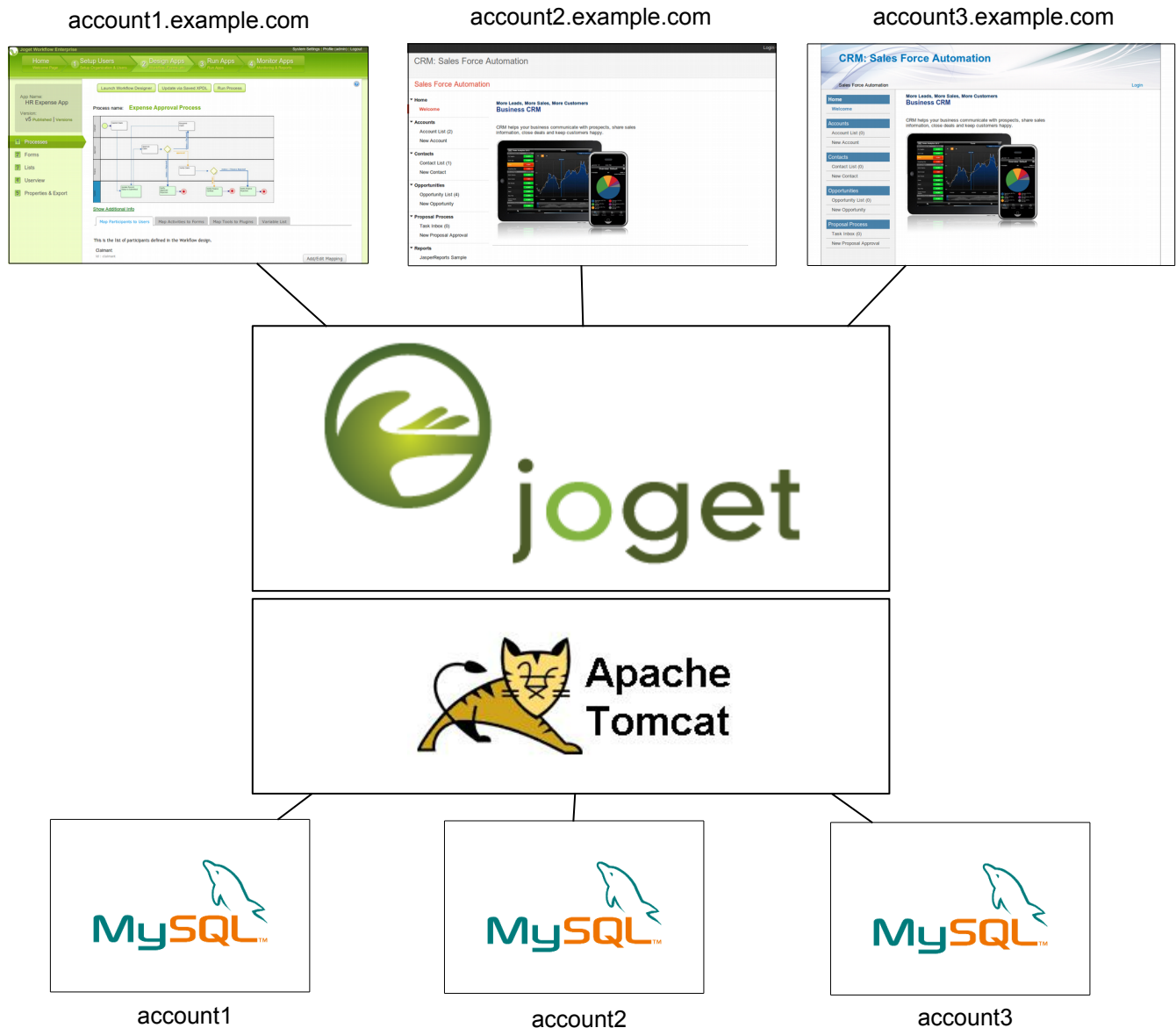


Joget Workflow Cloud Edition

Overview

Joget Workflow Cloud Edition is an enhanced version of Joget Workflow Enterprise that supports multi-tenancy, allowing for multiple accounts (tenants) to be hosted on one webapp. Each account appears to be a separate installation.

The architecture is based on a shared webapp connecting to separate databases or schemas. This allows for more cost effective delivery of services, since many accounts can be hosted on a single app server. At the same time, each account has its own separate database and file upload directory to preserve data isolation.



