embarcadero

Embarcadero Conference

Um único esforço, uma única base de código, múltiplas plataformas, múltiplos dispositivos

embarcadero

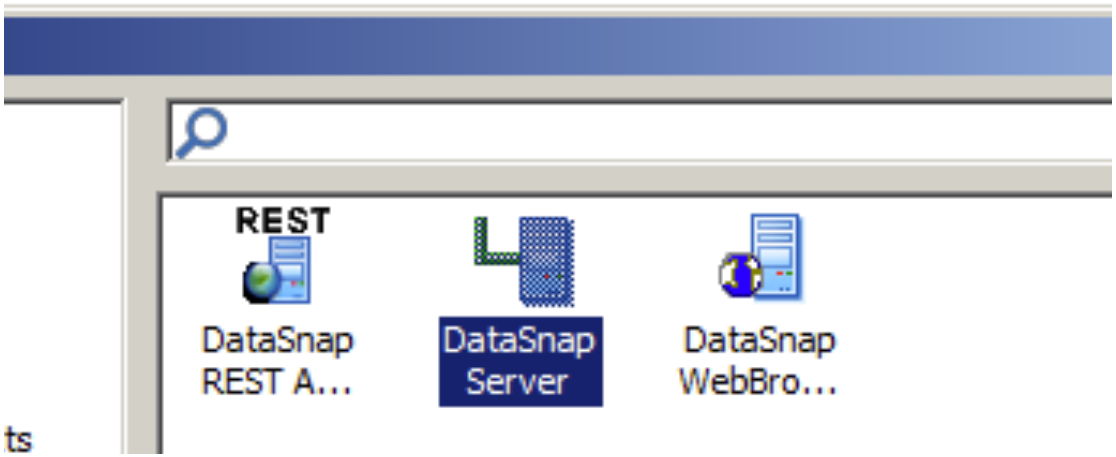
Adriano Santos

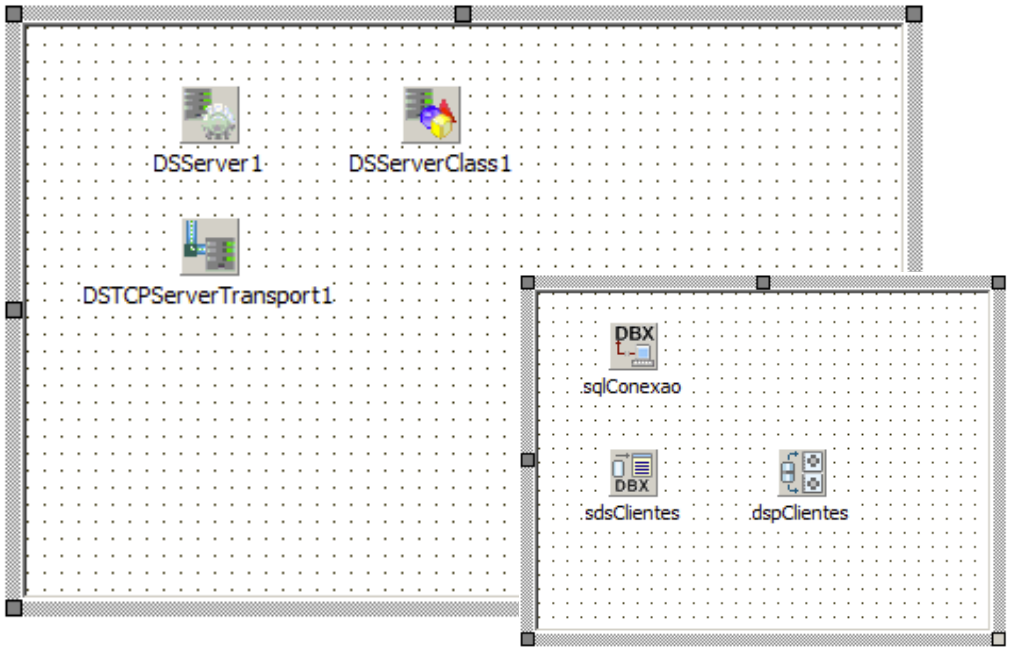
DataSnap + FireDAC + Apache

DataSnap Tradicional

- Protocolo TCP/IP;
- Uso de dbExpress;

- Componentes no Servidor
  - DSServer
  - DSServerClass
  - DSTCPServerTransport

