



#CodeGen

- #dotnetconf

@cloudgen_verona



Introduzione a ServiceStack

Un framework alternativo per la costruzione di Web API.

24/co

Community sponsors



Local Event



It's
FREE!

.NET Conf

Discover the world of .NET

September 12-14, 2018



Tune in: www.dotnetconf.net



Andrea Perazzolo



@hanpera1970



hanpera



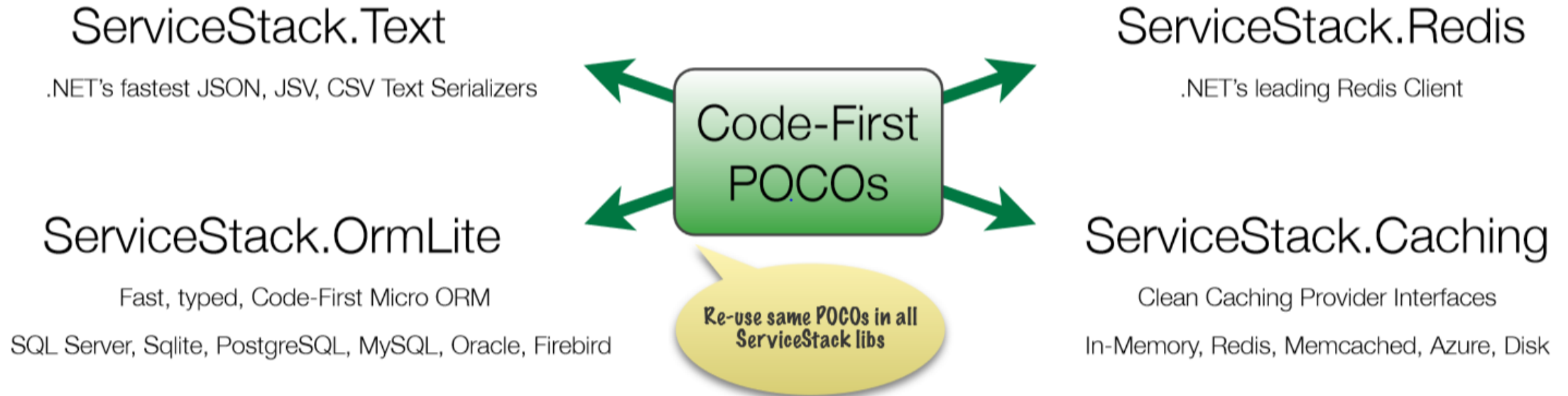
hanpera

- Servicestack è un framework open source creato per essere un'alternativa a WCF, ASP Net MVC e Web Api. Supporta endpoint SOAP e REST, configurazione automatica del formato dati, container iOc, mapping ORM, meccanismi di caching e molto altro.
- Prodotto commerciale con licenza (free fino a 10 endpoint)
- Supporta .Net Framework e .Net Core
- Supporta l'hosting su ASP.NET, Self Hosting, Windows Service, Xamarin.Mac

Caratteristiche



- **Simplicity**
- **Speed**
- **Web Services Best Practices**
- **Message-based Services** - Model-driven, code-first, friction-free development
- **Native Clients** - Clean, end-to-end typed idiomatic APIs for most major platforms
- **Modern** - No XML config, IOC built-in, no code-gen, conventional defaults
- **Smart** - Infers greater intelligence from your strongly typed DTOs
- **Effortless Features** - Most features enhance your existing DTOs making them trivial to enable
- **Multi Platform** - Supports .NET 4.5 and .NET Core platforms for hosting on Windows, OSX, Linux
- **Multiple Hosts** - Run in Web, Console, native Windows/OSX Desktop Apps, Windows Services
- **Host Agnostic** - Services are decoupled from HTTP and can be hosted in MQ Services
- **Highly testable** - Typed, idiomatic client APIs enable succinct, intuitive Integration tests
- **Mature** - Stable with over 8+ years of development
- **Preserve Investment** - modern libraries that are [Continuously Improved](#) (not abandoned or replaced)
- **Dependable** - Commercially supported and actively developed



More: node.js-powered Bundler, Logging, Auto-Mappers, Service Clients, Markdown Razor, Configuration Providers, Enhanced Funq IOC, ...

