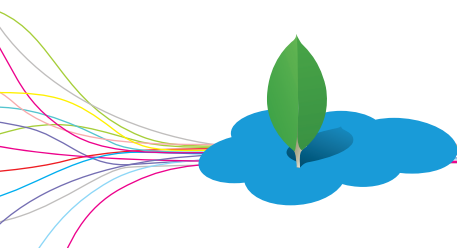




Integrating MongoDB With Salesforce the why and the how



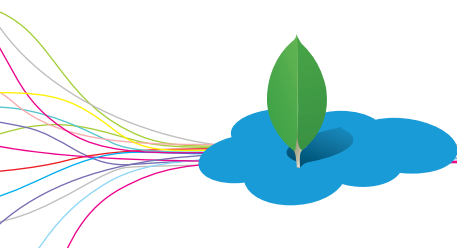
What is NoSQL Databases? No SQL means “Not only SQL.”	2
SalesForce and Database Integration	3
Why Enterprises Need to Integrate NoSQL Databases with SalesForce?	4
SalesForce and NoSQL.....	6
Integration.....	7
The Procedure	9



O1

What is NoSQL Databases? No SQL means “Not only SQL.”

Relational databases such as SQL Server and MySQL work well when the data is in a simple two-dimensional row-column structure, such as in an accounting spreadsheet. However, much of today's data is more complex, with multiple nesting levels and hierarchies. Also, data models are becoming volatile, capable of changing its form and structure frequently, depending on changes in the business environment. NoSQL databases cater to such requirements, catering to the storage and retrieval of data modeled in means other than tabular relations. Such databases facilitate, among other things, simplicity of design, horizontal scaling, and finer control over availability.

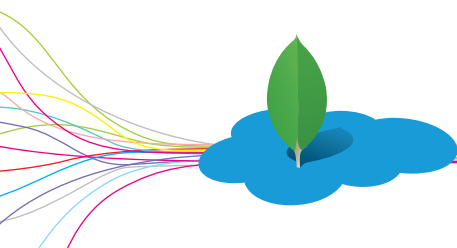


02

SalesForce and Database Integration

SalesForce allows the creation of tables as required and also the import of external tables as per requirements. However, there is no inbuilt option to connect SalesForce to an external database, such as an on-premises database already used by the business. Many businesses require such integration for a variety of reasons, mostly to get real time data into SalesForce.

The two popular options for connecting SalesForce to an external database are developing custom solutions using Data Loader tools and using third party database integration services. Both are easy and reliable options, though not always feasible.



O3

Why Enterprises Need to Integrate NoSQL Databases with Salesforce?

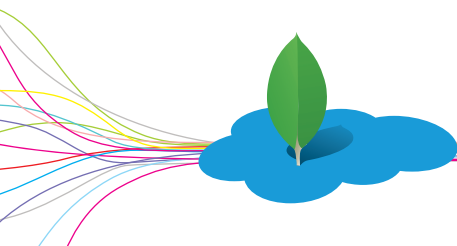
MongoDB is the most popular NoSQL database in vogue. Apart from it being a NoSQL database, its popularity stems from its open source nature, high performance, flexibility, and scalability.

MongoDB stores records in collections of recursive and multi-dimensional objects or documents, rather than as flat, two-dimensional tables. What this means is that the data is stored almost exactly the way it is represented in the program, free of complex mapping or joints that accompany traditional table structures of relational databases. MongoDB is a BSON (Binary-JSON) document database that is capable of holding different “documents” in a single collection, with the number of fields, content and size varying from one document to another. This does away with the need to convert or map application objects to database objects and lends a much simpler structure to the database. For instance, data with complex relationships, such as blogs and user comments are kept together in one object, and message meta-data is stored easily per message or message type without the need to maintain separate collections or schemas. All these make MongoDB the best option for modeling most entities included in web apps.

MongoDB not only has a flexible schema to store heterogeneous collections of content types, but is capable of evolving the structure of the data over time. Unlike a traditional relational database, MongoDB does not require a schema to be set up either, and this schema-less nature facilitates the flexible data model. Unlike a RDBMS database where adding a new column affects the entire database, adding a new field in MongoDB does not affect old rows or render the database unstable. It becomes very easy to declare, extend or alter extra fields to the data model, without coding and scripting.