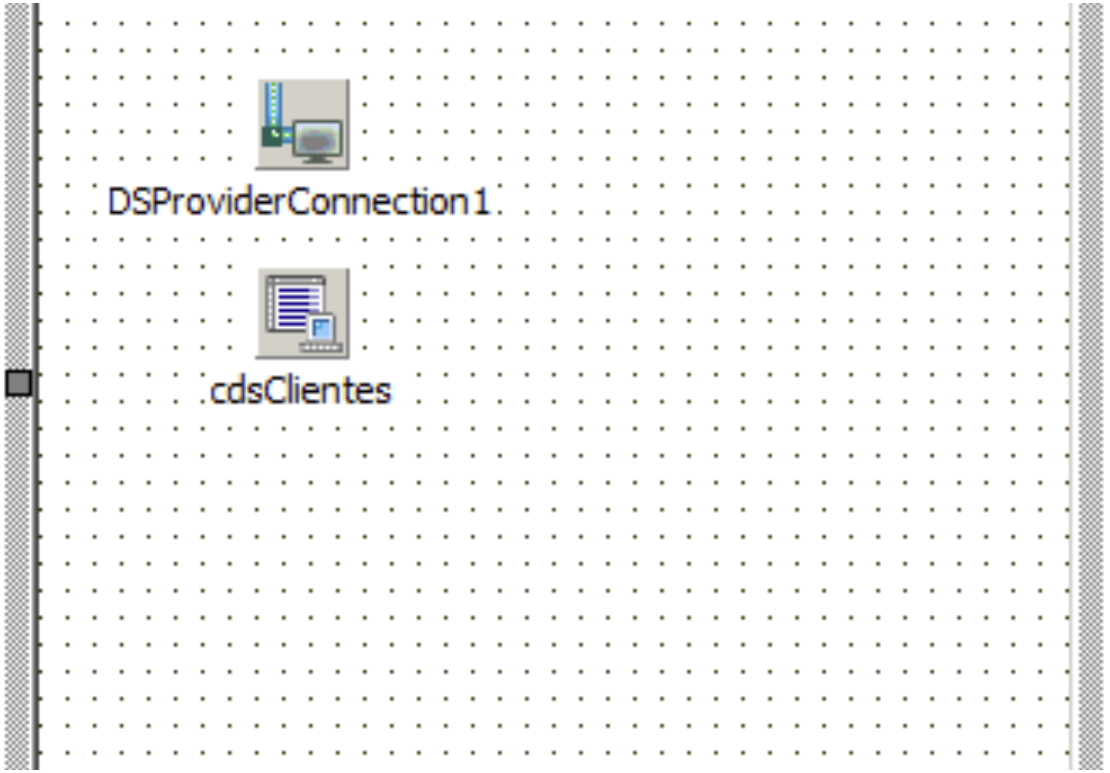




- Componentes no Servidor
  - SQLConnection
  - SQLDataSet ou SQLQuery
  - DataSetProvider

DataSnap Tradicional

- **Componentes no Cliente**
  - **DSProviderConnection**
  - **ClietDataset**



DataSnap com FireDAC

- Vantagens
  - Velocidade;
  - Escalabilidade;
  - Compatibilidade com outras tecnologias;
  - Acesso por HTTP e HTTPS;
  - Menor tráfego na rede;
  - Velocidade de download/upload dos dados;
  - Uso transparente em aplicações móveis;
  - Totalmente Stateless;
  - Menor número de requests ao servidor.

DataSnap com FireDAC

- Principais pontos no desenvolvimento
  - Classe: TFDJSONDataSets

```
/// <summary> List of FireDAC datasets that can be marshaled as JSON
/// </summary>
TFDJSONDataSetsBase = class
public type
  TItemPair = TPair<string, TFDAdaptedDataSet>;
  TItemList = class(TList<TItemPair>);
private
  [JSONReflect(ctTypeObject, rtTypeObject, TFDJSONInterceptor,nil,true)]
  [JSONOwned(false)] // Prevent unmarshal from freeing FList after construction
  FDataSets: TItemList;
  [JSONMarshalled(false)]
  FOwnsDataSets: Boolean;
public
  constructor Create;
  destructor Destroy; override;
end;
```

