

Daniele Teti

CEO & CTO

bit Time Professionals [it] | CTO Delphi Studio [es]



Software Architect, books author,
teacher, consultant, ML addicted

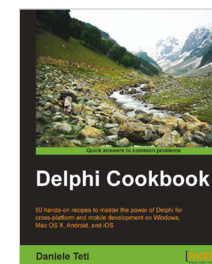
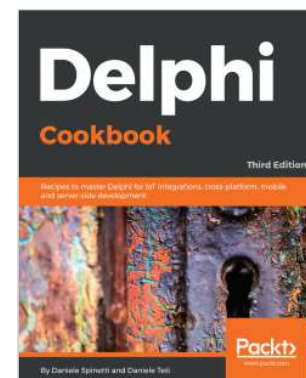
Open Source Projects

DMVCFramework, DelphiRedisClient,
ColumbusEgg4Delphi, LoggerPro, others

Blog www.danieleteti.it

Web www.bittimeprofessionals.it, www.delphistudio.es

Contacts d.teti@bittime.it, twitter: @danieleteti



DelphiMVCFramework

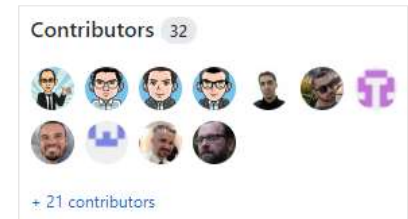
"REST (and a lot of other things) made simple as it should be"



What's DelphiMVCFramework?

- Designed to be a **powerful** and **easy** to use **Open Source** framework for web solutions;
- The **most popular** Delphi library on Github;
- Supports **REST** (RMM Level 3) and **JSON-RPC 2.0**;
- Support Server Sent Events for **push notifications**;
- Provides MVCActiveRecord (optional) to **access your data in a Object Oriented way**;
- **Standard** compliant, clients can use **any technology**;
- Used in some of the **biggest Delphi web systems in the world** (hundreds of thousands connections per second);

DelphiMVCFramework



- Completely **unit tested** (240+ unit tests)
- Works from Delphi 10 Seattle to Delphi 10.4 Sydney
- **10+ years** of development with **32 contributors** from all over the world;
- Greatly documented! Each feature has a sample (60+)!
- Free Facebook support group more than 3700 active members;
- Commercial support, training and custom development available on request (professionals@bittime.it)
- **Trusted by thousands of Delphi developers all around the world;**

DelphiMVCFramework as container of your business logic

- DMVCFramework is a server side solution;
- The real power is to not be tied to a particular client technology;
- It's possible to change the thin client without much effort maintaining all the business logic, and the real value of the system, on the server;

DelphiMVCFramework Deployments

- Console application (Windows, Linux, MacOS)
- GUI application (Windows VCL/FMX, Linux FMXLinux)
- Windows Service
- Linux daemon
- Apache Module (Windows, Linux)
- IIS ISAPI (Windows)

Most common development cycle

- Develop as
 - Console application
- Deploy small/mid systems as
 - Windows Service
 - Linux daemon
- Deploy big systems as
 - Apache Module (Linux, Windows)
 - IIS ISAPI (Windows)

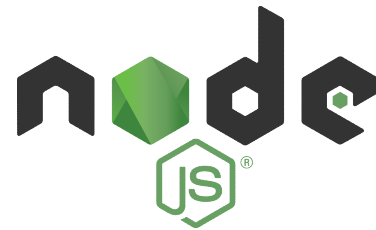
- Easier development and debug phase;
- Leverage all Delphi debugger power;
- Lower enviromental viscosity;
- The best from both worlds: easy development and full-flagged web server to host your module in production;