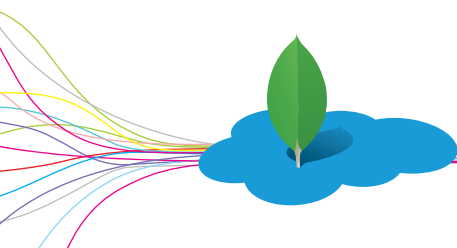
MongoDB also makes it easy to scale out by simply switching to another system. Scale reads by using replica sets and writes through sharding (auto balancing). When new machines are added, the working set gets distributed to it as well. All these make MongoDB perfectly suited for big data, mobile and social infrastructure, content management and delivery, and a host of other applications.



04

SalesForce and NoSQL

Like most serious stakeholders in the tech industry, SalesForce too is veering towards NoSQL. The SalesForce Analytics Cloud, released at the Dreamforce 2014 conference utilizes a Key-value-store based NoSQL database for its actions. Even otherwise, thanks to the growing popularity of NoSQL databases that make many businesses embrace it, the number of NoSQL database integration requests with SalesForce has risen by about 35% over the last one year.



05

Integration

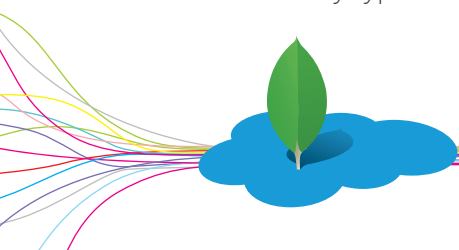
The most popular way to integrate Salesforce with an external database is through the Data Loader tool.

Data Loader is a prebuilt graphical tool that makes it possible to get data into Salesforce objects, extract data from database objects to a desired destination, make bulk deletion using the ID fields of the data, and more. This tool makes a perfect complement to the web-based Import Wizard of Salesforce. The Data Loader comes with many user friendly features such as both a wizard interface and a separate command line interface, a dedicated batch mode interface with database connectivity, ability to support large files ranging millions of rows, support for all inbuilt and custom objects and a built-in Comma Separated Value(CSV) file viewer, making it much more superior to the Import Wizard.

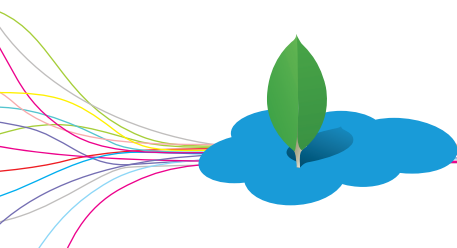
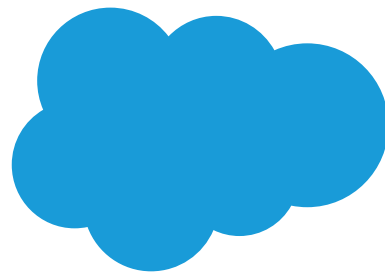
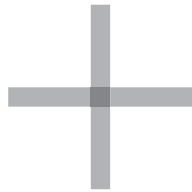
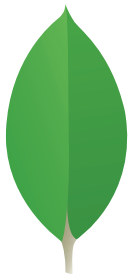
It makes sense to use the Data Loader instead of the Import Wizard when there is a need to automate or schedule data loads instead of a one-time load, when the requirement is to export data for backups on a regular basis, when the data in question is between 50,000 to 5,000,000 records, or when the Import Wizard does not support the object to be loaded. However, the Data Loader tool requires Force.com API, and as such it works only on Enterprise, Unlimited, Developer and Database edition, which allows use of APIs.

One method to integrate a NoSQL database such as MongoDB with Salesforce is to convert the MongoDB database into any traditional database formats, such as CSV files, and then use the Data Loader tool to integrate data from the CSV file to Salesforce.