Struttura



Built on raw ASP.NET IHttpHandler's

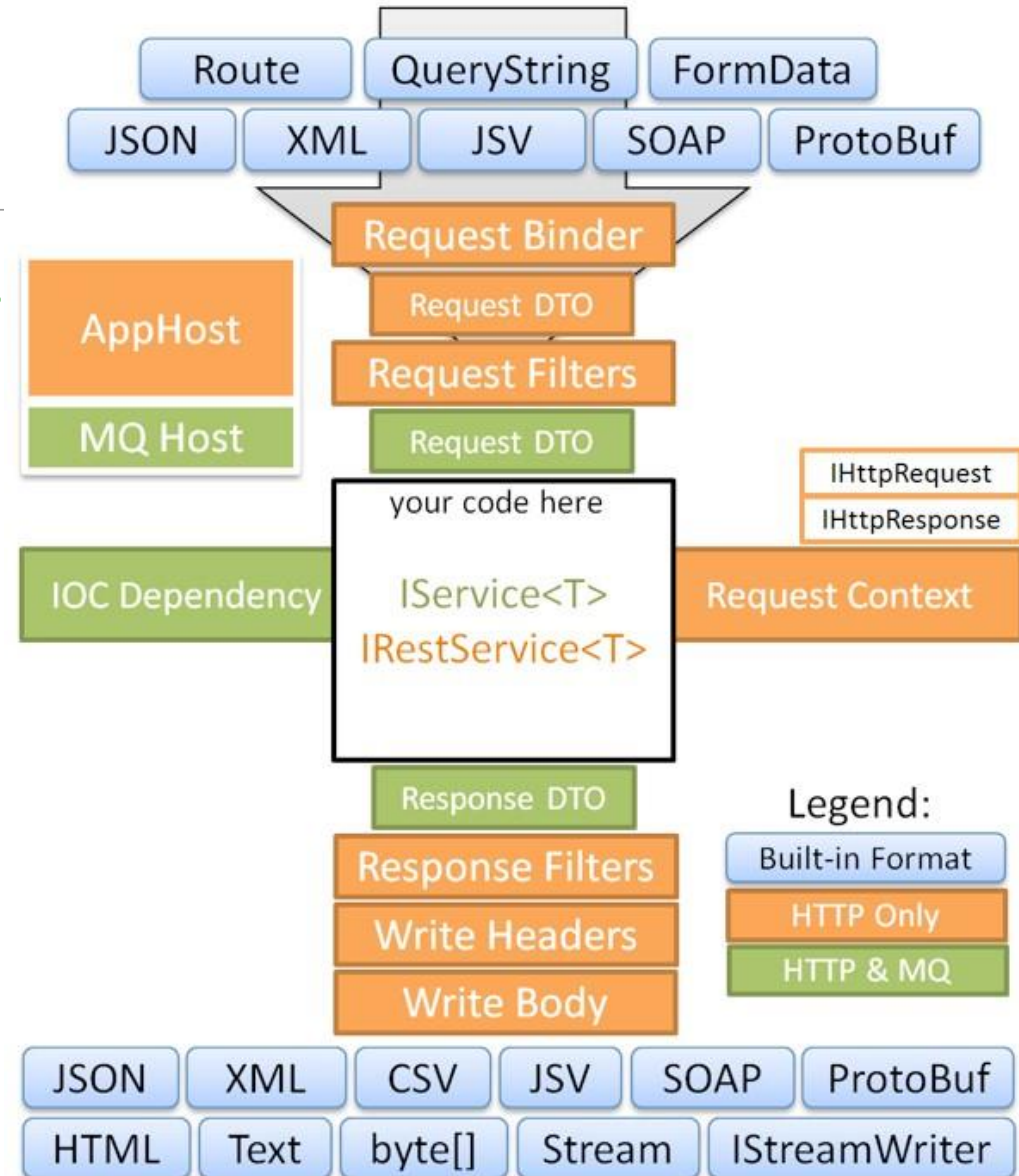
Stand-alone HttpListener, MQ, RCON hosts

