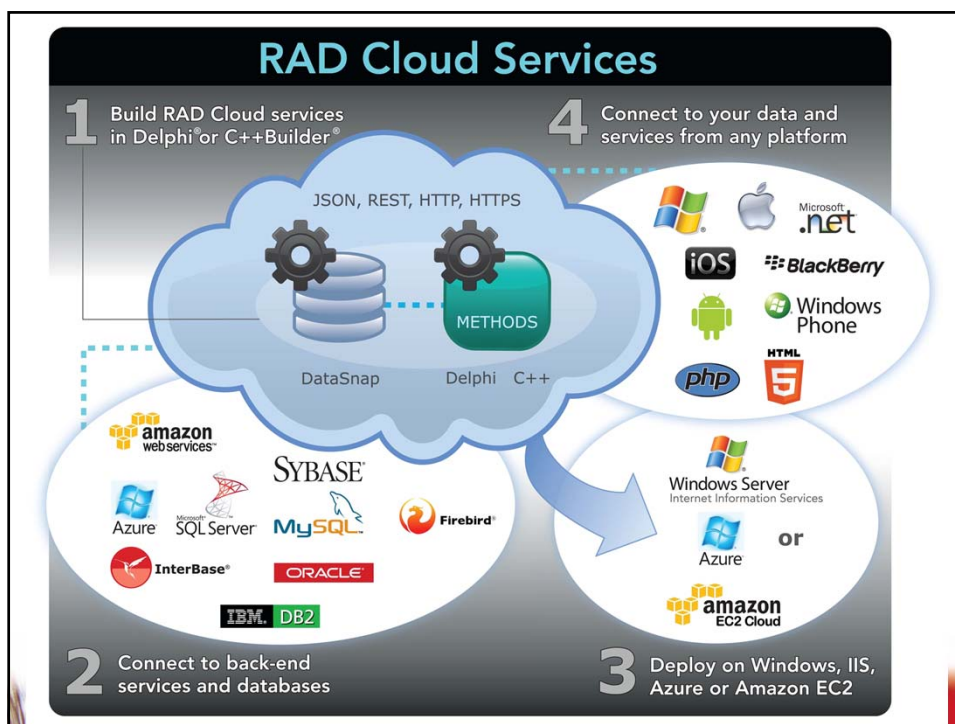


Agenda

- Introducción a RAD Cloud
- DBExpress
- Nueva API de la Nube
- ¿Qué hay de nuevo en DataSnap?



API de la Nube

¿ Qué es la API de la Nube?

“Una API de Delphi para acceder a los servicios de aplicación en las nubes de Amazon y Microsoft Azure.”



embarcadero

Rad Studio XE2 | www.danysoft.com/embarcadero

Danysoft

Soporte de servicios de la API



Almacenaje	Azure Blobs Service	Amazon S3
Tabla	Azure Table Service	SimpleDB
Cola	Azure Queue Service	Amazon SQS

embarcadero

Rad Studio XE2 | www.danysoft.com/embarcadero

Danysoft

Convenciones de nombrado

- Amazon y Microsoft utilizan diferentes nombres para las mismas cosas



Container	=	Bucket
Blobs	=	Object
Page	=	Multipart
ACL	=	ACL (Access Control List)

Microsoft Azure

- Blob Storage
 - Almacenaje no estructurado
 - Contenedor no definido
 - Para almacenar elementos muy grandes como por ejemplo: Imágenes, Video, Documentos, etc.
- Azure Tables
 - Almacenaje escalable y estructurado
 - Tipo de tablas NO SQL
 - Entidades almacenadas en una tabla
 - Diferentes conjuntos de propiedades diseñadas para diferentes tipos
- Queues
 - Mensajería persistente y asíncrona
 - Cada mensaje puede ser de hasta 8 kb

