

RAD SERVER

Marco Cantu, Delphi Product Manager marco.cantu@embarcadero.com Twitter: @marcocantu



AGENDA (INTRODUCTION)

- What is RAD Server
- Technical foundations of RAD Server
- JSON support
- Returning JSON for JavaScript and Ext JS clients
 - Supporting HTML and related static files
- Users and Permissions
- Analytics

AGENDA (ADVANCED)

- Exposing FireDAC database data
 - Streaming
 - Batch Move
 - MetaData management
- Advanced users management
 - Custom logins
- Distributed resource managements (ThingPoints)
- Notes on Deployment

PART I: RAD Server Introduction

WHY RAD SERVER?

- Rapidly connect your apps to your enterprise databases and services hosted on-premises or in a private cloud
- Make enterprise data easily available on every device, keeping it secure



 Includes user and groups usage analytics, along with REST API calls analytics

RAD SERVER QUICK OVERVIEW

THE PERFECT BACK-END FOR DELPHI AND C++BUILDER APPS



REST End Point Publishing



Integration Middleware



Application Services



EASY REST API END-POINT CREATION, PUBLISHING, AND MANAGEMENT

RAD Server makes it fast and easy to build flexible back-end servers for your multi-tier application solution. Developers simply load Delphi and C++ business logic into the server, and managed REST/JSON API end-points are generated. Developers or admins can easily configure API level access control to user groups, and measure and analyze application usage at the API, user, or service level. Since your Delphi and C++ APIs are published as heterogeneous REST/JSON end-points, RAD Server easily supports virtually any client type from VCL and FMX Desktop, Mobile and Wearable clients to popular JavaScript Web Frameworks. RAD Server easy to build and deploy robust heterogeneous multi-tier solutions.

{API} REST End-Point Publishing

Easy to use API publishing of business logic. Any Delphi or C++ code can be hosted as an API and auto-published as REST/JSON endpoints which are measured and managed by RAD Server.



Access Control

Group and user level access to control to all application APIs. Control who has access to what functionality. All access is user authenticated.



API Analytics

All REST API end-point activity is recorded and measured for robust statistics tracking and analytics. Analyze user, API, and services activity to gain insight into how your application is being utilized.



Desktop, Mobile & Web

All C++ and Delphi code hosted on RAD Server are published as REST/JSON end points consumable by any type of client for extreme flexibility and futureproofing.

Portes Substitute #		Anvious on your EMS S	Renter
Constant Asso Asso Automatic		Two faires	
		E B Prest	
Not looked		and Tandardarap	
Fut 800		-	
10.44		A Maderal Jury	
	Trees (1996)	Statester	
Test Convention	ta tuta no	Tutter.	
	Pulles syllethide #		
tiers desait benchmen Aut-	Greater New Sec. Advenue.	Technology	
A store		and Texture Dive	
Advantation Proceeding and Advantation and Advanced Adv	Fat bahat		
inally Data ICE/IEF-IEFE.	5vi 800 [] x755	Tarlardinipale	
Han Anderson 24/08/24 Will.	10,548	and Texterina	
wentered Medicity 2001		-	
		theisedirege	
	Test Convertion	Contractor	
	Des Josep Justices, Aut		
	les legt	- in	
	Second Tech Mesope and init 2017	Di Dente di	
Man Int All Dates			
C B + -	a teachart		
C D + -	340	10	
	"Minicipation" The s		
		-	
	Sentilister	(inc	

Integration Middleware

HIGH PERFORMANCE INTEGRATION WITH DATA, CLOUD, AND SMART DEVICES

RAD Studio includes high performance integrations that make it easy to connect Enterprise Data, Cloud Services and Smart IoT devices into your multi-tier application. RAD Studio's Enterprise DACs enable fast pooled connections to Oracle, Microsoft SQL Server, DB2, Sybase, and more. Client connection pooling also eliminates complex and heavy weight RDBMS client drivers and configurations. Clients applications connect to RAD Server via lightweight REST for data access, application services, and application API calls. RAD Server includes the new IoT ThingConnect and ThingPoint IoT Edgeware making it easy to connect a wide array of wireless smart devices and sensors into your application solution, even in remote and difficult to access locations.