Installation

Prerequisites

- Linux (The current bundle is only packaged for Linux)
- Java 7 and above
- MySQL 5 and above

Installation Steps

The installation steps are similar to the installation for the standard Joget Workflow. Refer to the Knowledge Base for more detailed instructions at <http://dev.joget.org/community/display/KBv5/Installation+on+Linux>.

The basic steps are:

1. Create a new directory e.g. /opt/joget and extract the tar.gz bundle into that directory
2. Install the Java Runtime Environment (JRE) or Java Development Kit (JDK) version 7 and above
e.g. in Ubuntu: `sudo apt-get install openjdk-7-jdk`
3. Install MySQL Server version 5 and above
e.g. in Ubuntu: `sudo apt-get install mysql-server`
4. Create an empty database 'jwdb' in the MySQL server
5. Execute the setup script to create the required database tables: `./setup.sh`

Alternatively, manually create the database tables as follows:

- 5.1 Populate the jwdb database with the SQL script in data/jwdb-empty.sql
- 5.2 Edit wflow/app_datasource-default.properties to match your database settings
6. Execute the bundled Apache Tomcat application server: `./cloud-tomcat.sh run`
7. Access the home page `http://localhost:8080`
8. Access the default Workflow Management Console at `http://localhost:8080/jw`

Configuration

DNS Configuration

Each account / tenant is identified by a domain name, so the DNS needs to be configured to point desired domains to Joget. For convenience, it is recommended to use a wildcard DNS record (http://en.wikipedia.org/wiki/Wildcard_DNS_record). For example:

```
*.example.com
```

In this case requests to all subdomains of example.com (e.g. account1.example.com, account2.example.com, account3.example.com) are directed to Joget.

For local testing purposes, it is possible to just modify the local hosts file [http://en.wikipedia.org/wiki/Hosts_\(file\)](http://en.wikipedia.org/wiki/Hosts_(file))

Profile Configuration

Each account (tenant) is tied to a datasource profile.

In Community or Enterprise Edition, only 1 profile is selected at a time. In the Cloud Edition, this is enhanced such that all accounts and profiles are enabled concurrently.

Open the file **{joget.home}/wflow/app_datasource.properties**. An example:

```
account1.example.com=jwc_account1
account2.example.com=jwc_account2
currentProfile=default
```

- The key to the left of the equals is the domain name, while the value to the right is the profile.
- Each profile has a corresponding file **app_datasource-{profile}.properties** which contains the database settings.
- When a request is received, any matching domain name will use the associated profile.
- **currentProfile** is a special key to indicate the default profile to use if there are no matching domains.
- In terms of file uploads and system generated files, each profile stores its files in the subdirectory **{joget.home}/wflow/app_profiles/{profile}**

Account Provisioning

There are a few approaches to provision (create) an account.

Manual Provisioning

As described in the previous Accounts Configuration section, each account is essentially a datasource profile. So an account can be manually created as follows:

1. Create a new Joget Workflow database using the script provided in `{joget.home}/data/jwdb-cloud.sql`, or any other custom scripts
2. Create a new profile file `{joget.home}/wflow/app_datasource-{profile}.properties` by copying from `app_datasource-default.properties` and changing the values accordingly.
3. Add an entry into `{joget.home}/wflow/app_datasource.properties`. The entry would be `{domain name}={profile}`
4. The profile is active immediately, and no restart is necessary.

Scripted Provisioning

There is also an Ant build file and script available that simplifies account creation.

1. Execute the file `{joget.home}/cloud-provision-account.sh`
2. At the prompt for Account Profile Name, enter the required account name e.g. demo
3. The next prompt is for the Main Domain Name, enter the main domain e.g. example.com
4. Subsequent inputs will have default values that you can just Enter and accept, or change before submitting. The inputs are as follows:

