

# What is Grails4Notes<sup>(TM)</sup> ?

Justin Hill, CTO, Prominic.NET, Inc.

Copyright (c) 2014. All rights reserved. Trademarks mentioned herein are the rights of their respective owners.

- ◆ About me and Prominic:
- ◆ Co-founder and CTO of Prominic.NET, Inc.
- ◆ OpenNTF.org board member & cloud hosting provider
- ◆ Prominic has been in the Notes/Domino hosting business since 1998
- ◆ We host many high availability, multi-data center clustered SaaS applications built using IBM<sup>(TM)</sup> software technologies
- ◆ Prominic has helped with scalability, reliability, and cloud virtualization — emphasizing the value of new technologies
- ◆ We have deep expertise in multi-tenant performance solutions
- ◆ Strive to be the trusted technology advisor for our customers

- ◆ Audience: IBM Lotus Notes/Domino<sup>(R)</sup> veterans
- ◆ Anyone unfamiliar with Notes should read Julian's excellent "what is it?" summary at:
- ◆ <http://www.nsftools.com/misc/WhatIsNotes.htm>
- ◆ Among other things, Lotus Notes was the first true NoSQL database on the market
- ◆ (With many other features built-in like PKI security and e-mail, but they are not relevant to this talk...)

- ◆ You probably know what Notes is very well
- ◆ You may not know much or anything about Grails
- ◆ We will get to it... after we focus on Notes for a bit
- ◆ Back to the current market for popular web application databases

- ◆ Now there are many Notes-like NoSQL databases:
- ◆ MongoDB
- ◆ CouchDB
- ◆ Many more...
- ◆ And new graph style (think social network) databases:
- ◆ Neo4J
- ◆ <http://redpilldevelopment.com> is leading the way for Notes-based Graph databases

- ◆ Today, many new technologies are free and “good enough” to be worthy of CIO & developer attention
- ◆ Ruby on Rails, PHP, .NET, etc.
- ◆ A million Javascript frameworks... (Angular, jQuery...)
- ◆ Some things Notes/Domino first made easy have become commoditized and free:
- ◆ Rapid app prototyping for unstructured data
- ◆ Create / read / update / delete (CRUD) for your prototype with automatic SQL DB made Rails famous!

- ◆ Some of the best minds in the Notes industry will even help you migrate to MongoDB
- ◆ <http://ldcvia.com>
- ◆ But before you jump ship, think through things a bit...
- ◆ Trading one unstructured document data store (Notes) for another (MongoDB) may not help as much as you hope in the long run
- ◆ Do your homework on security, developer head count requirements, and market prices for “full stack engineers”
- ◆ Compare a total re-write (expensive & time consuming!) to hiring the best Domino/XPages developers and mix in some new tricks

- ◆ What new tricks? Notes has been around forever...
- ◆ But so have light bulbs, computers, and cars
- ◆ Those inventions have all continued to be enhanced
- ◆ Notes has had recent innovations as well, such as XPages and DAOS
- ◆ A big upcoming performance improvement feature I am not allow to discuss publicly yet

- ◆ IBM's introduction of XPages has been great
- ◆ But Java Server Faces are not a panacea - they focus on the User Interface and Mobile, NOT the database storage format
- ◆ You probably have a lot of business logic and data in the NSF database format
- ◆ If you are in XPages, you probably have been doing Notes for a while
- ◆ Your databases may have gotten bigger and slower
- ◆ XPages does not necessarily solve your data set size problem
- ◆ DAOS, high speed SSD storage (>> IOPS) have helped

- ◆ Facebook<sub>(TM)</sub>, Twitter<sub>(TM)</sub>, and Google<sub>(TM)</sub> have massive data sets today (and millions of servers — so not really equivalent)
- ◆ You should not have to worry about NSF's having too many views or documents
- ◆ After all, a million documents for Notes is a lot in a single NSF, but nothing for a relational DB
- ◆ RDBMS was all the rage for a while, but lately not as much...
- ◆ Because NoSQL has gained momentum for web development
- ◆ You already have a good NoSQL database with a proven track record of keeping data safe and secure when used properly