Amazon WS

- **Simple Storage Service (S3)**
 - Almacenaje de Blobs
 - Almacenaje no estructurado
 - Contenedor definido
 - Para almacenar elementos muy grandes como por ejemplo: Imágenes, Video, Documentos, etc.
- **SimpleDB - Tables**
 - Almacenaje escalable y estructurado
 - Tipo de tablas NO SQL
 - Entidades almacenadas en una tabla
 - Diferentes conjuntos de propiedades diseñadas para diferentes tipos
- **Simple Queue Service (SQS)**
 - Mensajería persistente y asíncrona
 - Cada mensaje puede ser de hasta 64 kb

Despliegue en la Nube

- **El gestor de despliegues soporta:**
 - Microsoft Azure
 - Amazon EC2

¿Qué hay de nuevo en DataSnap?

¿Qué hay de nuevo en DataSnap?

- Mejoras
 - Server
 - Interfaces REST
 - Callback
 - Authentication Manager
- Nueva monitorización y control de DataSnap

Server

- Información del soporte de las conexiones DataSnap HTTP en TSQLConnection y TDSRESTConnection
- Nuevo TDBXChannelInfo.DBXClientInfo (registro TDBXClientInfo) que informa:
 - Dirección IP
 - Protocolo
 - Nombre de la aplicación (cuando sea posible) – solo para HTTP

Soporte de HTTPS

- Para servidores DataSnap “Stand-alone”
- Para servidores WebBroker “Stand-alone”
- “Paso a paso” de DataSnap actualizados con el soporte HTTPS

Gestor de sesiones

- Soporte de sesión para el protocolo TCP/IP
- Soporte para almacenar y devolver TObject en las sesiones
- Sesiones creadas y destruidas con el protocolo TCP/IP informan los AddSessionEvent y RemoveSessionEvent
- Soporte para múltiples túneles de “Callback” en el interfaz DataSnap REST

Gestor de túneles de Callback

- La clase TDSCallbackTunnelManager en la unidad DSServer
- Permite el seguimiento cuando:
 - El “Callback” se añade o se quita
 - El Túnel se crea o se cierra

Callbacks

- Los callbacks pesados soportan ahora soportan ahora emisiones a callbacks específicos
- GetAllChannelNames añadido al DSServer para obtener el nombre de todos los canales de callback

Eventos del lado del cliente

- Notificaciones cuando sea que el túnel del gestor se abre o cierra; o cuando se añaden o quitan canales, disponible para:
 - Eventos de cliente Delphi DBX
 - Eventos de cliente Delphi REST
 - Eventos de cliente JavaScript REST

Eventos de cliente Delphi DBX

- El comp. **TDSClientCallbackChannelManager**, tiene el evento **OnChannelStateChange**
- Firma del método:
(Sender: TObject; const EventItem: TDSClientChannelEventItem);
- Tipo de evento **TunnelClosedByServer** para conexiones cerradas por el servidor o cuando se pierde la conexión.

Eventos de cliente Delphi REST

- El comp. **TDSRestClientChannel** component, tiene el evento **OnChannelStateChange**
- Firma del método:
(Sender: TObject; const EventItem: TDSRESTChannelEventItem);
- El tipo de evento **ChannelClosedByServer** para pérdidas abruptas de conexión con el servidor

Eventos de cliente JavaScript

- Establecen una función de evento para ser invocada cuando los cambios sucedan en el canal o uno de sus callbacks

```
var channel = new ClientChannel (clientId, channelName);  
channel.onChannelStateChange = HandleChannelEvent;
```

- Si el servidor se cierra correctamente, el cliente obtiene un evento de tipo "ClientChannel .EVENT_CHANNEL_STOP"
- Si se pierde la conexión abruptamente el cliente obtiene un evento del tipo "ClientChannel. EVENT_SERVER_DISCONNECT".

