How to Plug into NetBeans in 60 Minutes

Eben Haber (Couchbase)
Geertjan Wielenga (NetBeans)

Why Create Plugins At All?

- Developers like having everything in one place at their finger tips.
- Switching between different applications and command line wastes time.
- Techology companies want to promote their solutions.
- Creating a plugin for a technology brings the solution to developers, enabling them to play with basic features and maybe pay for advanced functionality.

ORACLE Couchbase

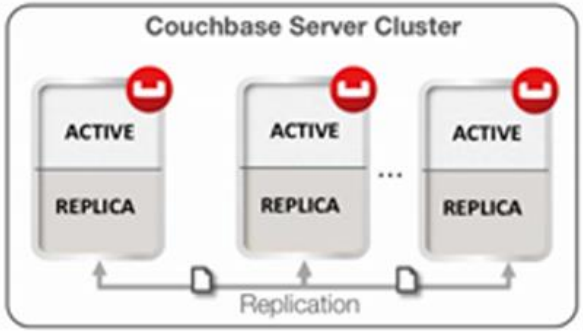
Overview

- Explain the process of creating plugins for NetBeans IDE.
- Example: Plug-in to access Couchbase NoSQL document store.
- We'll go through a complete process from beginning to end.
- You'll be ready to create a plugin of your choice at the end of this session.

1. Couchbase

- Example based on Couchbase
  - Couchbase is an open-source distributed document database
  - In a cluster, it provides high-performance, high availability
  - In any config, it provides storage, querying of JSON documents
  - Free Community Edition available

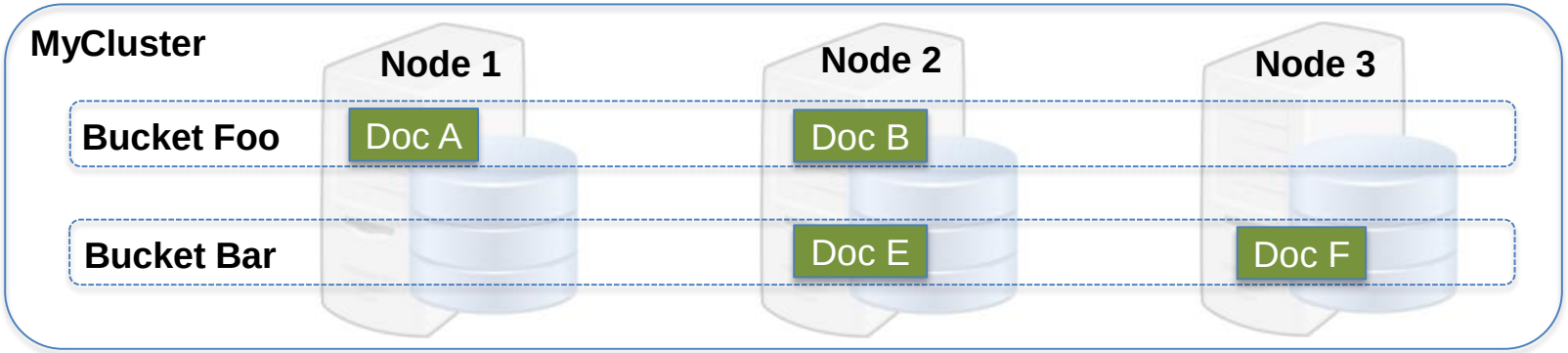
- Why JSON?
  - Lots of data in JSON (e.g., web data, twitter, etc.)
  - Mappings from POJOs to JSON (e.g., Jackson)
  - Structure makes JSON queryable (see N1QL)



ORACLE Couchbase

2. Couchbase Architecture

- Each Couchbase *Cluster* comprised of *Nodes* (i.e., servers) & *Buckets* (i.e., keyspaces)
- Each *Bucket* has *Documents,* sharded across *Nodes* in the *Cluster*
- A *Document* has unique *Key,* and contains JSON or any other data



Using Couchbase

Couchbase is used as a document repository

Put documents in

Retrieve them via their primary key, or…

Retrieve JSON docs via N1QL queries (similar to SQL), e.g.:

```
select * from Tweets where screenname = 'ebenhaber';
```

Only need to worry about Nodes when monitoring, maintenance

E.g., might add new Nodes for scaling out to improve perf, reliability

NetBeans Couchbase Plugin could be used for basic monitoring, querying, or providing data services to other NetBeans plugins

ORACLE Couchbase

3. Create Levels of Requirements for Plugin

- **Prototype:** Hack together a few basic scenarios.
- **Minimal:** Create an absolutely bare-minimum set of requirements.
- **Medium:** Provide the next level of requirements.
- **Full:** What would “full” support provide?
- **Advanced:** Advanced and sophisticated features.

4. Identify Relevant Extension Points in NetBeans

- It helps if you already know NetBeans a bit.
- Explore the latest release of NetBeans and find similar features.
- For example, Couchbase is a database technology,
  so find how other database technologies are supported.

5. Understand NetBeans Extension Mechanism

- Virtual FileSystem, extended via XML files.
- Couchbase will be visualized in a hierarchical structure, therefore need to find NetBeans API to support this.
- NetBeans Nodes API provides hierarchical infrastructure.
- NetBeans Actions API provides loosely coupled action infrastructure.

6. Set Up a NetBeans Plugin Project

- NetBeans modules are defined in Ant or Maven.
- Demo: Setting up a NetBeans Maven module project.
- Add a dependency on Couchbase API.
- Check into GitHub repo.

7. Create a Simple “Hello World” Scenario

- Hook a root node for Couchbase
  into the correct place within NetBeans IDE.
- Demo:
  - Create a new layer.xml file in the project.
  - Hook it into a new root node.
  - Run it and show the node in the applicable place.

ORACLE Couchbase

8. Prototype and Release Minimal Requirements

- Sketch out scenarios for Couchbase users of NetBeans.
- Create a package structure that is easy to maintain.
- Create a node hierarchy for the Couchbase terminology.
- Add actions, but as few as possible for the minimal requirements.
- Keep it as simple as possible.

9. Get Feedback and Engage the Community

- Interact with community.
- Publish on plugins.netbeans.org.
- Write articles and blogs.
- Use Twitter and Facebook.

10. Implement the Next Level of Requirements

- Be careful to not get feature creep.
- Stay focused on the requirements for each stage.
- Whenever a new stage is complete, release a new version of the plugin.
- Try assign tasks to new members involved in the project.
- Keep going at a steady pace.

Conclusion

- Creating a NetBeans plugin is a process.
- Keep focused on specific and well-defined aims.
- Helps to be knowledgeable of the technology you want to support.
- Lots of NetBeans API documentation and helpful community.
- Creating NetBeans plugins is fun!