- ◆ Remember DB2NSF and the promise to have critical Notes data in a RDBMS?
- ◆ Exciting Lotusphere circa-2007 demo of “select count(\*) from maildb.inbox”
- ◆ <http://vowe.net/archives/007431.html>
- ◆ Then we saw how much work it was to use it and build field mappings
- ◆ <http://www.kalechi.com/comments.nsf/dx/06022009043158PMBLIT62.htm>
- ◆ You want to have your cake and eat it, too!
- ◆ You don't want to waste time defining SQL tables just to put a simple form together...
- ◆ Sometimes your Notes forms have been used heavily and now you need to refer to data made from other Notes forms — to generate a report, etc.
- ◆ Businesses need reports, so we end up writing custom Agents in Java or Lotuscript
- ◆ Structured Query Language (SQL) was invented to solve those ad-hoc JOINS and SELECT statements

- ◆ Sure, there is the Notes SQL driver & BIRT server
- ◆ <http://www.ibm.com/developerworks/lotus/documentation/notessql/>
- ◆ It will let you do some queries, but it relies upon the NSF database format
- ◆ Which still gives you no control over query execution operations (ie: how the DB answers your question!)
- ◆ DB2 Visual Explain would have let you optimize any query with custom indexes — a key RDBMS feature

- ◆ But what if you have a million documents (records) in your Notes database?
- ◆ And you have made 300+ views to get different pieces of data out
- ◆ And people are telling you the app is slow now
- ◆ How do you fix this and keep your users happy without buying expensive, ultra-high speed storage for your servers?

- ◆ A lot has happened since IBM tried to introduce DB2NSF and then phased it out
- ◆ They probably had valid reasons for not continuing to support the product
- ◆ Maybe it was so they could have the resources to deliver XPages...
- ◆ Regardless, we as a developer community often need more flexibility on large data sets

- ◆ Maybe it is time to try mixing something new into our tool set
- ◆ Without having to start from scratch in an entirely new language
- ◆ While retaining the value of everything we already know and can rapidly develop apps in

- ◆ Have you heard of Grails?
- ◆ It is like Ruby on Rails, but built on the multi-platform & enterprise-grade Java Virtual Machine
- ◆ It is open source and free
- ◆ It can be called by Java code, and can call Java code: It co-exists with Java extremely well.
- ◆ It adds a dynamic language (Groovy) similar to Ruby
- ◆ Plus automatic SQL RDBMS object-relational mapping “Grails” just like the “Rails” part of RoR

- ◆ How would we use Grails in a Notes or XPages application?
- ◆ It turns out that you don't really have to use it much at all, but it can still help you a lot
- ◆ Again: you can get an amazing amount of benefit from Grails with virtually no effort
- ◆ How?
- ◆ By adding a bit more information to your Notes fields!

- ◆ Most of us build “relational-like” Notes views and forms
- ◆ There is usually a field with an ID of some kind
- ◆ Other Notes forms usually refer to this field
- ◆ Typical example:
- ◆ Invoice form has a CustomerID field
- ◆ Customer form has a CustomerID field
- ◆ Views for “Invoices” and “Customers”

- ◆ This is how we all usually build Notes apps
- ◆ So put this relationship detail directly into the fields!
- ◆ Keep the definition with the field in Domino Designer using a Lotuscript method with a REM statement
- ◆ Select which fields are important for use in SQL
- ◆ Even add some formal documentation to the field to describe it (like a JavaDoc reference)
- ◆ This is starting to sound like we might have a formal business object definition!

- ◆ The annotation of your the fields on your Notes form is kept WITH the field in the NSF
- ◆ It looks like something like this:
- ◆ Sub Grails4NotesPropertiesDONOTMODIFY
- ◆ %REM
- ◆ This function is used to store data for the Grails4Notes application.
- ◆ Do not edit this directly - use the Grails4Notes Designer application.
- ◆ -----
- ◆ \$GrialsDomainClassPackageName=com.yourcomanyname;
- ◆ \$GrailsSanitizerType=notSanitizer;
- ◆ \$TableName=Invoices;
- ◆ \$GrailsDomainClassName=Invoice;
- ◆ %END REM
- ◆ End Sub

- ◆ So what's next after annotating the fields?
- ◆ Automatically generate corresponding Grails class objects which represent Notes forms
- ◆ Grails will, in turn, automatically generate an SQL table!
- ◆ You can use Navicat<sup>(TM)</sup> Data Modeler to quickly see an Entity-Relationship-Diagram of your Notes data
- ◆ <http://navicat.com/products/navicat-data-modeler>
- ◆ It just takes a minute and few clicks to have a formal map of your Notes structure — it is really cool to see

- ◆ Structure is great, but you really want the data
- ◆ The automatically generated Grails classes for your Notes forms inherit from a parent superclass
- ◆ The parent superclass has a built-in method `loadFromDominoDocument( Document current_doc)`
- ◆ The import ordering is important so relationship integrity can be enforced in the RDBMS schema
- ◆ So the import tool will let you select which databases and which views, in which order, to import!

