



DelphiDay
italian conference

Server REST con WiRL, Neon e InstantObjects

Integriamo 3 librerie Delphi open-source per sviluppare
in poco tempo un server REST funzionante



Carlo Barazzetta

Ethea S.r.l. - Socio fondatore e
Responsabile progetti IT

carlo.barazzetta@ethea.it

x.com/CarloBarazzetta

github.com/EtheaDev

linkedin.com/in/carlo-barazzetta



19-20 Giugno 2025
Piacenza



OPEN-SOURCE PROJECTS **COMPONENTS**

github.com/EtheaDev

SVG Icon ImageList ^{355*}

github.com/EtheaDev/SVGIconImageList

Icon Fonts ImageList ^{222*}

github.com/EtheaDev/IconFontsImageList

Styled Components ^{174*}

github.com/EtheaDev/StyledComponents

OPEN-SOURCE PROJECTS **COMPONENTS**

github.com/EtheaDev

Delphi GoogleMap ^{82*}

github.com/EtheaDev/DelphiGoogleMap

Markdown Help Viewer ^{74*}

github.com/EtheaDev/MarkdownHelpViewer

DBAware Labeled Components ^{30*}

github.com/EtheaDev/DBAwareLabeledComponents

OPEN-SOURCE PROJECTS

SHELL EXTENSIONS

github.com/EtheaDev

SVG Shell Extensions ¹⁶³

github.com/EtheaDev/SVGShellExtensions

SKIA Shell Extensions ^{72*}

github.com/EtheaDev/SKIAShellExtensions

Markdown Shell Extensions ^{84*}

github.com/EtheaDev/MarkdownShellExtensions

Fattura Elettronica Explorer ^{24*}

github.com/EtheaDev/FExplorer

OPEN-SOURCE PROJECTS

OTHERS

github.com/EtheaDev

InstantObjects ^{103*}

github.com/EtheaDev/InstantObjects

VCL Theme Selector ^{59*}

github.com/EtheaDev/VCLThemeSelector

Delphi MVC

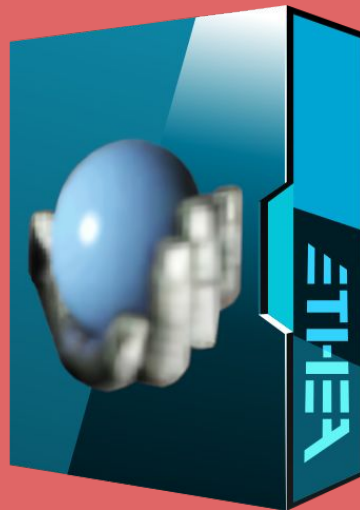
[github.com/carloBarazzetta/Delphi MVC](https://github.com/carloBarazzetta/Delphi_MVC)

AGENDA

1. **InstantObjects: ORM+OPF**
 - a. La struttura delle classi / attributi
 - b. Peculiarità ClassName / Id - InstantReference - InstantParts
2. **Neon: Serializzatore/Deserializzatore**
 - a. customizzazione per InstantObjects
3. **WiRL: Server REST con InstantObjects**
 - a. Context per connessione InstantObject
 - b. Risorse di base per InstantObjects
 - c. Gestione utenti - ruoli e permessi



InstantObjects ORM+OPF



1

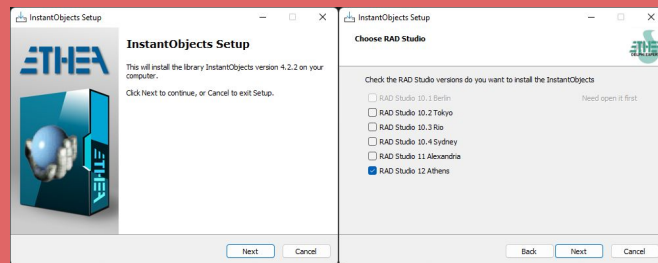
3 Librerie OpenSource

→ Documentazione e Repo Git di WiRL/Neon:

- ◆ <https://wirl.delphiblocks.dev/>
- ◆ <https://github.com/delphi-blocks/WiRL>
- ◆ <https://github.com/paolo-rossi/delphi-neon>

→ Documentazione Repo Git e Installer di InstantObjects:

- ◆ <https://ethea.it/docs/instantobjects/>
- ◆ <https://github.com/EtheaDev/InstantObjects>
- ◆ https://github.com/EtheaDev/InstantObjects/releases/latest/download/InstantObjects_Setup.exe



InstantObjects: ORM / OPF

- Prima libreria ORM / OPF per Delphi
- Forte integrazione con l'IDE
- Gestione del Model via GUI e Codice (bidirezionale)
- InstantSelector e InstantExposer
 - ◆ InstantAttributes
 - ◆ InstantReference / InstantPart
 - ◆ InstantReferences / InstantParts
 - Chiave: ClassName e Id