delphimvcframework

RESTful or JSON-RPC standard API

No Vendor Lock-In!
You choose your
clients technology

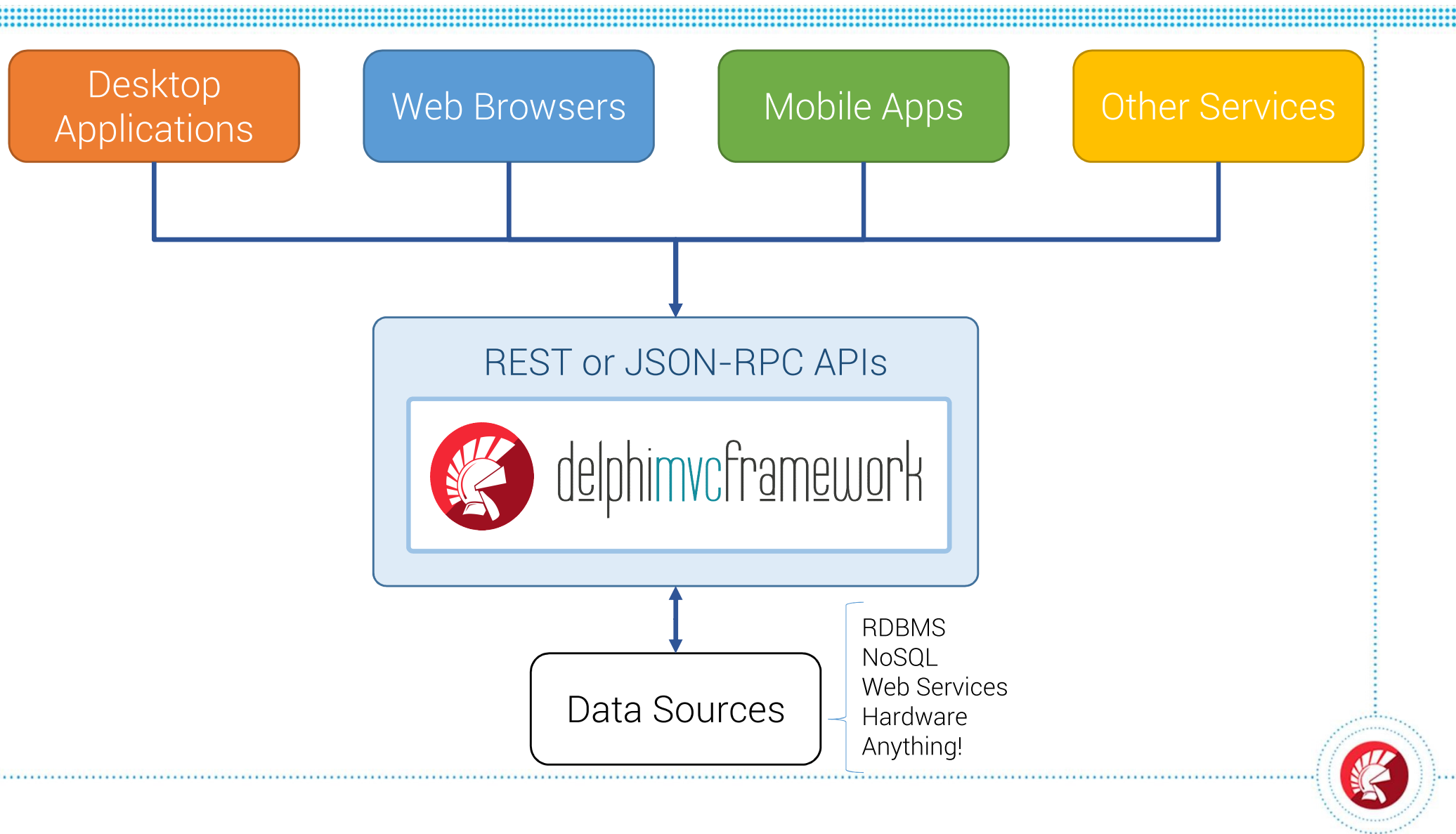


python



ELECTRON





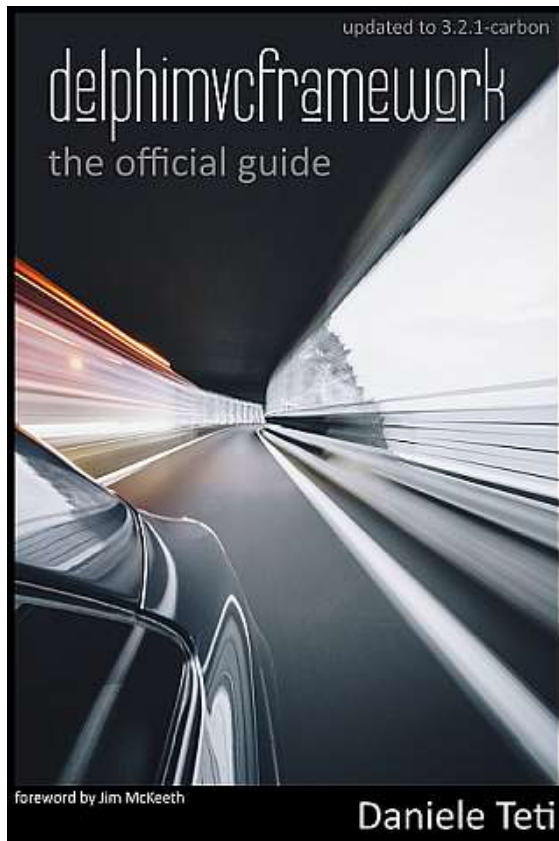
DelphiMVCFramework's Communities

Official Facebook support group more than 3700 members (english);
<https://www.facebook.com/groups/delphimvcframework>

Facebook support group for Brazilian developers (portuguese);
<https://www.facebook.com/groups/delphimvcframework-br>

Telegram Group in Spanish Language
<https://t.me/dmvcframework>

DelphiMVCFramework – The Official Guide



The perfect guide to leverage the power of DMVCFramework

The Guide is all what you need to fastly go beyond the basics

Available in English,
Portuguese and Spanish



Talk is cheap.
Show me the code.