Runs Everywhere



- XSP
- mod_mono
- FastCGI
- Console

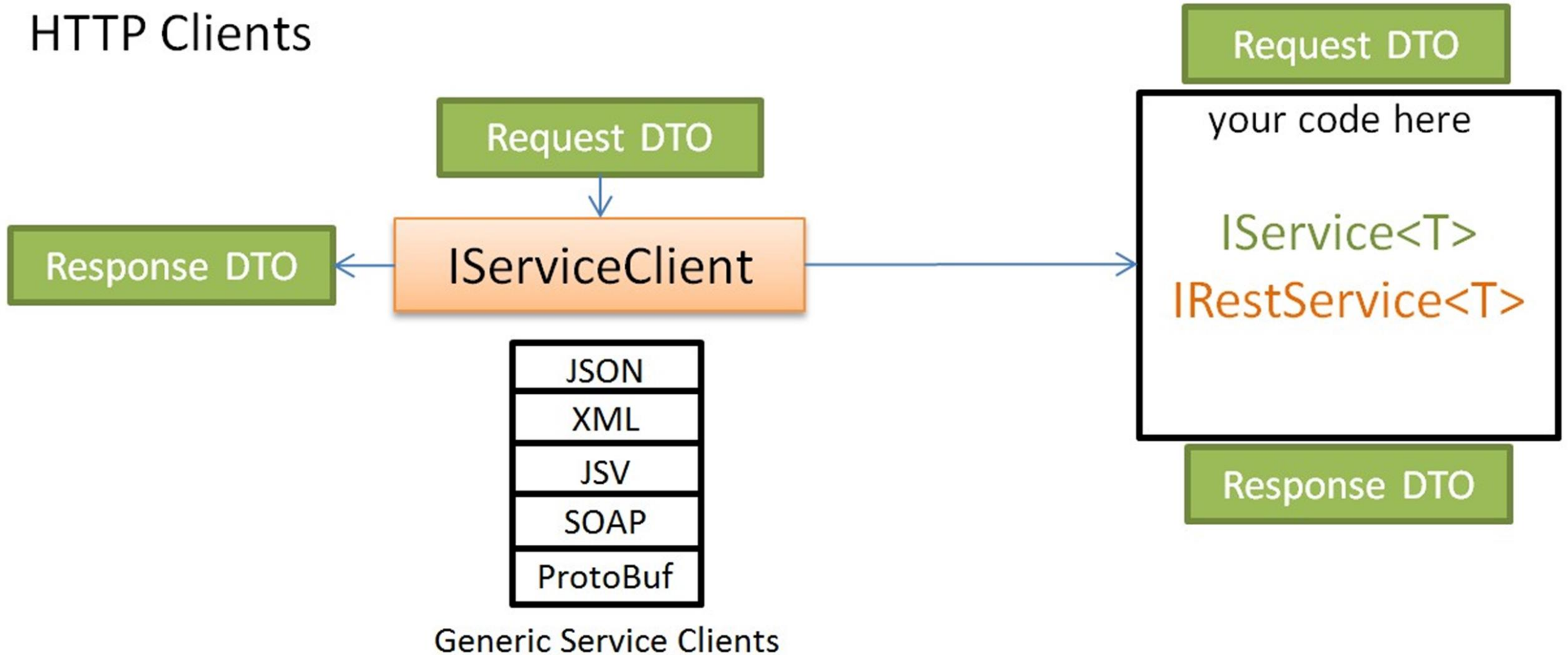
Clean Slate using Code-First POCO's
New Auth, Caching, Session Providers



Http Clients



HTTP Clients



Your First ASP.NET Web API (C#) Tutorial:

```
public class ProductsControllerV2 : ApiController
{
    public IEnumerable<Product> GetAllProducts()
    {
        return products;
    }

    public Product GetProductById(int id)
    {
        var product = products.FirstOrDefault((p) => p.Id == id);
        if (product == null)
        {
            throw new HttpResponseException(HttpStatusCode.NotFound);
        }
        return product;
    }

    public Product GetProductByName(string categoryName)
    {
        var product = products.FirstOrDefault((p) => p.Name == categoryName);
        if (product == null)
        {
            throw new HttpResponseException(HttpStatusCode.NotFound);
        }
        return product;
    }

    //Adding existing services with more RPC method calls
    public IEnumerable<Product> GetProductsByCategory(string category)
    {
        return products.Where(
            (p) => string.Equals(p.Category, category,
                StringComparison.OrdinalIgnoreCase));
    }

    public IEnumerable<Product> GetProductsByPriceGreaterThan(decimal price)
    {
        return products.Where((p) => p.Price > price);
    }
}
```