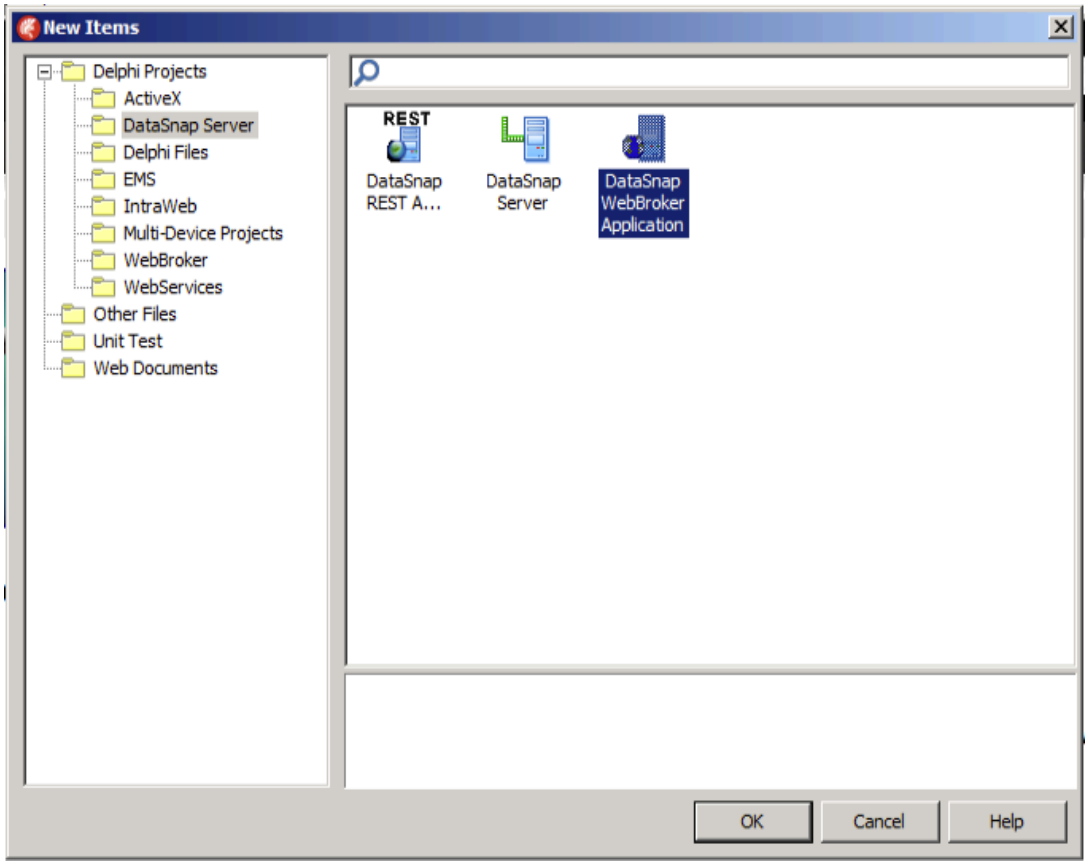
DataSnap com FireDAC

- Principais pontos no desenvolvimento
  - Exportação de objetos em JSON para o Cliente e vice-versa através da classe TFDJSONDataSets

```
public
  { Public declarations }
  function EchoString(Value: string): string;
  function ReverseString(Value: string): string;

  // Strongly typed methods
  function GetDepartmentNames: TFDJSONDataSets;
  function GetDepartmentEmployees(const AID: string): TFDJSONDataSets;
  procedure ApplyChangesDepartmentEmployees(const ADeltaList: TFDJSONDeltas);

  // Equivalent TJSONObject methods (C++ compatible)
  function GetDepartmentNamesJSON: TJSONObject;
  function GetDepartmentEmployeesJSON(const AID: string): TJSONObject;
  procedure ApplyChangesDepartmentEmployeesJSON(const AJSONObject: TJSONObject);

  function FileDownload(sFileName: string): TStream;
  procedure FileUpload(fStream: TStream);
```

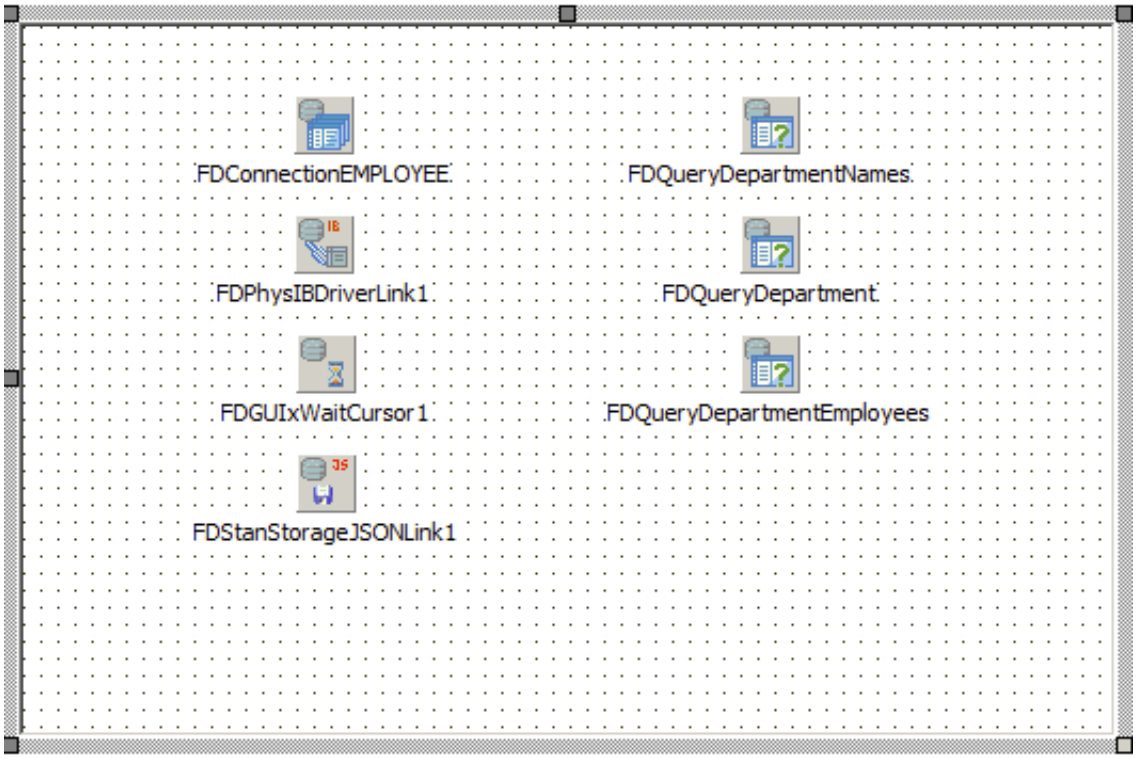
DataSnap com FireDAC

- Criação do Servidor com FireDAC
  - File > New > Other > DataSnap Server > DataSnap WebBroker Application



DataSnap com FireDAC

- Criação do Servidor com FireDAC
  - Acesso a dados utilizando FDConnection e FDQuery;



DataSnap com FireDAC

- A Classe Data.FireDACJSONReflect
  - Possui métodos completos para leitura e escrita de DataSets e também para transformação de objetos JSON;
  - Alguns métodos
    - ListAdd;
    - GetListValue;
    - GetListValueByName;

DataSnap com FireDAC

- Exportando um DataSet inteiro

```
function TServerMethods1.GetDepartmentNames:
  TFDJSONDataSets;
begin
  FDQueryDepartmentNames.Active := False;

  Result := TFDJSONDataSets.Create;
  TFDJSONDataSetsWriter.ListAdd(Result,
    FDQueryDepartmentNames);
end;
```