InstantObjects: Model e Demo

→ Struttura logica del Model di esempio

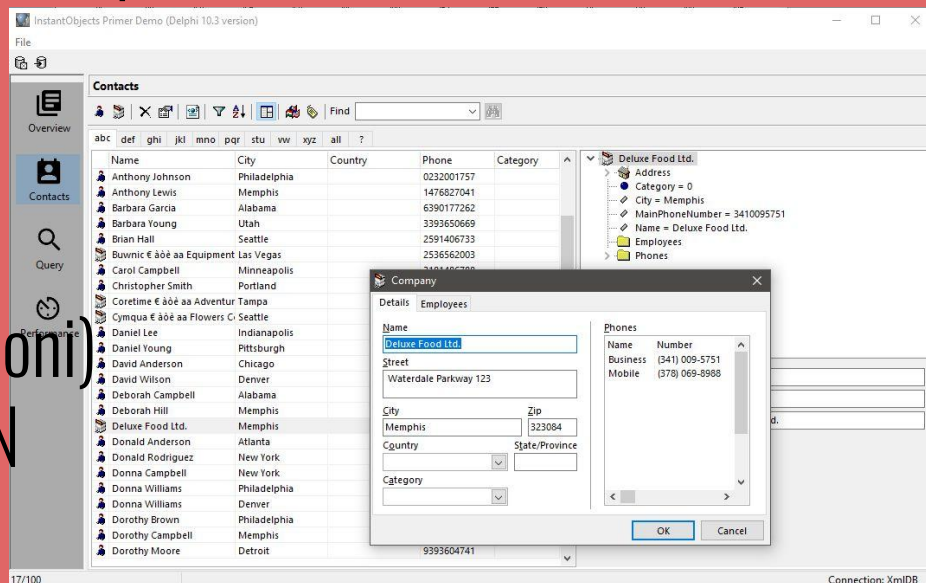
◆ TContact

- TPerson
- TCompany

→ Demo PrimerCross

◆ Diversi “Broker” (connessioni)