The Data Loader tool is however not always suited for integrating MongoDB with Salesforce, for it makes automation very difficult. The difficulty comes owing to the fact that MongoDB, though resembling a SQL database, is markedly different, with its own set of languages to handle data requests and an entirely different API to manage the database. Manipulating a normal SQL database entails extracting, adding or changing data in CSV files, SQL files, or even Excel files. Mongo DB however does not use such databases and instead uses much more complex JSON (JavaScript Object Notation) database files, which though more powerful are not yet as popular as SQL. The problem is compounded by the fact that there is no standard NoSQL query language, since there are many types of NoSQL database, each coming with its own USPs and limitations.



It is far easier to develop a single solution that directly pushes MongoDB data to Salesforce and convert the same into an automated solution. What this effectively means is that every NoSQL database solution requires a unique API solution to integrate it with Salesforce. Unlike SQL and CSV database files, NoSQL databases do not have one work-for-all click and play data loaders.



06

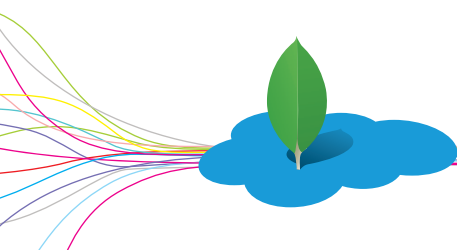
The Procedure

The major work in MongoDB to Salesforce integration is to create custom REST API solutions to access MongoDB JSON files, and pull and push data from Salesforce databases.

The following are the broad steps in the process:

- **Step 1: Authentication:** Authenticate the user to access the MongoDB and Salesforce databases.
- **Step 2: Use MongoDB API for accessing data.** Most RDBMS databases use the JSON database format. MongoDB instead uses BSON, which is a modified form of JSON, and has its own set of APIs to access the BSON data. Accessing the databases will require a custom JavaScript program that can use MongoDB BSON HTTP interfaces to pull and push data “documents” as required. There are no plug-and-play one-size-fits-all options available for this stage.
- **Step 3: Create a custom REST API solution to push that data into Salesforce interface or Salesforce database.** The method would depend largely on the type of database built using MongoDB. For example, when MongoDB is used to store documents, it would require a different REST API solution compared to the one that just pushes data into Salesforce. This is the main reason why Data Loader is not suitable with NoSQL databases. This, however, will change as more and more companies shift to NoSQL database.

There are several dedicated data migration services that facilitate this end, but a common thread around all such integration projects is the indispensability of some coding. There are also several paid data migration solutions that help with the above steps, including coding. Unless the business has competent in-house professionals who are not just technically equipped to handle such tasks, but also able to understand the business requirements of integration in-depth, it may be a good idea to get assistance from such third-party providers.



Suyati Technologies

Suyati is a young, upwardly mobile company focused on delivering niche IT services to support myriad Digital Engagement strategies. Our expertise also includes integration and delivery of CRM, CMS and e-commerce solutions.

www.suyati.com

services@suyati.com

1. <http://www.algoworks.com/blog/how-to-integrate-mongodb-with-SalesForce>
2. <https://zapier.com/zapbook/mongodb/SalesForce/>
3. <http://www.mongodb.com/presentations/social-analytics-mongodb-SalesForce-marketing-cloud-formerly-buddy-media>
4. <http://www.quora.com/Whats-the-best-way-to-integrate-SalesForce-with-mongoDB>
5. <http://www.algoworks.com/blog/how-to-integrate-SalesForce-with-external-database>
6. <http://blog.mongolab.com/2012/08/why-is-mongodb-wildly-popular/>
7. <http://www.zdnet.com/article/rdbms-vs-nosql-how-do-you-pick/>
8. <http://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-nosql-databases/>
9. <http://www.zdnet.com/article/rdbms-vs-nosql-how-do-you-pick/>
10. <http://blog.iprofs.nl/2011/11/25/is-mongodb-a-good-alternative-to-rdbms-databases-like-oracle-and-mysql/>
11. http://www.tutorialspoint.com/mongodb/mongodb_advantages.htm
12. <http://java.dzone.com/articles/when-use-mongodb-rather-mysql>