5 vs 2
Services &
Entry Points

The Same Services in ServiceStack:

```
public class FindProducts : IReturn<List<Product>>
{
    public string Category { get; set; }
    public decimal? PriceGreaterThan { get; set; }
}

public class GetProduct : IReturn<Product>
{
    public int? Id { get; set; }
    public string Name { get; set; }
}

public class ProductsService : IService
{
    public object Get(FindProducts request)
    {
        var ret = products.AsQueryable();
        if (request.Category != null)
            ret = ret.Where(x => x.Category == request.Category);
        if (request.PriceGreaterThan.HasValue)
            ret = ret.Where(x => x.Price > request.PriceGreaterThan.Value);
        return ret;
    }

    public object Get(GetProduct request)
    {
        var product = request.Id.HasValue
            ? products.FirstOrDefault(x => x.Id == request.Id.Value)
            : products.FirstOrDefault(x => x.Name == request.Name);

        if (product == null)
            throw new HttpError(HttpStatusCode.NotFound, "Product does not exist");

        return product;
    }
}
```

SOA Design: Services differentiated by calling semantics and Return type:
e.g. Find* filters results, whilst Get* fetches unique records:

Confronto con le Web API

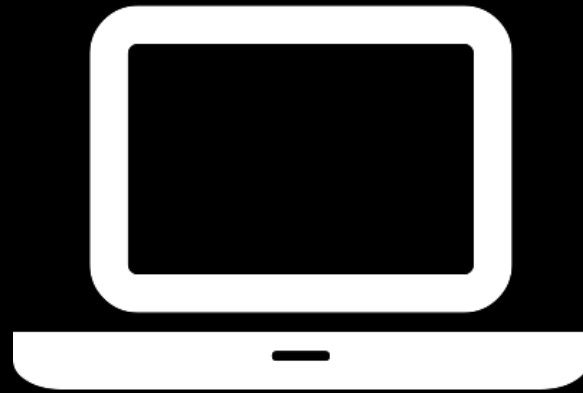


Differenza tra lo stile RPC delle Web API (o WCF)

```
public interface IWcfCustomerService
{
    Customer GetCustomerById(int id);
    List<Customer> GetCustomerByIds(int[] id);
    Customer GetCustomerByUserName(string userName);
    List<Customer> GetCustomerByUserNames(string[] userNames);
    Customer GetCustomerByEmail(string email);
    List<Customer> GetCustomerByEmails(string[] emails);
}
```

In contrasto con lo stile a messaggi:

```
public class Customers : IReturn<List<Customer>>
{
    public int[] Ids { get; set; }
    public string[] UserNames { get; set; }
    public string[] Emails { get; set; }
}
```



Creazione progetti



Features

Come aggiungere funzionalità al framework

Tutte le feature sono in forma di plugin che si possono aggiungere al progetto API rendendolo modulare.

Servicestack arriva con un buon numero di Feature già preconfezionate.

- AuthFeature
- OpenApi
- Postman
- Cors
- Proxy
-
- Custom IPlugin

L'autenticazione è un plugin configurabile nel quale si possono definire uno o più AuthProvider

Oltre ai provider forniti da Servicestack è possibile fare una customizzazione o scriverne di propri (molti sono scritti dalla Community)

Una rotta autenticata si identifica con l'attributo [Authenticate]

Le autorizzazioni si definiscono tramite attributi [RequireRole] e [RequirePermission] che confrontano il valore con Ruoli e Permission salvati nella UserAuth (oggetto sessione dell'utente)

Providers forniti



Auth Providers

From the overview we can see the built-in AuthProviders that are included:

- **Credentials** - For authenticating with username/password credentials by posting to the `/auth/credentials` service
- **API Keys** - Allowing users to authenticate with API Keys
- **JWT Tokens** - Allowing users to authenticate with JWT Tokens
- **Basic Auth** - Allowing users to authenticate with HTTP Basic Auth
- **Digest Auth** - Allowing users to authenticate with HTTP Digest Authentication
- **Custom Credentials** - By inheriting `CredentialsAuthProvider` and providing your own Username/Password `TryAuthenticate` implementation
- **AspNetWindowsAuthProvider** - Allowing users to authenticate with Windows Authentication

OAuth Providers

- **Twitter OAuth** - Allow users to Register and Authenticate with Twitter
- **Facebook OAuth** - Allow users to Register and Authenticate with Facebook
- **GitHub OAuth** - Allow users to Register and Authenticate with GitHub
- **Yammer OAuth** - Allow users to Register and Authenticate with Yammer
- **Yandex OAuth** - Allow users to Register and Authenticate with Yandex
- **VK OAuth** - Allow users to Register and Authenticate with VK
- **Odnoklassni OAuth** - Allow users to Register and Authenticate with Odnoklassni

OAuth2 Providers

The [ServiceStack.Authentication.OAuth2](#) NuGet package provides OAuth2 Providers support to ServiceStack that includes:

- **Instagram OAuth2** - Allow users to Register and Authenticate with Instagram OAuth2
- **Google OAuth2** - Allow users to Register and Authenticate with Google OAuth2
- **LinkedIn OAuth2** - Allow users to Register and Authenticate with LinkedIn OAuth2
- **Microsoft Live OAuth2** - Allow users to Register and Authenticate with Microsoft Live OAuth2

Sono forniti i seguenti repository:

OrmLite: OrmLiteAuthRepository in ServiceStack.Server

OrmLiteAuthRepositoryMultitenancy

Redis: RedisAuthRepository in ServiceStack

In Memory: InMemoryAuthRepository in ServiceStack

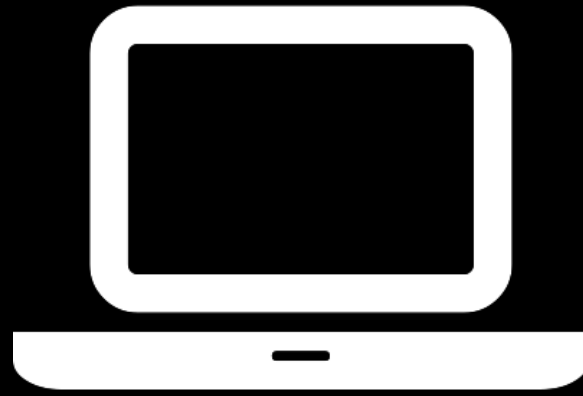
AWS DynamoDB: DynamoDbAuthRepository in ServiceStack.Aws

Mongo DB: MongoDBAuthRepository in ServiceStack.Authentication.MongoDB

Raven DB: RavenUserAuthRepository in ServiceStack.Authentication.RavenDB

Marten: MartenAuthRepository in ServiceStack.Authentication.Marten - GitHub project

LiteDB: LiteDBAuthRepository in ServiceStack.Authentication.LiteDB



Inserimento Feature

Non solo API



Dalla versione 5 vengono forniti template per i principali framework Web (Vue, React, Angular ecc.)

Permette di creare pagine con una sua sintassi Razor (sotto Asp Net Core è stato sostituito dal Razor nativo).

- Creazione di un Host di API singolo
- Applicazioni a MicroServizi dove un host può essere client di un altro tramite la condivisione dei DTO.
- Host di Auth e Single SignOn

<https://docs.servicestack.net/> per la documentazione

StackOverflow per help, topics

<https://github.com/ServiceStack/ServiceStack> per il codice sorgente

Presentazione: <https://www.slideshare.net/newmovie/what-istheservicestack-14819151?ref=https://servicestack.net/features>

Ebook Syncfusion Succintly

<https://www.syncfusion.com/ebooks/servicestack>

Grazie

Domande?



hanpera



@hanpera1970



hanpera