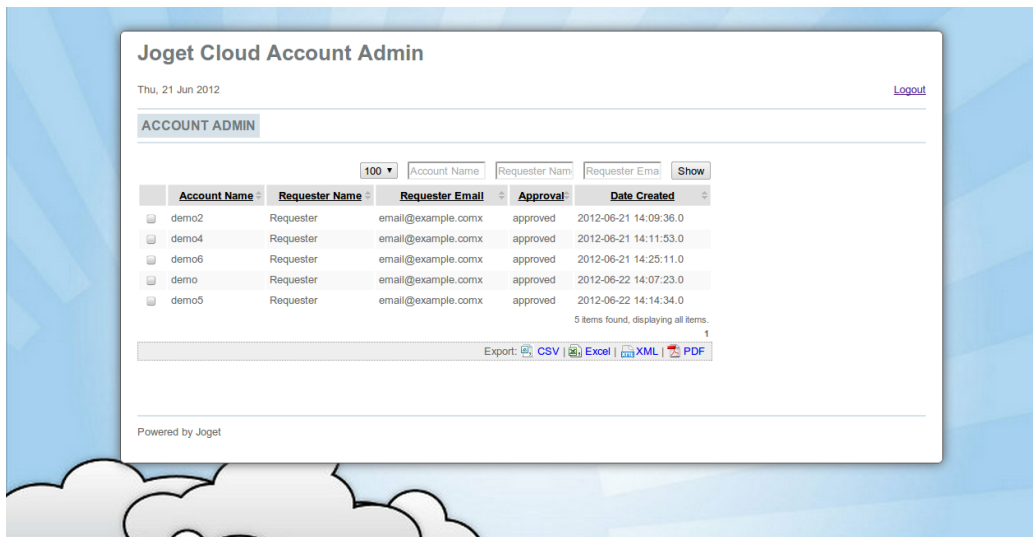
Account Profile Name	Desired account name e.g. demo
Main Domain	Main domain e.g. example.com
Account Domain Name	Account domain e.g. demo.example.com
Account Database Name	Database name for the account e.g. jwc_demo
Account Database User	Database user for the account e.g. jwc_demo
Account Database Password	Database password for the account e.g. jwc_demo
Home Directory	Joget home directory, defaults to current directory .
wflow Directory	Joget configuration directory, defaults to ./wflow
MySQL host name	Database host name
MySQL port	Database port
MySQL username	Database admin user (with permission to create database)
MySQL password	Database admin password

5. Upon submission of all the inputs, the script will create the database and perform the necessary configuration for the account.

Cloud Admin App

For convenience, a simple Joget app is provided to automate the account request approval and provisioning process.

1. Import the app into Joget
2. Go to Properties & Export and set the “domain” to the appropriate value for the main domain e.g. example.com
3. Publish the app
4. Access the app at <http://localhost:8080/jw/web/userview/cloud/admin/>
5. Login to view the Account List



6. Select New Account from the drop-down menu to request for a new account.

The screenshot shows the 'Joget Cloud Account Admin' web application with the 'New Account' form open. The form includes a dropdown menu for 'ACCOUNT ADMIN' with options 'Account List (5)', 'New Account', and 'Inbox (0)'. Below this, there's a text input for 'Account Name' and a note: 'Please key in your desired account name. If the account is available, your domain will be [Account Name].cloud.joget.com'. The 'Requester Details' section has fields for 'Requester Name', 'Requester Email', and 'Request Comments'. At the bottom, there's a checkbox for 'Terms and Conditions' with the text 'I accept the terms and conditions' and a 'Submit' button. The footer indicates 'Powered by Joget'.

7. When a request is submitted, it needs to be approved. Approve the request, and a BeanShell tool will be executed to provision the account.
8. Verify that the account is created at the requested domain e.g. demo.example.com
9. Modify the app where necessary:
 - Go to Processes > Map Participants to Users to map the participants accordingly
 - Go to Processes > Map Tools to Plugins, and configure the Email Tools to key in the appropriate email settings and email messages
 - Edit Forms > Account Request Form to modify the form contents where necessary

Account Removal

Manual Removal

To disable an account, perform the following:

- Remove the profile entry from {joget.home}/wflow/app_datasource.properties. This effectively disables the account.

To fully delete the account:

- Remove the profile entry from {joget.home}/wflow/app_datasource.properties.
- Delete the profile properties file {joget.home}/wflow/app_datasource-{profile}.properties
- Delete the associated account database
- Delete the associated account profile directory {joget.home}/wflow/app_profiles/{profile}
- If the Cloud Admin App is used, delete the record from the db table app_fd_cloud_account

Maintenance

Logging

As with a standard Joget installation on Apache Tomcat, the log file is at {joget.home}/apache-tomcat-{tomcat.version}/logs/catalina.out

Updates

Updates are to WAR files are done as per a standard Joget Workflow installation.

Backup

Backups should be done on all the MySQL databases, as well as the configuration and data files in {joget.home}/wflow