Linus Torvalds

quote fancy

DEMOS

- Simple REST API using Datasets
- Simple REST API using MVCActiveRecord
- Simple JSON-RPC API
- Simple JSON-RPC API Client
- Python JSON-RPC Client

Selected Success Cases

- DMVCFramework is trusted and used by thousands of developers around the world, so there are a lot of success cases out there;
- Let's see just 2 of them...

Success Case #1: Remote Support

- Similar to TeamViewer™, DMVCFramework is used for users handling, licenses, dispatching
- DMVCFramework deployed as Apache cluster handles
- C.a. 10.000 "remote assistance" connections per hour, million of connections per month;
- Clients are from 100+ countries;
- Clients OS are MS Windows, macOS, Android and iOS;

Success Case #2: ERP

- Lot of API calls, huge databases with hundreds of tables, stored procedures and triggers;
- 500 instances in server farm, one for each customer
 - Many concurrent users for each customer;
- Clients are Web (Javascript) and Desktop (Delphi);

Links

Project Github repository

<https://github.com/danieleteti/delphimvcframework>

This session Github repository (slides + demos)

<https://github.com/danieleteti/delphicon2020>

DMVCFramework – The Official Guide

<https://leanpub.com/delphimvcframework> (English)

<https://leanpub.com/delphimvcframework-br> (Portuguese)

<https://leanpub.com/delphimvcframework-es> (Spanish)

Training, custom development and consultancy

professionals@bittime.it

Publish your data on the Web (1/3)

type

```
[MVCPath('/api/customers')]
TCustomersController = class(TMVCController)
public
    [MVCPath]
    [MVCHTTPMethods([httpGET])]
    procedure GetCustomers;

    [MVCPath('/($ID)')]
    [MVCHTTPMethods([httpGET])]
    procedure GetCustomerByID(const ID: Integer);
end;
```

Publish your data on the Web (2/3)

```
procedure TCustomersController.GetCustomers;
var
  lCustDM: TCustomersDM;
begin
  lCustDM := TCustomersDM.Create(nil);
  try
    lCustDM.MyQuery.Open('select * from customers order by code');
    Render(ObjectDict(False)
      .Add('data', lCustDM.MyQuery));
  finally
    lCustDM.Free;
  end;
end;
```

Publish your data on the Web (3/3)

```
procedure TCustomersController.GetCustomerByID(const ID: Integer);
var
    lCustDM: TCustomersDM;
begin
    lCustDM := TCustomersDM.Create(nil);
    try
        lCustDM.MyQuery.Open('select * from customers where id = ?', [ID]);
        if lCustDM.MyQuery.Eof then
            raise EMVCEException.Create(HTTP_STATUS.NotFound, 'Customer not found');
        Render(
            ObjectDict(False)
                .Add('data', lCustDM.MyQuery, nil, dstSingleRecord));
    finally
        lCustDM.Free;
    end;
end;
```

Accessing Data

- DMVCFramework is not opinionated about data access, choose what works for you, however you will love TMVCActiveRecord!
- TMVCActiveRecord is a powerful ActiveRecord Design Patten implementation to access your database data in an very OO way.
- TMVCActiveRecord supports for nullable types allows to easily map database data in business objects!
- Stop struggle with business logic mixed in datasets; we are not in '90!

TMVCActiveRecord – Declaring Entities

```
[MVCTable('articles')]
TArticle = class(TMVCActiveRecord)
private
  [MVCTableField('id', [foPrimaryKey, foAutoGenerated])]
  fID: NullableInt32;
  [MVCTableField('code')]
  fCode: NullableString;
  [MVCTableField('description')]
  fDescription : string;
  [MVCTableField('price')]
  fPrice: Currency;
public
  property ID: NullableInt32 read fID write fID;
  property Code: NullableString read fCode write fCode;
  property Description: string read fDescription write fDescription;
  property Price: Currency read fPrice write fPrice;
end;
```