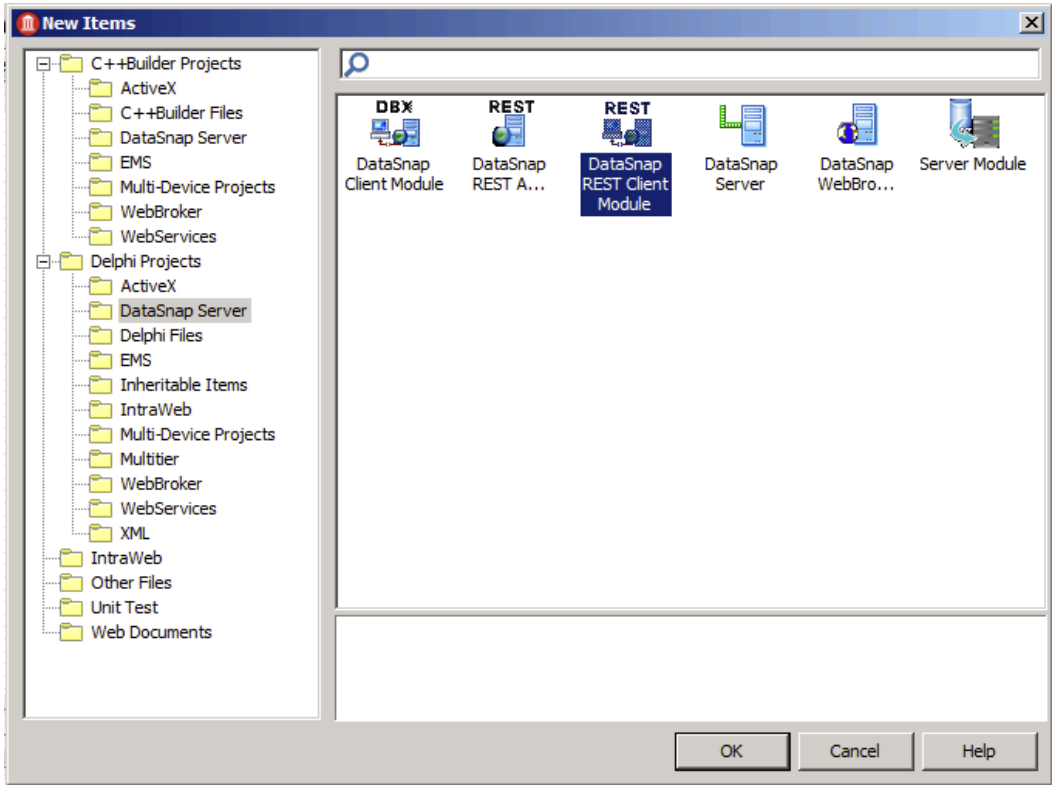
DataSnap com FireDAC

- Exportando vários DataSets no mesmo método GET

```
function TServerMethods1.GetDepartmentEmployees(const AID: string)
  : TFDJSONDataSets;
Begin
  FDQueryDepartmentEmployees.Active := False;
  FDQueryDepartment.Active          := False;
  FDQueryDepartment.Params[0].Value := AID;
  FDQueryDepartmentEmployees.Params[0].Value := AID;

  Result := TFDJSONDataSets.Create;

  TFDJSONDataSetsWriter.ListAdd(Result, sDepartment,
    FDQueryDepartment);
  TFDJSONDataSetsWriter.ListAdd(Result, sEmployees,
    FDQueryDepartmentEmployees);
end;
```

DataSnap com FireDAC

- Criando o cliente e lendo os ServerMethods
  - File > New > Other > DataSnap Server > DataSnap REST Client Module
  -



DataSnap com FireDAC

- Importando um DataSet do servidor

```
procedure TDepartmentsClientForm.GetDepartmentNames;
var
  LDataSetList: TFDJSONDataSets;
begin
  try
    LDataSetList :=
      ClientModule2.ServerMethods1Client.GetDepartmentNames();
    UpdateDepartmentNames(LDataSetList);
  except
    on E: TDSRestProtocolException do
      HandleRESTException(ClientModule2.DSRestConnection1,
        'Get Departments error', E)
    else
      raise;
  end;
end;
```

DataSnap com FireDAC

- Importando um DataSet do servidor

```
procedure TDepartmentsClientForm.UpdateDepartmentNames(const ADataSetList
  : TFDJSONDataSets);
begin
  FDMemTableDepartments.Active := False;

  Assert(TFDJSONDataSetsReader.GetListCount(ADataSetList) = 1);

  FDMemTableDepartments.AppendData(TFDJSONDataSetsReader.GetListValue
    (ADataSetList, 0));
end;
```

DataSnap com FireDAC

- Importando VÁRIOS DataSets do servidor

```
procedure TDepartmentsClientForm.UpdateDepartmentEmployees
  (ADataSetList: TFDJSONDataSets);
var
  LDataSet: TFDDataSet;
begin
  LDataSet := TFDJSONDataSetsReader.GetListValueByName(ADataSetList,
    sDepartment);
  FDMemTableDepartment.Active := False;
  FDMemTableDepartment.AppendData(LDataSet);

  LDataSet := TFDJSONDataSetsReader.GetListValueByName(ADataSetList,
    sEmployees);
  FDMemTableEmployee.Active := False;
  FDMemTableEmployee.AppendData(LDataSet);
end;
```

DataSnap com FireDAC

- Enviando alterações ao Servidor (ApplyUpdates)
  - Obter o(s) Delta(s) dos DataSet(s)
    - Classe: TFDJSONDeltas;
    - Ex.:

```
TFDJSONDeltasWriter.ListAdd(Result, sEmployees,
  FDMemTableEmployee);
TFDJSONDeltasWriter.ListAdd(Result, sDepartment,
  FDMemTableDepartment);
```

  - Enviar ao servidor

```
ClientModule2.ServerMethods1Client.ApplyChangesDepartmentEmployees
      (LDeltaList);
```