- ◆ For example:
- ◆ Customer documents should be imported before Invoice documents
- ◆ This allows the Invoice's reference to the CustomerID to be satisfied
- ◆ The RDBMS database would not accept an Invoice reference to a missing CustomerID
- ◆ So now you have your data in both Notes and SQL!

- ◆ It is probably time for you to brush up on your SQL JOIN statements:
- ◆ [http://www.w3schools.com/sql/sql\\_join.asp](http://www.w3schools.com/sql/sql_join.asp)
- ◆ Instead of creating a new Lotuscript or Java agent the next time you want a report, try something like this:
- ◆ `SELECT invoices.total, customers.name`
- ◆ `FROM invoices`
- ◆ `INNER JOIN customers`
- ◆ `ON invoices.customerid = customers.customerid;`

- ◆ How would you use this to speed up existing apps?
- ◆ Web apps are especially easy to analyze
- ◆ Domino HTTP logs tell precisely how long each URL operation takes
- ◆ We wrote an analyzer to rank your HTTP logs and show you those > 6 seconds long (time configurable)
- ◆ Examine the WebQueryOpen, WebQuerySave, Embedded Views, etc. of those URLs
- ◆ Migrate the time consuming lookups to SQL queries!

- ◆ Your Domino Classic and XPages apps can use the SQL database drivers over JDBC or ODBC to the RDBMS
- ◆ Postgres SQL is an open source, enterprise-grade RDBMS, easy to use on multiple platforms including Mac<sup>(TM)</sup>
- ◆ Eliminate many unnecessary views from your NSF when you can use SQL queries now
- ◆ [http://www.martinscott.com/struturo.nsf/ID/tech articles perf engr](http://www.martinscott.com/struturo.nsf/ID/tech%20articles%20perf%20enrg)
- ◆ Has lots of great information on performance tuning Notes databases

- ◆ So what is Grails4Notes then?
- ◆ A tool to bring the best of Grails to the Notes world
- ◆ We emphasized minimizing your learning curve
- ◆ Use Grails4Notes to formalize your thinking about your Notes fields and form structures
- ◆ Automatically document your Notes fields, forms, and data relationships

- ◆ The Grails4Notes<sup>(TM)</sup> Architecture:
- ◆ Designer - web interface to annotate your existing fields/forms to add structure and generate the corresponding Grails domain classes
- ◆ Importer - unidirectional process to load Notes data into an RDBMS through Grails
- ◆ Broker - authentication and business logic layer between web browser, Java<sup>(TM)</sup>, or Apache<sup>(TM)</sup> Flex clients to RDBMS (in progress but optional for your use)
- ◆ Synchronizer - bidirectional process to keep Notes and RDBMS data in sync (on roadmap, but more intricate)

- ◆ Is this vaporware, just an idea, or maybe a pipe dream?
- ◆ This is real & has been in active development since March 2014.
- ◆ We are very far along on the journey ... thousands of development hours invested so far
- ◆ In fact, there are other amazing features we have not discussed yet, including:
  - ◆ Automatic user interface generation with targets for mobile iOS<sup>(TM)</sup> and Android<sup>(TM)</sup> clients plus desktop web browsers
  - ◆ Real time dynamic updating between multiple clients — think of two support reps viewing the same customer seeing one another's updates to the customer's information (requires GraniteDS<sup>(TM)</sup>)

- ◆ What is the current status?
- ◆ It mostly works already (stay tuned for demo & instructional videos)
- ◆ Some DXL import crashes still require debugging
- ◆ Actively refining it for use on internal and some customer NSFs
- ◆ Does not yet support Rich Text Fields or bi-directional synchronization (both features are on the roadmap)
- ◆ Does already support attachments
- ◆ We are looking for more example databases to test — if you want to help
- ◆ Send us your NSF with NO data in it and a description of your field structure, and we will send you back the Entity-Relationship diagram.
- ◆ Designer and Importer will be free for Open Source projects.

- ◆ Please stay tuned for more information:
- ◆ E-mail: Support@Prominic.NET
- ◆ Twitter: @Grails4Notes
- ◆ <http://grails4notes.com>
- ◆ <https://twitter.com/Grails4Notes>
- ◆ <https://www.linkedin.com/groups/Grails4Notes>