Nuevos Recursos para JSON en Delphi y C++ Builder

Fernando Rizzato
Lead Software Consultant, *Latin America*

Embarcadero Technologies Copyright 2017

AGENDA

- JSON
  - ¿Qué es?
  - ¿Para que sirve?
- Frameworks
  - El clásico “JSON Objects Framework”
  - El nuevo “Readers/Writers JSON Framework”
- Componentes Personalizados JSON
- DataSet -> JSON / JSON -> DataSet
- Demos

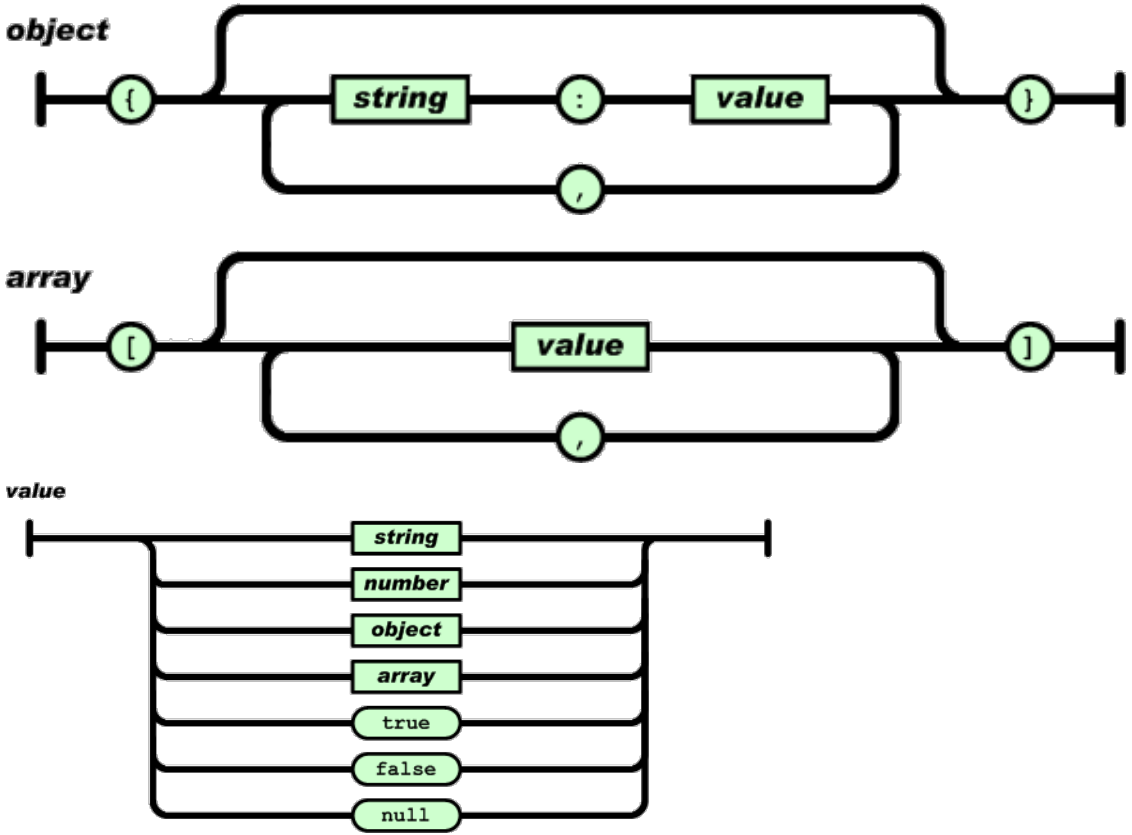
Embarcadero Technologies Copyright 2017

¿QUÉ ES JSON?

- **J**ava**S**cript **O**bject **N**otation
- Especificados originalmente a principios de 2000
- Formato de intercambio de datos ligero y independiente de la plataforma
- Se puede utilizar como una alternativa a otros formatos de intercambio de datos, tales como XML
- Es compatible con la mayoría de las lenguajes de programación modernas

Embarcadero Technologies Copyright 2017

¿QUÉ ES JSON?




```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

Embarcadero Technologies Copyright 2017

JSON OBJECTS FRAMEWORK

- Basado en el modelo **DOM** (Document Object Model), donde los objetos se mantienen en la memoria
- Clases principales: TJSONValue, TJSONObject, TJSONArray, TJSONNumber, TJSONString, TJSONTrue, TJSONFalse, TJSONNull