DataSnap com FireDAC

- Enviando alterações ao Servidor (ApplyUpdates)

```
procedure TDepartmentsClientForm.ApplyUpdates;
var
  LDeltaList: TFDJSONDeltas;
begin
  LDeltaList := GetDeltas;
  try
    ClientModule2.ServerMethods1Client.ApplyChangesDepartmentEmployees
      (LDeltaList);
  except
    on E: TDSRestProtocolException do
      HandleRESTException(ClientModule2.DSRestConnection1,
        'Apply Updates error', E)
    else
      raise;
  end;
end;
```

DataSnap com FireDAC

- Aplicando as alterações no lado Servidor

```
procedure TServerMethods1.ApplyChangesDepartmentEmployees(const ADeltaList
  : TFDJSONDeltas);
var
  LApply: IFDJSONDeltasApplyUpdates;
begin
  //Criação do objeto para receber o Apply
  LApply := TFDJSONDeltasApplyUpdates.Create(ADeltaList);
  //Aplicamos o Delta dos Departamentos
  LApply.ApplyUpdates(sDepartment, FDQueryDepartment.Command);

  if LApply.Errors.Count = 0 then
    //Se não ocorrerem erros, aplicamos o Delta do Employee
    LApply.ApplyUpdates(sEmployees, FDQueryDepartmentEmployees.Command);

  if LApply.Errors.Count > 0 then
    raise Exception.Create(LApply.Errors.Strings.Text);
end;
```

embarcadero

Apache

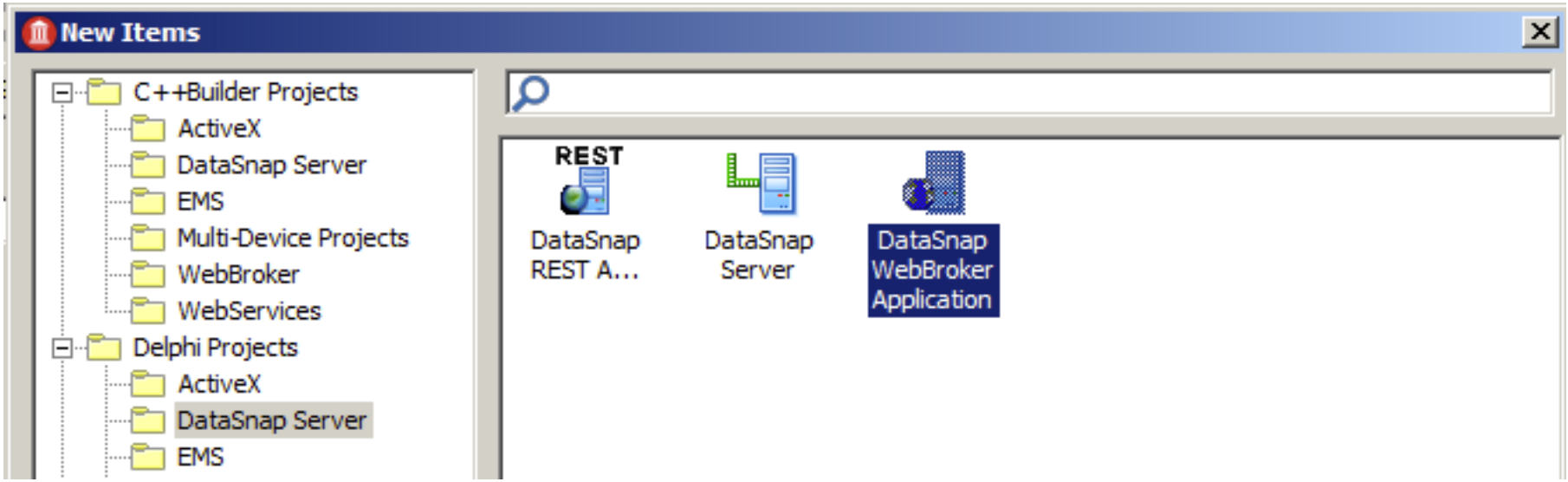
Distribuindo DataSnap Server

DataSnap com FireDAC versus Apache

- Como distribuir meu servidor de aplicações no Apache?
  - Versão do Apache 2.2 (Recomendável);