Enterprise Data

High performance built-in connectivity to all popular Enterprise RDBMS servers. Based on FireDAC developers have easy connectivity and a wealth of features for connecting with and working with data from a variety of sources.

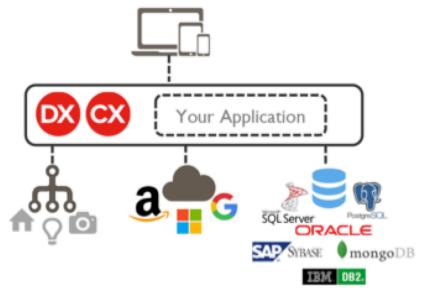


IoT Connection components enables fast and easy integration of IoT Smart devices into your app clients or back-end. RAD Server's IoT Edgeware extends it's reach to where your Bluetooth IoT devices are physically located.



Cloud Services

Easily integrate REST cloud services from a variety of cloud, social, and BAAS platforms such as Google, Amazon, Facebook, Kinvey, Parse and more.





Application Services

KEY BUILT-IN CORE SERVICES TO POWER YOUR APPLICATION

RAD Server includes a powerful set of built-in core servers to power your application back-end eliminating the need to build the key foundational components of your server application. Core services like User directory services, authentication, and access control, push notifications, JSON data-storage, and user proximity and indoor/outdoor user location tracking and fencing.

Push Notifications

Send programmatic or on-demand notifications to your application users.

R User/Group Management

Create and manage users, groups, and access control via the RAD Server management portal.

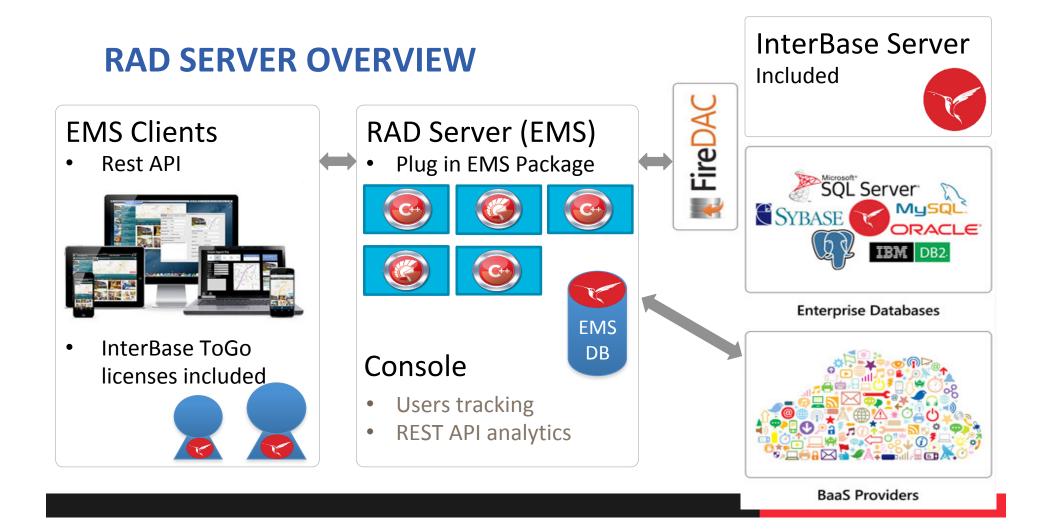
Built-in Secure Datastore

Easily store and retrieve JSON data securely and without requiring a separate database server.

