Introduction to CouchDB

CouchDB is a relatively new database management system, designed from the ground up to suit modern software applications that tend to be web-based, document-oriented, and distributed in nature. For several decades now, relational database management systems have reigned supreme in application and database development, with the likes of Oracle, SQL Server, and MySQL being used in every type of software application imaginable.

When object-oriented development started to gain traction in the early 1990s, many believed that object-oriented database systems would closely follow suit. Since then, however, there has been a large shift in focus in software development. The breakthrough of dynamic web applications and mobile technology has led to developers looking for lightweight, inexpensive, and well-documented solutions. Many of these developers were prolific at SQL, and open source relational databases such as MySQL simply made the most sense. Today, MySQL is used on millions of web sites across the world.

The relational model that these databases are built on, however, was designed many years ago, when the World Wide Web and Internet were unheard of. Although the strict schema-based structure these databases adhere to is required in some web applications, such as transaction systems, it is not a good fit for many modern web projects, such as blogs, wikis, and discussion forums, which by their nature are a better fit to a document-oriented database.

What Is CouchDB?

CouchDB is a document-oriented database management system, released under the open source Apache License. In contrast to most database systems, it stores data in a schema-free manner. This means that, unlike traditional SQL-based databases, there are no tables and columns, primary and foreign keys, joins, and relationships. Instead, CouchDB stores data in a series of documents and offers a JavaScript-based view model for aggregating and reporting on the data.

If you are wondering where the name CouchDB came from, you may be surprised to hear that it is in fact an acronym. According to the CouchDB wiki, Couch stands for “Cluster Of Unreliable Commodity Hardware,” indicating that CouchDB is intended to run distributed across a cluster of cheap servers. Anyone who has dealt with replication in databases before will know that it is rarely a simple task, but the exact opposite applies when it comes to CouchDB. Add to this the fact that CouchDB is developed in Erlang OTP, a fault-tolerant programming language that offers excellent concurrency features, and you know that your CouchDB database will scale well without a loss of reliability and availability.

Currently, CouchDB is available for most UNIX-based systems, including Linux and Mac OS X. Binary installers are available for Ubuntu, Fedora, CentOS, FreeBSD, and Mac OS X systems through each system’s individual package manager. Windows support is pretty sketchy currently, although an
unofficial binary installer is in the works. Alternatively, CouchDB can be built from source on virtually any POSIX system. I will discuss how to install CouchDB on Linux and Mac OS X in the next two chapters. You will also get a look at an unofficial application for Mac OS X called CouchDBX, which allows you to simply download and run a CouchDB server immediately, no installation or configuration necessary.

**CouchDB: The Story So Far**

In April 2005, Damien Katz posted on his blog about a new database engine he was working on. Details were sparse at this early stage, but what he did share was that it would be a “storage system for a large scale object database” and that it would be called CouchDB. His objectives for the database were for it to become the database of the Internet and that it would be designed from the ground up with web applications in mind.

Katz began working on the database soon after his blog post, choosing C++ as the platform to build it on. Right from the very beginning, CouchDB was designed to be schema-free and indexable, using a combination of append-only storage and atomic updates. It was clear that Katz was heavily influenced by Lotus Notes, the product he worked tirelessly on for many years. The choice of using append-only storage meant that data in a CouchDB database would never be overwritten, but rather it would become “outdated,” with the newer data taking precedence.

In November 2005, Katz announced that he was working on the Fabric formula language. Katz was previously involved in the development of the Lotus Notes Formula language, which Fabric inherited many features from. In December 2005, Katz published a blog post outlining his goals and ambitions for CouchDB, stating that it was “Lotus Notes built from the ground up for the Web.” It was in this blog post that many of the features that exist in CouchDB today were put forward, such as document orientation, distributed architecture, bidirectional replication, and offline access. Further validating the notion that CouchDB would be the “database for the Web” was Katz’s hope that CouchDB would be a great database engine for applications such as e-mail, bug tracking, timesheet management, blogs, and RSS feeds, amongst others.

A big milestone in CouchDB development was the announcement in February 2006 that its underpinning codebase was being moved, in its entirety, from C++ to Erlang. This purpose-built programming language was developed by Ericsson and is heavily used in the telecommunications industry. It is highly centered on the ideas of concurrency control, fault tolerance, and distributed applications, and as a result, Katz believed it was the perfect fit for CouchDB.

Another breakthrough came in April 2006 when it was announced that CouchDB would be solely accessible via an HTTP-based RESTful API. What this means is that rather than connecting to the database server using a client application, you would use any software capable of interacting with an HTTP web server to make requests, which would in turn perform database actions, returning an appropriate response when finished. This means you can manage the database by simply visiting URLs in your web browser, using command-line tools such as curl or, more importantly, via any programming language that supports HTTP requests.

The first publically available release of CouchDB, version 0.2, was made available for download in August 2006. At the time, CouchDB would run only on Microsoft Windows. Not much was said about CouchDB over the next 12 months, but in August 2007, Damien Katz announced that he had decided to scrap XML in the favor of JavaScript Object Notation (JSON) and to get rid of the Fabric formula language altogether, instead choosing to use JavaScript as a query engine. This decision is arguably the most important one made to date in the CouchDB project, and it sparked a huge amount of interest in the project.

In November 2007, version 0.7.0 was released, and it came with a host of new features. CouchDB now featured a JavaScript view engine based on Mozilla Spidermonkey and an attractive web-based