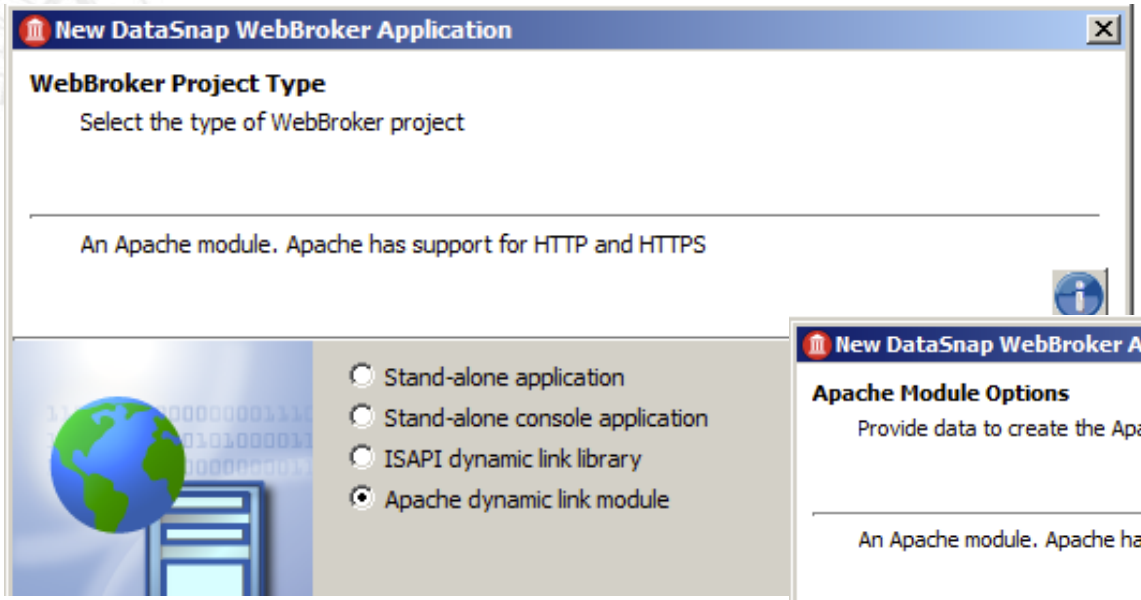
DataSnap com FireDAC versus Apache

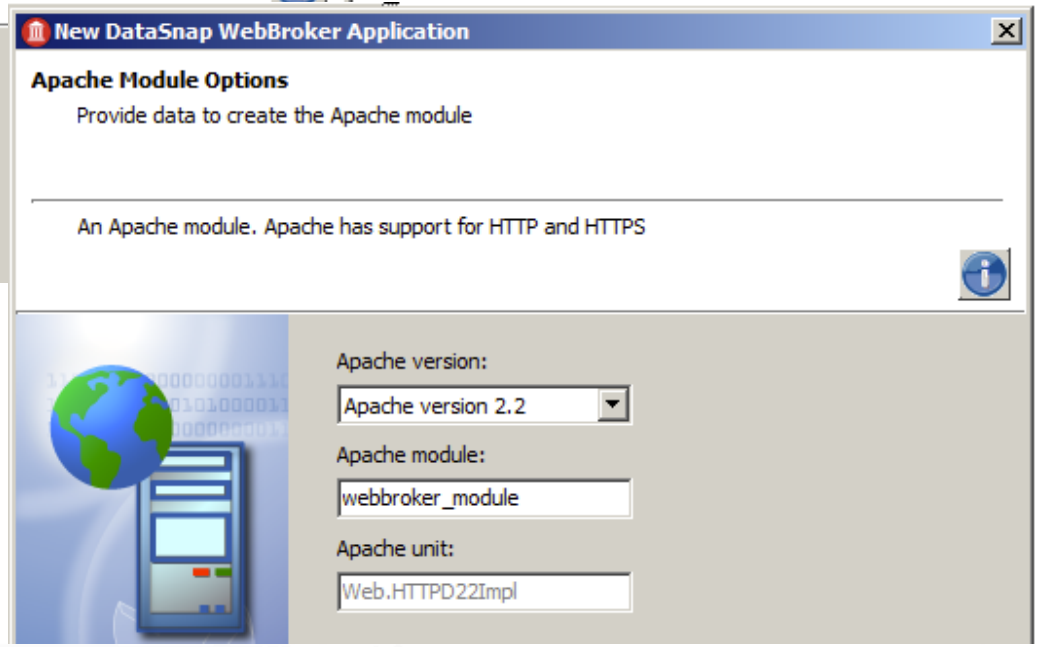
- Como distribuir meu servidor de aplicações no Apache?
  - Primeiro passo: Criar um novo projeto utiilzando a opção:
    File > New > Other > DataSnap Server > DataSnap WebBroker Application



DataSnap com FireDAC versus Apache

- Como distribuir meu servidor de aplicações no Apache?





DataSnap com FireDAC versus Apache

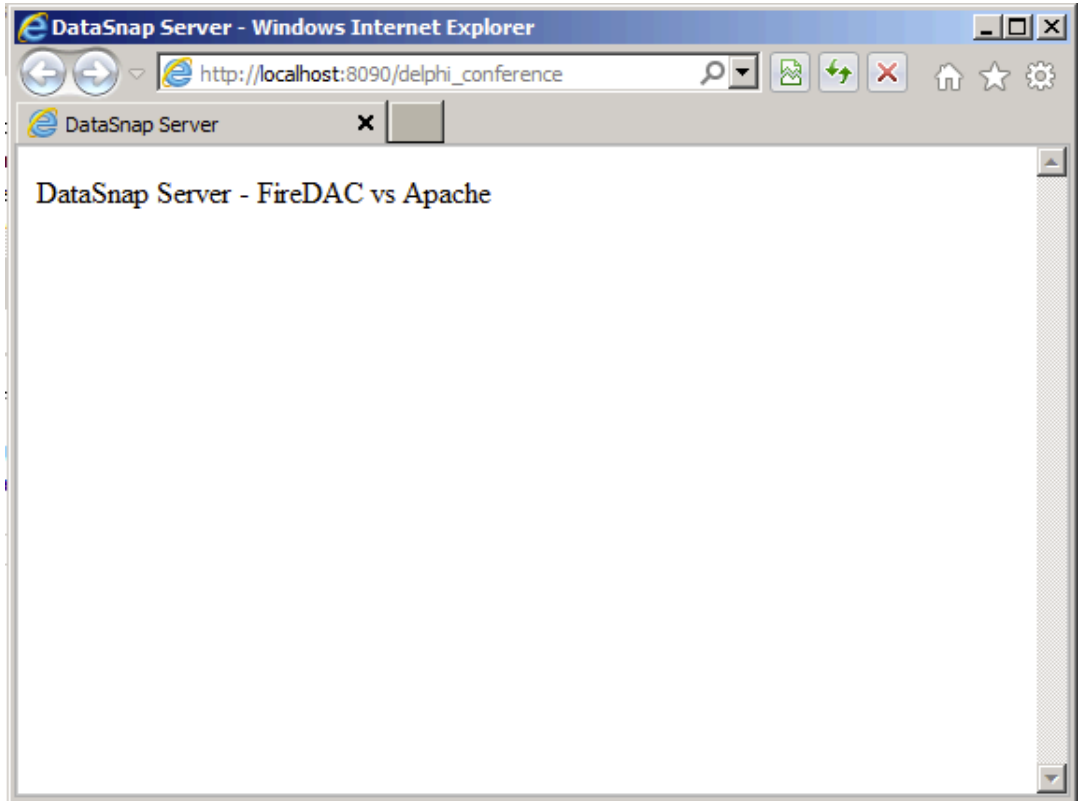
- Como distribuir meu servidor de aplicações no Apache?