- Stream in XML o JSON

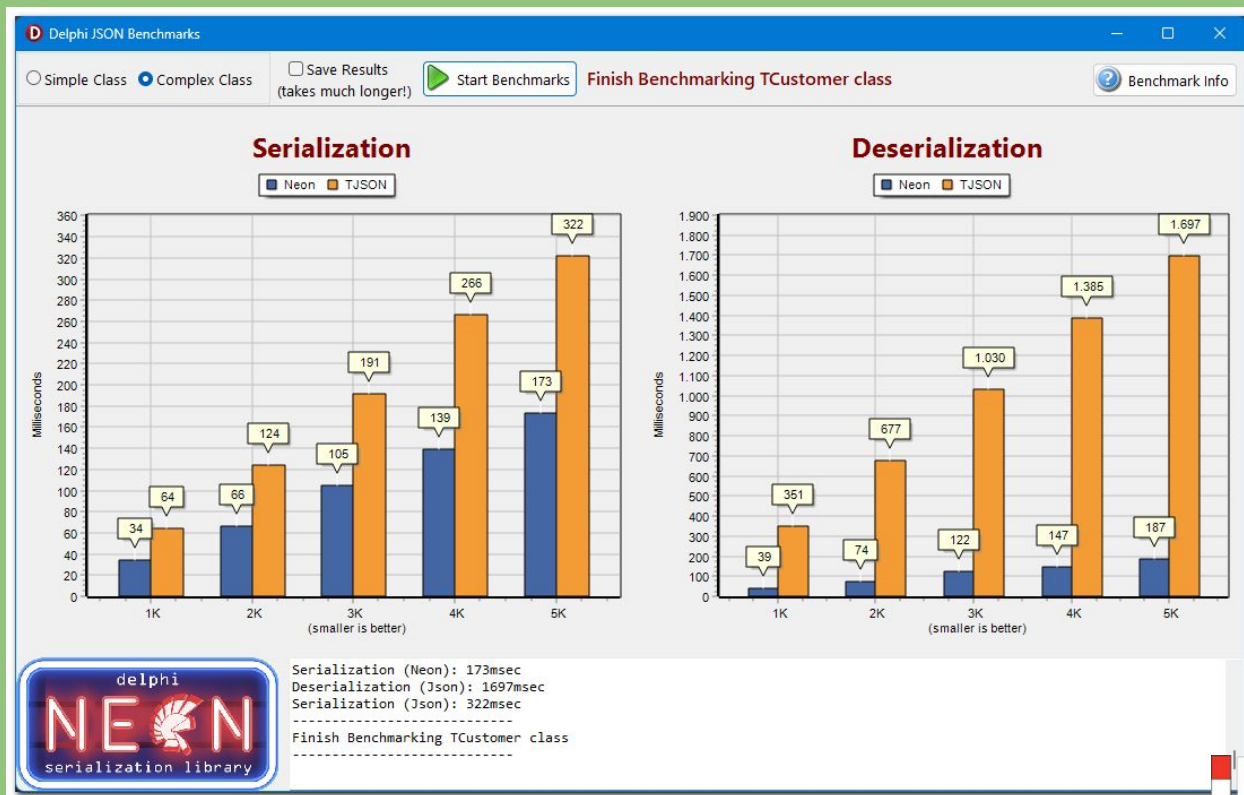




Delphi-Neon: serialization library

2

Delphi-NEON: performance!



Delphi-NEON e InstantObjects

- Neon: libreria per Serializzare / Deserializzare oggetti
- Possibilità di “customizzare” la serializzazione di Oggetto

◆ Instant.Neon.Serializers:

- TInstantReferenceSerializer
- TInstantReferencesSerializer
- TInstantPartsSerializer
- TInstantObjectListSerializer
- TInstantObjectReferenceListSerializer
- TInstantAttributeMetadataSerializer
- TInstantClassMetadataSerializer
- TInstantGraphicSerializer



WiRL: Delphi REST Library

3



Cenni su WiRL

WiRL è stato creato per semplificare l'implementazione dei servizi RESTful in Delphi ma, soprattutto, per consentire la massima interoperabilità con i client REST scritti in altri linguaggi e strumenti. WiRL si ispira alle specifiche Java JAX-RS e cerca di essere conforme ai 6 vincoli REST.

WiRL è un framework REST di alto livello che espone oggetti Delphi semplici come risorse web RESTful applicando attributi a queste classi.

```
[Path('customers')]
TCustomerResource = class
public
  [GET]
  [Produces('TMediaType.APPLICATION_JSON')]
  function SelectCustomers: TCustomerList;

  [POST]
  [Consumes('TMediaType.APPLICATION_JSON')]
  [Produces('TMediaType.APPLICATION_JSON')]
  function InsertCustomer(ACustomer: TCustomer):
    TCustomer;
end;
```

InstantObjects con WiRL

- Context InstantObject
 - ◆ InstantObjects.WiRL.Data (inject Context)
 - fornisce InstantFireDACConnector
- Classidi base:
 - ◆ InstantObjects.WiRL.Server.Resources.Base
 - Metodi POST, PUT, DELETE e Retrieve...
- Risorse generiche:
 - ◆ InstantObjects.WiRL.Server.Resources

Demo Primer WiRL: rest/primer/

→ POST: basic_auth, form_auth, body_auth

◆ TUser resource:

- GET rest/primer//TUser/{UserId}
- POST rest/primer//TUser/ChangePassword

→ Recupero delle chiavi:

- GET rest/primer/keys/{ClassName}

→ Recupero di dati:

- GET rest/primer/data/{ClassName}
- GET rest/primer/data/{ClassName}/{Id}

Demo Primer WiRL: rest/primer/

→ QUERY con IQL (Instant Query Language)

```
[GET, Path('/{AClassName}')]
, RolesAllowed('reader')
, Produces(TMediaType.APPLICATION_JSON)]
function RetrieveInstantObjectList([PathParam] AClassName: string;
[QueryParam] Where: string = '';
[QueryParam] OrderBy: string = '';
[QueryParam] ChangedFrom: string = '';
[QueryParam] ChangedTo: string = '';
[QueryParam] PageCount: integer = 0;
[QueryParam] RecordCount: integer = 0):
TInstantObjectList<TInstantObject>; override;
```

Problema: Model ricorsivi

→ Si risolve grazie a TInstantReference:

```
{  
  "ClassName": "TPerson",  
  "Id": "006AF519D9A91C42A701C2155427499B"  
  "Name": "Jason Thomas",  
  "City": "New York",  
  "MainPhoneNumber": "1323856504",  
  "Employer": {  
    "ClassName": "TCompany",  
    "Id": "35ACE94E62FEC34398D18DFAEC6E4E9F"  
  },  
}
```


Conclusioni

4

Vantaggi e Svantaggi

→ Vantaggi:

- ◆ 3 Librerie già pronte all'uso, ben documentate
 - Autori italiani, supporto immediato 👍
- ◆ Esempi già pronti all'uso
- ◆ Utile per vedere come si customizza Neon
- ◆ Utile come Demo “evoluto” di un server REST con WiRL

→ Svantaggi:

- ◆ Soluzione legata a InstantObjects... ma adattabile ad altri
- ◆ framework



THANK YOU