Practical Essbase Web Services

For Fun and Profit

Jason Jones

Key
Performance
Ideas

Jason Jones

- Essbase
- Programming
- Mobile development
- ODI
- Blogging
- Open source

Agenda

- Web services in a nutshell
- Essbase connectivity options
- Essbase Web Services sample walkthrough
- Closing thoughts
- Questions (feel free to ask during!)

Web Services Essentials

- Use HTTP for communication (like web pages)
  - Often easier to get to than various ports
- Doesn't lock client to a particular technology/ library
- Often use XML for messages
  - JSON is the new hotness
- Statelessness

Choosing a Connection Technology

- Compatible with solution technology
- Performance
- Solution deployment
- Ease of development
- Existing skillset
- Versioning

Programmatic Essbase Data Retrieval

- Visual Basic API
- C API
- MaxL
- XMLA
- Java API
- Essbase Web Services

Visual Basic API

- Advantages
  - Good fit for automating within Excel
- Disadvantages
  - Limited usefulness in enterprise solutions

C API

- Advantages
  - Windows solutions (geared towards Visual C++)
- Disadvantages
  - Lower level (development time?)
  - Windows solutions
  - Feasibility on other platforms?

MaxL

- Advantages
  - Simple
  - Possible to fetch data with MDX of MaxL
- Disadvantages
  - Not a robust enterprise solution
  - Very narrow usefulness

XMLA

- Advantages
  - Used by more than Essbase
  - Probably a good fit if your product wants to “check the box” with respect to Essbase compatibility
  - Connect directly from client
  - No need for your own separate middle tier
- Disadvantages
  - Abstracts away/hides Essbase functionality

Java

- Advantages
  - Robust API
  - Extensive functionality
  - Decent documentation
  - Works on any platform Java works on
  - I'm biased towards this ☺
- Disadvantages
  - Requires Java (i.e., can't use on iOS)

Essbase Web Services

- Advantages
  - No need to write middle tier
  - Language agnostic
  - WSDL/SOAP is common paradigm
- Disadvantages
  - No middle tier ☺
    - You might just want one

Getting Essbase Web Services Setup

- Overview
  - APS setup with Essbase Web Services
    - Requires modern Essbase deployment (11.1.2.2)
  - Proxy generation
    - Built off of a WSDL file
  - Custom Programming

APS Configuration/Confirmation

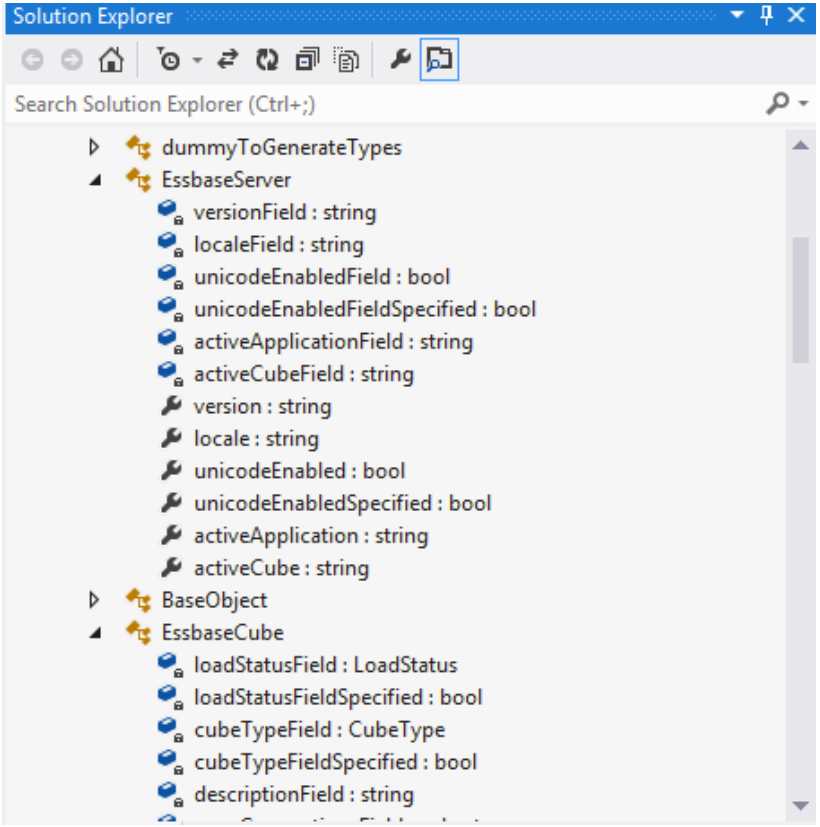
- Web service root location
  - Verify services available/correct URL
  - E.g. http://localhost:9000/essbase-webservices/DatasourceService?wsdl
  - There are multiple endpoints
- Web Service WSDL stubs
  - Used to generate proxies

Proxy Generation Example

- C#
  - svcutil.exe
  - Generates classes based on WSDL
  - Bug in current WSDL (fixed?)
- Other Languages
  - Objective-C, JavaScript, PHP, Python, Ruby
  - Google “<your language> WSDL generator”

Programming

- Connect to server
- Interact via proxy classes (C# in this example)
- Will probably want to wrap this API with your own



Endpoints

- DatasourceService
- AdminService
- QueryService

Examples of DS Objects Available

- EssbaseServer
  - Version
  - Active Cube
  - Active App
  - …
- EssbaseApplication
  - Load Status
  - Num Connections
  - Elapsed Time
  - …
- EssbaseCube
  - Num Dimensions
  - Cube Type
  - Cube Calc Info
  - …
- CubeRuntimeInfo
  - Block Size
  - Total Blocks
  - Compression Ratio
  - …

Web Services Sweet Spot?

- Likely Use Case Parameters
  - Not writing solution in Java
  - Not writing solution in Windows/C++
  - Can't/won't write custom middle tier
  - XMLA not deep enough
- So…
  - *Want to access Essbase data/metadata directly (no additional middle tier) from a technology other than Java/C++ that can't wrap the C API.*
  - ADF, JavaScript, C#

Statelessness

- Ability to make a request that is not dependent on previous requests having been made
- Popular development paradigm for good reasons (RESTful web services)
- Works well with ephemeral connections, particularly mobile
- Example data retrieval request
  - Connection information (server, app, db, credentials)
  - Base grid + grid retrieval options + operations

Closing Thoughts

- Essbase Web Services have a specific use case; consider all of your options
- If you are serious about connecting to Essbase servers from your product, consider writing your own middle-tier
  - Better opportunities to return results in your native format (JSON, custom XML)
  - More opportunities for performance tuning/caching
- Rules of good programming apply
  - Common library, structure, etc.

Thank You! Questions?

Jason Jones

Direct 206.427.1373

Email jjones@keyperformanceideas.com

Twitter @jwj