Entrada no http.conf do Apache:

```
LoadModule delphiconference_module modules/
  mod_delphiconference.dll

 <Location /delphi_conference>
    SetHandler mod_delphiconference-handler
 </Location>
```

DataSnap com FireDAC versus Apache

- Como distribuir meu servidor de aplicações no Apache?
  - Apache dynamic link module

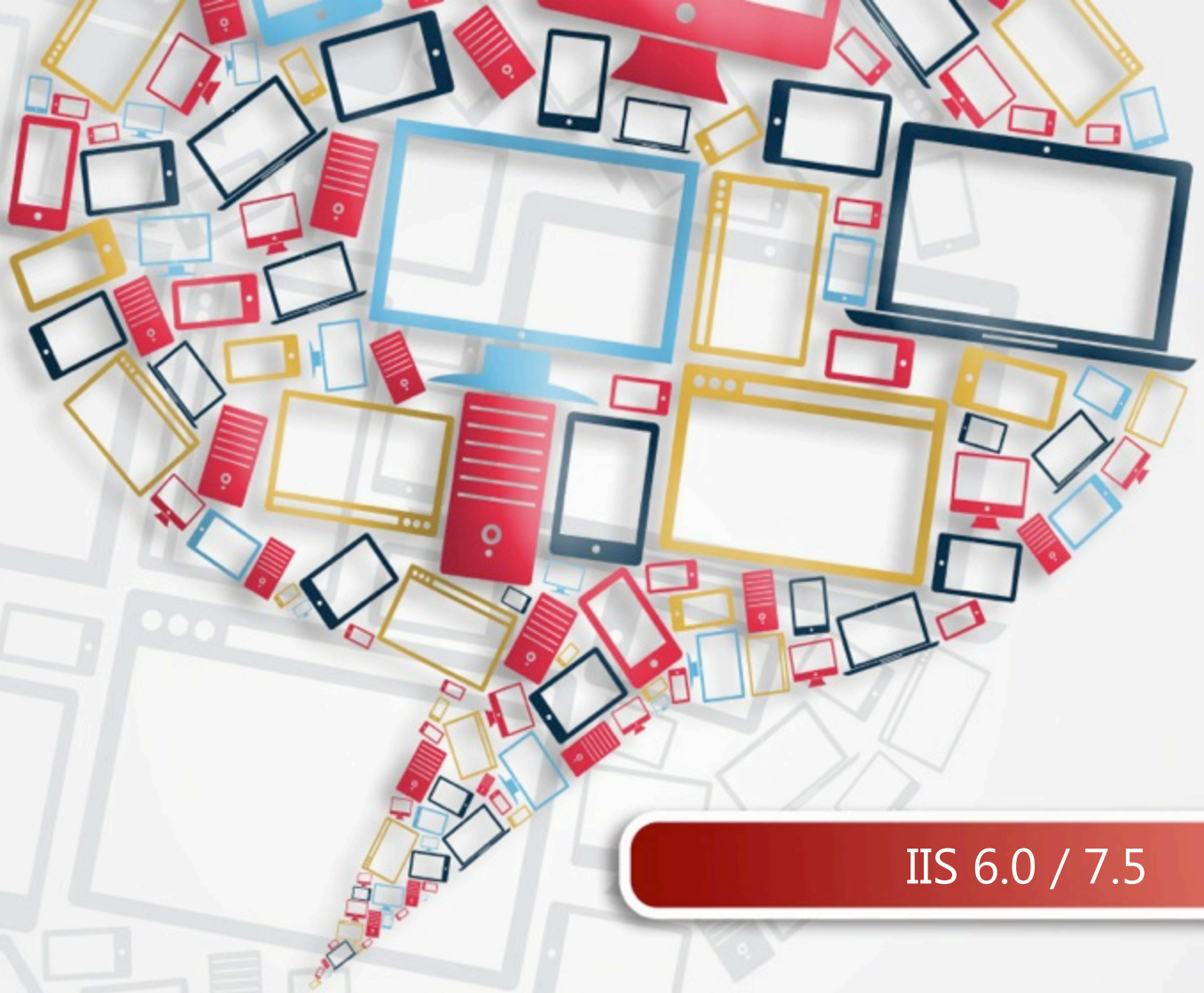


DataSnap com FireDAC versus Apache

- Como distribuir meu servidor de aplicações no Apache?
  - Como conectar-se agora ao Apache?
  - Modificar as propriedades:
    - Port: 8090 *a porta usada no Apache
    - UrlPath: delphi_conference
    - Host: retirar o localhost

```
procedure TForm2.SpeedButton1Click(Sender: TObject);
var LDataSetList: TFDJSONDataSets;
begin
  LDataSetList :=
    ClientModule3.ServerMethods1Client.GetDepartmentNames();
  FDMemTable1.Active := False;
  Assert(TFDJSONDataSetsReader.GetListCount(LDataSetList) = 1);
    FDMemTable1.AppendData(TFDJSONDataSetsReader.GetListValue(
  LDataSetList, 0));
end;
```

