



FireDAC TFDMemTable: DataSets em Memória

Fernando Rizzato
Lead Software Consultant, *Latin America*



AGENDA

- Criando e Preenchendo
- Persistência em Design e Runtime
 - .SaveToFile, .LoadFromFile
 - .SaveToStream, .LoadFromStream
- .CopyDataSet versus .Data
- .CloneCursor
- Otimizando a Performance
- Aplicações Comuns

CRIANDO E PREENCHENDO

```
with FDMemTable1.FieldDefs do begin
  with AddFieldDef do begin
    Name := 'f1';
    DataType := ftInteger;
  end;
  with AddFieldDef do begin
    Name := 'f2';
    DataType := ftString;
    Size := 50;
  end;
end;

with FDMemTable1 do begin
  Open;
  Append;
  Fields[0].AsInteger := ...;
  Fields[1].AsString := ...;
  Post;
end;
```

PERSISTÊNCIA EM DESIGN E RUNTIME

- SaveToFile, LoadFromFile, SaveToStream, LoadFromStream
- Formatos
 - sfAuto
 - Baseado na extensão do arquivo
 - sfBinary (uses FireDAC.Stan.StorageBin)
 - Mais eficiente
 - sfXML (uses FireDAC.Stan.StorageXML)
 - Menos eficiente
 - sfJSON (uses FireDAC.Stan.StorageJSON)
 - Amplamente aceito, especialmente mobile

.COPYDATASET VERSUS .DATA

- .CopyDataSet copia os registros do dataset de origem (source)
- .CopyDataSet funciona como uma sincronização de datasets
- .CopyDataSet funciona com qualquer dataset, enquanto .Data somente com datasets FireDAC
- .CopyDataSet dispara todos os eventos, enquanto .Data apenas atribui toda a massa de dados
- .CopyDataSet copia somente os valores atuais, enquanto .Data preserva versões e estado (inserted, deleted, updated)
- .Data é muito mais rápido que .CopyDataSet

.CLONECURSOR

- Compartilha os dados que pertencem a um outro dataset
- O armazenamento de dados interno é fisicamente o mesmo para ambos datasets
- AReset = False e AKeepSettings = False, propriedades de origem são copiadas para o destino
- AReset = True, propriedades retornam para o valor padrão no dataset de destino
- AReset = False e AKeepSettings = True, propriedades no destino não são alteradas

OTIMIZANDO A PERFORMANCE

```
FDMemTable1.LogChanges := False;  
FDMemTable1.FetchOptions.RecsMax := 300000; //Sample value  
FDMemTable1.ResourceOptions.SilentMode := True;  
FDMemTable1.UpdateOptions.LockMode := lmNone;  
FDMemTable1.UpdateOptions.LockPoint := lpDeferred;  
FDMemTable1.UpdateOptions.FetchGeneratorsPoint := gpImmediate;
```

```
FDMemTable1.BeginBatch;  
try  
  for i := 1 to 1000 do begin  
    FDMemTable1.Append;  
    // ...  
    FDMemTable1.Post;  
  end;  
finally  
  FDMemTable1.EndBatch;  
end;
```

APLICAÇÕES COMUNS

- FireDAC JSON Reflection
 - [Http://docwiki.embarcadero.com/RADStudio/en/Tutorial:_Using_a_REST_DataSnap_Server_with_an_Application_and_FireDAC](http://docwiki.embarcadero.com/RADStudio/en/Tutorial:_Using_a_REST_DataSnap_Server_with_an_Application_and_FireDAC)
 - <http://docwiki.embarcadero.com/Libraries/en/Data.FireDACJSONReflect>
- JSON DataSet Adapter
 - [http://docwiki.embarcadero.com/CodeExamples/en/REST.SurfSpotFinder_Sample_\(Delphi\)](http://docwiki.embarcadero.com/CodeExamples/en/REST.SurfSpotFinder_Sample_(Delphi))
 - <http://docwiki.embarcadero.com/Libraries/en/REST.Response.Adapter.TRESTResponseDataSetAdapter>

Mãos a Obra!

DEMOS

RECURSOS ADICIONAIS

■ Documentação

- <http://docwiki.embarcadero.com/Libraries/en/FireDAC.Comp.Client.TFDCustomMemTable>
- <http://docwiki.embarcadero.com/Libraries/en/FireDAC.Comp.Client.TFDMemTable>
- http://docwiki.embarcadero.com/RADStudio/en/TFDMemTable_Questions

■ Blogs

- CodeRage 9: **TFDMemTable** & CDS Compared
 - <http://youtu.be/iNgHJakYWkU>
 - Q&A: <http://wiert.me/2014/10/30/coderage-9-qa-log-clientdatasets-and-fdmemtables-compared/>
- CodeRage 9: **FireDAC** Tips, Tricks and News
 - <http://youtu.be/gljfudAKlTI>

OBRIGADO!

Perguntas?

Você pode me encontrar em:

@FernandoRizzato

fernando.rizzato@embarcadero.com

Siga-nos em

fb.com/DelphiBrasil

fb.com/EmbarcaderoBR