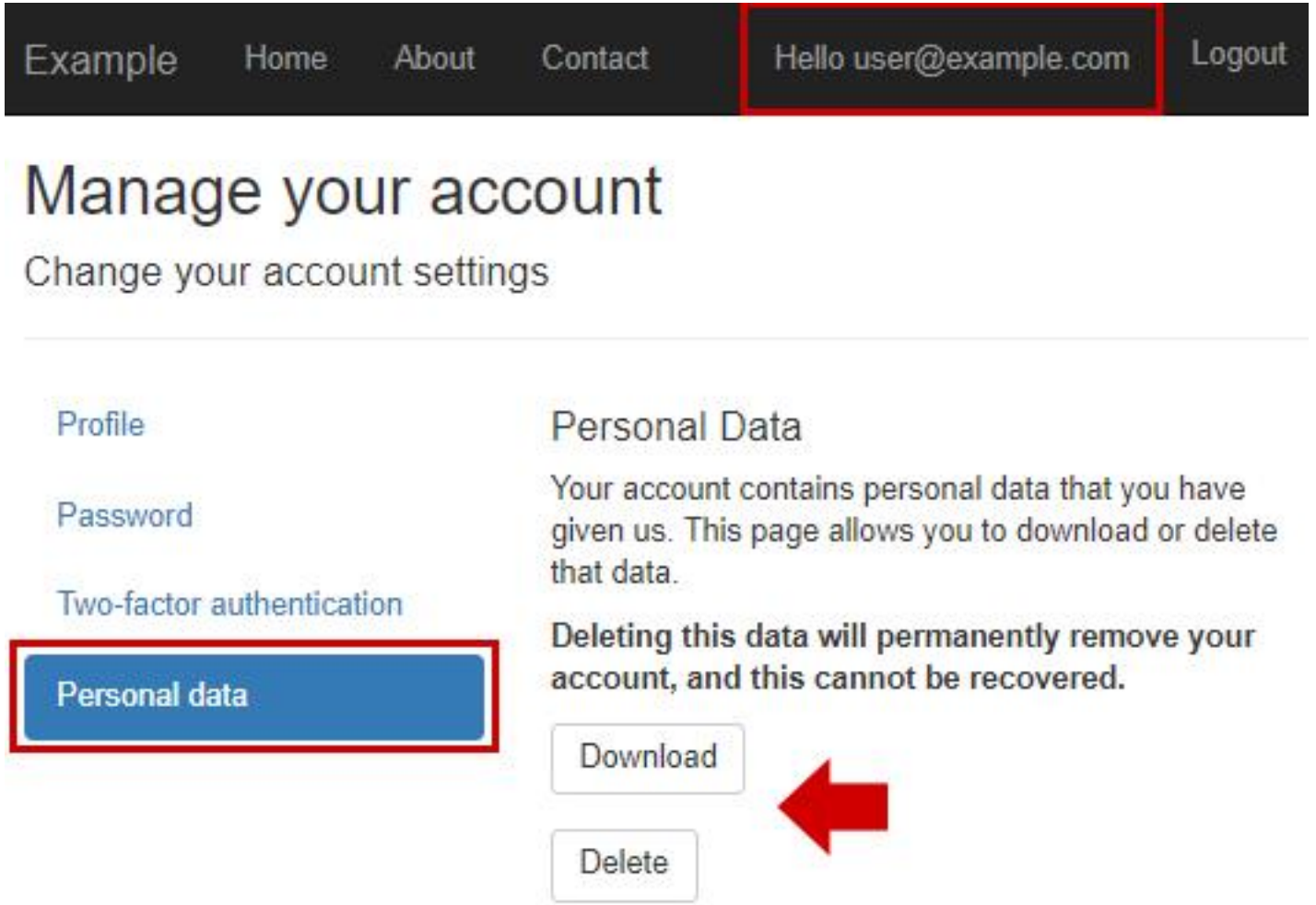
```
app.UseCookiePolicy();
```

Blocca tutti i cookie non essenziali (nuova proprietà IsEssential su HttpCookie) (non quelli di auth) fino a consenso

CommunityDays.it

Nuovi template di Identity GDRP-Ready

- Scaricamento dei dati personali in formato JSON
- Cancellazione dei dati
- Dati in possesso dell'applicazione



CommunityDays.it

Identity UI as a lib

Grazie ad una nuova funzionalità che consente di tenere le View Razor dentro una class lib e compilarle

Identity UI è distribuito con le sue view compilato su NuGet

In caso di aggiornamento vengono aggiornate anche le parti di UI

Resta possibile aggiungerle esplicitamente con lo scaffolder

CommunityDays.it

Demo

Razor UI Lib & Identity UI

ASP.NET Core SignalR

Framework per real-time app portato da ASP.NET

Riscritto da zero

Usa WebSockets, Server-sent Events, Long Polling

Integrato con il runtime (Authentication, Authorization, etc)

Transport in JSON (testo) o MessagePack (binario)

Rimossa la dipendenza da jQuery

Nuova API lato client più semplice

Nuova API lato server tutta async

Attivato come middleware

CommunityDays.it

Demo

SignalR

CommunityDays.it

Docker e ASP.NET Core 2.1

Nuove immagini basate su Alpine (distro Linux ridotta all'osso)

- 2.1-runtime-alpine
- 2.1-runtime-deps-alpine*

Solo 83 MB anziché 219 come baseline

* Solo depedencies, senza runtime, per self-contained deployment

CommunityDays.it

Demo

Alpine con Docker

Grazie!

- Il materiale sarà online nei prossimi giorni su http://www.communitydays.it

CommunityDays.it