embarcadero



IIS 6.0 / 7.5

Distribuindo DataSnap Server

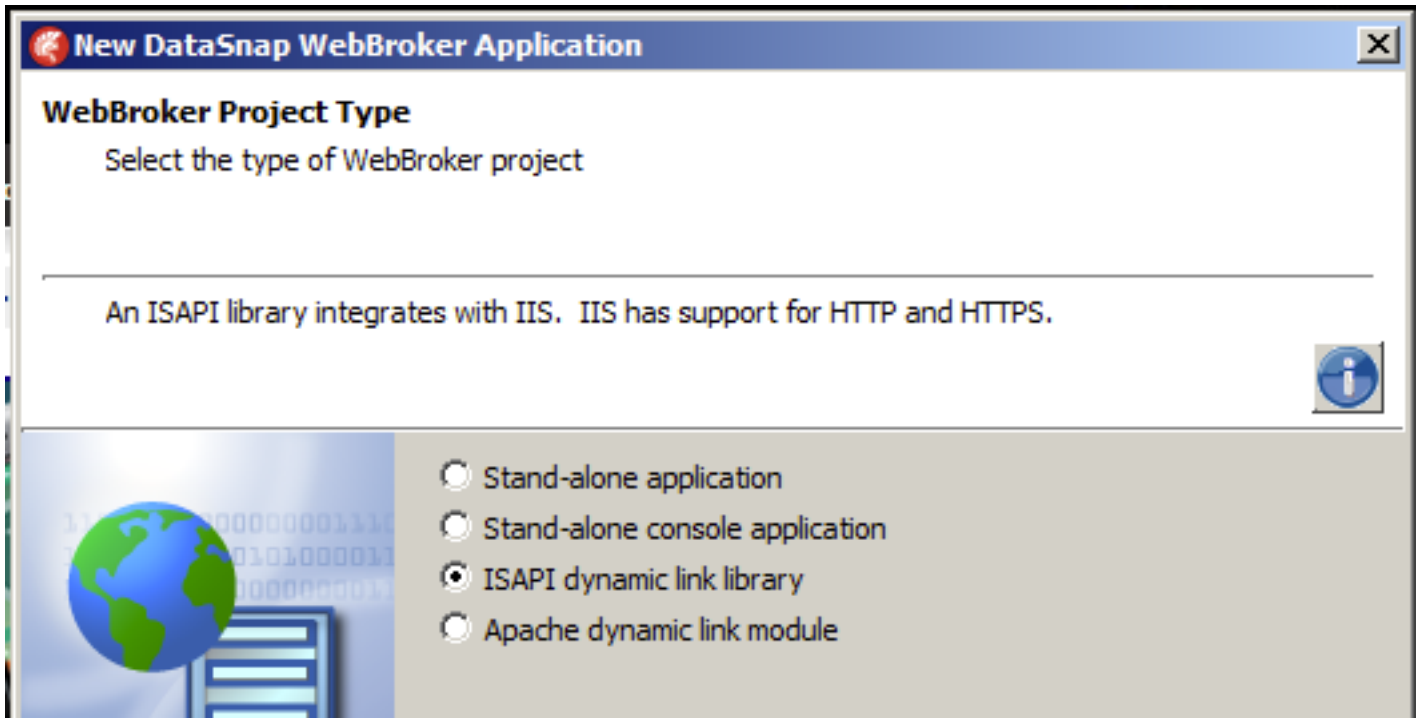
DataSnap com FireDAC versus IIS 7.5

- Como distribuir meu servidor de aplicações no IIS?
  - Modificar as propriedades:
    - Port: 8082 *a porta usada no Apache
    - UrlPath: http://localhost:8082/dc2014/IIS_Server.dll
      Host: retirar o localhost

```
procedure TForm2.SpeedButton1Click(Sender: TObject);
var LDataSetList: TFDJSONDataSets;
begin
  LDataSetList :=
    ClientModule3.ServerMethods1Client.GetDepartmentNames();
  FDMemTable1.Active := False;
  Assert(TFDJSONDataSetsReader.GetListCount(LDataSetList) = 1);
    FDMemTable1.AppendData(TFDJSONDataSetsReader.GetListValue(
  LDataSetList, 0));
end;
```

DataSnap com FireDAC versus IIS 7.5

- Como distribuir meu servidor de aplicações no IIS?



DataSnap com FireDAC versus IIS 7.5

- Como distribuir meu servidor de aplicações no IIS?
  - Distribuição Dica:
  - http://edn.embarcadero.com/article/40873

embarcadero

Bônus Pack

Failover e LoadBalance com Apache

Apache como LoadBalance e Failover

- Como utilizar o Apache como servidor de Failover e LoadBalance para aplicações DataSnap?

  - https://blogs.oracle.com/oswald/entry/easy_http_load_balancing_with

  - Módulos a ativar:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/
  mod_proxy_balancer.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Apache como LoadBalance e Failover

- Como utilizar o Apache como servidor de Failover e LoadBalance para aplicações DataSnap?

```
<Proxy balancer://DataSnap_Server_Balancer>
    BalancerMember http://localhost:8081
    BalancerMember http://localhost:8082
    BalancerMember http://localhost:8083
    BalancerMember http://localhost:8084
    Order allow,deny
    Allow from all
</Proxy>

ProxyPass / balancer://DataSnap_Server_Balancer/
```

asrsantos@gmail.com
www.tdevrocks.com.br
Twitter: @asrsantos
Twitter: @tdevrocks