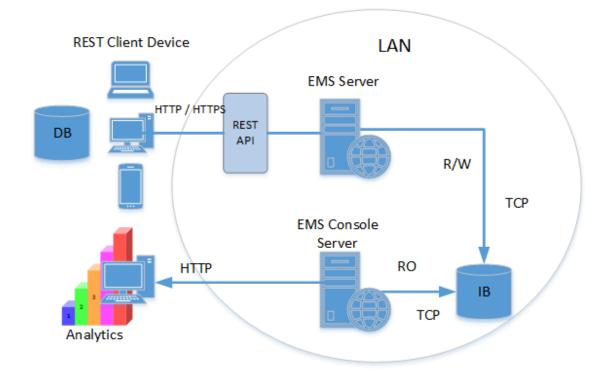
User Location/Proximity

Track user movement both indoors and outdoors, and respond to proximity events when users enter or exit custom beacon zones or approach designated beacon points.





RAD SERVER ARCHITECTURE



RAD SERVER ARCHITECTURE

- RAD Server Server: Exposes REST API
 - Built-in resources
 - Users, Groups, Version
 - Custom resources
 - Developers create packages with modules that register new resources
- RAD Server Database: Stores users and analytics data
 - User InterBase, license is included
- RAD Server Console: Track Status and Usage

FIRST STEPS IN PUBLISHING APIS

- Creating an EMS resource package
 - Implementing the REST APIs
 - Executing the server
 - Calling the APIs form a browser or a client app
- Examining the console
 - And the user management app
- Configuration wizards
 - And the EMS.INI file key settings

RAD SERVER PACKAGES

- Add resources to RAD Server
 Mapped to URI
- Wizards help create packages and add a resource to an existing package
 - RAD Server Package Wizard, optionally adds first resource
 - RAD Server Module Wizard, for more resources

间 EMS Package Wizard		X	
Package An EMS package regis clients.	ters resource(s) and resource endpoints that may be called by EMS		
Create a package with resource. A resource extends the REST API of the EMS server.			
Package	Create empty package		
Resource	Oreate package with resource		
Endpoints			

RAD SERVER CLIENTS

- Experimenting with a browser
- Using the REST debugger
- Clients make REST API calls
 - Any HTTP library would work
- BaaS architecture for the client side
 - EMS Provider component to connect
 - Use BAAS Components for services
 - TBackendQuery, TBackendUser, TBackendEndpoint
 - Handy TEMSFireDACClient component
 - For transferring data and delta packets between client and EMS server

MORE ON JSON SUPPORT

- JSON = JavaScript Object Notation
- JSON but not only
 - JSON objects and strings
 - JSON streaming
- Returning static files
 - For testing locally HTML + JS apps

RESOURCES AND ENDPOINTS

- Get and GetItem
- Put, Post, Delete
- Query params and other HTTP request information

USERS AND PERMISSIONS

Basic users management

- User endpoints
- Users authentication
- BackendAuth component
- Permissions
 - Users authorization
 - Groups

ANALYTICS

Built-in user and API analytics

PART II: RAD Server Advanced

RAD SERVER AND FIREDAC

- FireDAC JSON Streaming
 - SaveToStream, LoadFromStream with sfJSON format
 - Expose as resource, consume via BackendEndPoint
- Using the BatchMove architecture
- Use FDSchemaAdapter
 - Captures master/detail relationships on server
 - Exposes the same on the client
 - Direct hooks to streaming, invocation

USING FDSCHEMAADAPTER

- Let's look at a demo based on FDSchemaAdapter
 - A RAD Server
 - A VCL desktop EMS client
- Using:
 - FireDAC with master/detail structure
 - Streaming based on FDSchemaAdapter
- Demo ships with RAD Studio

ADVANCED FEATURES

- Advanced users management and custom login modules
- Integrated push notifications support
- EdgeModules or ThingPoints
- Multi-tenancy

DOCUMENTING YOUR APIS

- Automatic doc support
- Publishing custom API endpoints docs
 - Swagger RESTful API Documentation Specification
 - JSON or YAML formats

DEPLOYING RAD SERVER

- InterBase Server
- RAD Server: Server and Console applications
 - Stand alone servers for development (5 users)
 - Deployment ISAPI/Apache library
- Run on Win32 or Win64
- Custom APIs are deployed using EMS packages
 - Standard Delphi or C++Builder BPL files
- EMS.INI

THANKS! Any questions?

You can find me at: @marcocantu marco.cantu@embarcadero.com