Soporte de session en JavaScript

- initSessionData function (en serverFunctionExecutor.js) ahora toma un segundo parámetro opcional.
- El parámetro se fijará a la "Cookie" de Id general (dssessionid).
- Permite que tengas un completo control sobre que aplicación usa un ID/cookie de sesión.

Resultados JSON y llamadas REST

- El evento OnResult se ha añadido a los componentes TDSHTTPWebDispatcher y TDSHTTPService
- Se permite cambiar el formato del resultado JSON en las llamadas REST.
- Firma del método:
(Sender: TObject; var ResultVal: TJSONValue;
const Command: TDBXCommand; var
Handled: Boolean);

Parámetros de consulta en llamadas REST

- GetInvocationMetadata().QueryParams
- Devuelve un TStrings que contiene pares de clave – valor para todos los parámetros de consulta pasados a través de la URL de la llamada REST

Monitorización y control de DataSnap

Monitorización y control de DataSnap

“Los servidores DataSnap en XE2 son capaces de fácilmente monitorizar las conexiones, y cerrar cualquier conexión que deseen.”

¿Qué puedes hacer?

- Tomar el control de las conexiones cliente
- Monitorizar las respuestas de los clientes
- Limitar el número de peticiones por conexión
- Mejorar la seguridad
- Más...

¿Cómo funciona?

- Los componentes TDSTCPServerTransport tienen dos nuevos eventos:
 - onConnect
 - onDisconnect
- Las conexiones se enlazan con un ID de sesión

Escuchando los eventos TCP/IP

```
1 procedure TForm1. ServerTransportConnectEvent(Event: TDSTCPConnectEventObject);
2 begin
3     //Add both the connection and Channel (TDSTCPChannel) to a dictionary for later
4     use
5     FConnections.Add(TIdTCPConnection(Event.Connection), Event.Channel);
6 end;
```



```
1 procedure TForm1. ServerTransportDisconnectEvent(Event: TDSTCPDisconnectEventObject);
2 begin
3     //Remove the connection and its associated channel from the dictionary
4     FConnections.Remove(TIdTCPConnection(Event.Connection));
5 end;
```

Detectando desconexiones anormales

- Por defecto, en evento OnDisconnect no será notificado al cliente si este abruptamente pierde la conexión
- El comportamiento depende de la configuración del OS
- Utilizamos los métodos TDSTCPChannel para controlar el comportamiento de cada conexión:
 - EnableKeepAlive
 - DisableKeepAlive

Como utilizar “EnableKeepAlive”

- Si la conexión está inactiva durante más de 10 segundos, se envía un paquete para chequear si el cliente está ahí.

Event.Channel.EnableKeepAlive(10000);

Como usar “EnableKeepAlive”

- Si el cliente no responde el OS volverá a reenviar el paquete un número X de veces (en Win 7 serán 10).
- Se puede especificar el número de milisegundos a esperar entre reintentos si el cliente no responde al paquete.

Event.Channel.EnableKeepAlive(10000, 100);

Habilitar KeepAlive

- Disponible en el componente TDSTCPServerTransport
- Tres nuevas propiedades publicadas
 - KeepAliveEnablement - (Set: Default, Enabled, Disabled)
 - KeepAliveTime - Integer, valor en milisegundos
 - KeepAliveInterval - Integer , valor en milisegundos

Cómo usar el DisableKeepAlive

- Obtener el canal asociado a una conexión y llamada dadas:
DisableKeepAlive;

Cerrando conexiones de clientes

- Las conexión puede cerrarse en cualquier momento
- El canal nos suministra la instancia de la conexión

```
//Get the associated Channel for the given connection, and if successful close it  
if FConnections.TryGetValue(Connection, Channel) then  
    Channel.Close;
```

Demostración

- Seguimiento de todas las conexiones TCP/IP y sus sesiones correspondientes
- Como cerrar manualmente una conexión TCP/IP desde el servidor
- Como detectar cuando un cliente ha perdido la conectividad a Internet mientras está conectado.