```
uses
  System.JSON;

procedure TFormJsonWrite.ButtonDOMClick(Sender: TObject);
var
  objStocks, objS1, objS2: TJSONObject;
  arrStocks: TJSONArray;

begin
  objStocks := TJSONObject.Create;
  try
    arrStocks := TJSONArray.Create;

    objS1 := TJSONObject.Create;
    objS1.AddPair('symbol', TJSONString.Create('ACME'));
    objS1.AddPair('price', TJSONNumber.Create(75.5));
```

Embarcadero Technologies Copyright 2017

READERS/WRITERS JSON FRAMEWORK

- Enfoque similar al estándar **XML SAX**, con un mejor rendimiento y el consumo de memoria reducido
- Clases principales: TJsonTextWriter, TJsonTextReader, TJSONObjectBuilder

```
uses
  System.JSON,
  System.JSON.Types,
  System.JSON.Writers;

procedure TFormJsonWrite.ButtonWriterClick(Sender: TObject);
var
  StringWriter: TStringWriter;
  Writer: TJsonTextWriter;
begin
  StringWriter := TStringWriter.Create();
  Writer := TJsonTextWriter.Create(StringWriter);
  try
    Writer.Formatting := TJsonFormatting.Indented;

    Writer.WriteStartObject;
    Writer.WritePropertyName('Stocks');
    Writer.WriteStartArray;
```

Embarcadero Technologies Copyright 2017

COMPONENTES PERSONALIZADOS JSON

- TJSONDocument
  - Encapsula la funcionalidad de análisis de JSON y proporciona acceso a JSON en tiempo de diseño. Este componente no es visual, por lo que se puede utilizar en proyectos FireMonkey y VCL
- TJSONTreeView
  - Componente visual para ver el JSON en forma de árbol. Este componente trabaja con el “TJSONDocument” y es un descendiente del componente "TTreeView" VCL. Por lo tanto, sólo se utilizará en proyectos VCL

- **Autor:** Pawel Glowacki (Embarcadero)

Embarcadero Technologies Copyright 2017

DATASET -> JSON / JSON -> DATASET

- FireDAC Reflection
  - FireDAC en ambos "lados”
  - Excelente rendimiento, fácil de usar, incluye clases para la extracción de delta, etc.
- Serialización de los Datasets en JSON estándar
  - Creando su propia clase
  - Utilizando uno de los muchos *frameworks* disponibles
  - Por lo general, se crean como "class helpers", haciendo su uso muy simplificado

Embarcadero Technologies Copyright 2017

DEMOS

Embarcadero Technologies Copyright 2017

RECURSOS ADICIONALES

- Documentación
  - http://www.json.org/
  - https://en.wikipedia.org/wiki/JSON
  - https://www.ietf.org/rfc/rfc4627.txt
  - http://docwiki.embarcadero.com/RADStudio/Berlin/en/JSON
  - http://docwiki.embarcadero.com/RADStudio/Berlin/en/JSON_Objects_Framework
  - http://docwiki.embarcadero.com/RADStudio/Berlin/en/Readers_and_Writers_JSON_Framework
  - http://docwiki.embarcadero.com/Libraries/Tokyo/en/Data.FireDACJSONReflect
  - http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Tutorial:_Using_a_REST_DataSnap_Server_with_an_Application_and_FireDAC

Embarcadero Technologies Copyright 2017

RECURSOS ADICIONALES

- Blogs y Vídeos
  - https://community.embarcadero.com/blogs/entry/how-to-convert-an-object-to-json-and-back-with-a-single-line-of-code-497
    https://community.embarcadero.com/blogs/entry/learn-how-to-use-the-new-json-features-in-rad-studio-10-seattle-webinar-march-2nd-wednesday
  - https://www.youtube.com/watch?v=onX1MoE3mUM

- Frameworks/Componentes
  - https://github.com/pglowack/DelphiJSONComponents
  - https://github.com/danieleteti/delphimvcframework (Mapper)
  - https://github.com/ezequieljuliano/DataSetConverter4Delphi (DataSet Converter 4 Delphi)

Embarcadero Technologies Copyright 2017

GRACIAS!

Preguntas?

Me puedes encontrar en:
@FernandoRizzato
fernando.rizzato@embarcadero.com

Síguenos en
fb.com/EMBTLatAm

Embarcadero Technologies Copyright 2017