



# Joget DX

# Basic System Administration

 <http://facebook.com/jogetworkflow>

 <http://twitter.com/jogetworkflow>

# Prerequisites

---

1. Basic knowledge on how to create Joget App.
2. Proficient in server administration (i.e. Application server, Database Server, Networking).

# Content

---

1. Typical stack for Joget
2. Basic Database Management (MariaDB)
3. Basic Application Server Management (Tomcat)
4. Web Log Viewer
5. Application Performance Monitoring (APM)



# Chapter 1

## Typical Stack for Joget

# Typical Platform Requirement

Apache Tomcat



MariaDB



MariaDB

JDK





### Apps

Progressive Web Apps  
Various Form Factors



Customer Relationship



Employee Services Portal



Expenses Claims App



Internal Service Request App



### Development

Simple, Visual,  
Drag and Drop.

Form Builder	List Builder	Userview Builder	Process Builder

**User Management**

- Organizations
- Groups
- Users

**Monitoring**

- Processes
- Application Performance
- Logging & Audit

**System**

- Workflow Engine
- Security and Directory
- Plugin Management

**Plugin Architecture**

**Add-on Builders**

**Integration API**

- API JSON
- API Java
- API JavaScript



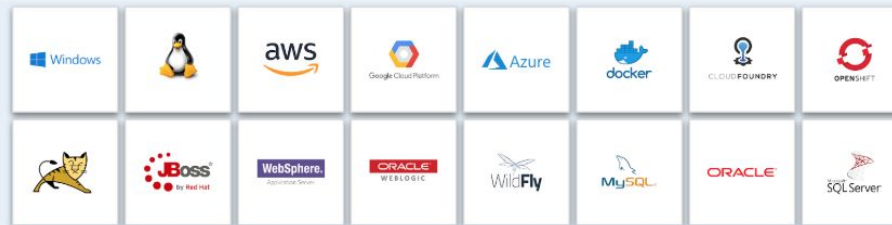
### Platform

Open, Flexible and  
Extensible.



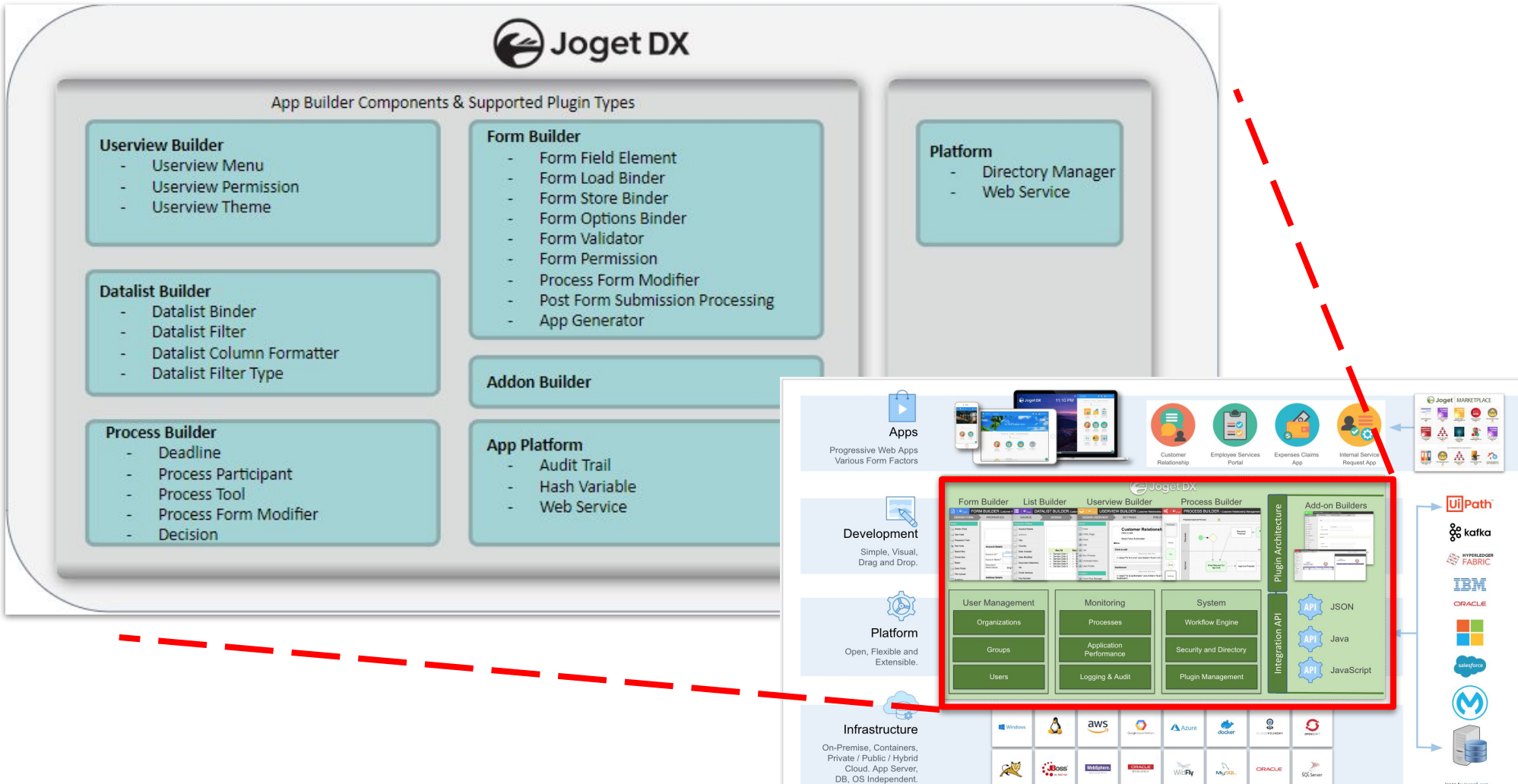
### Infrastructure

On-Premise, Containers,  
Private / Public / Hybrid  
Cloud. App Server,  
DB, OS Independent.

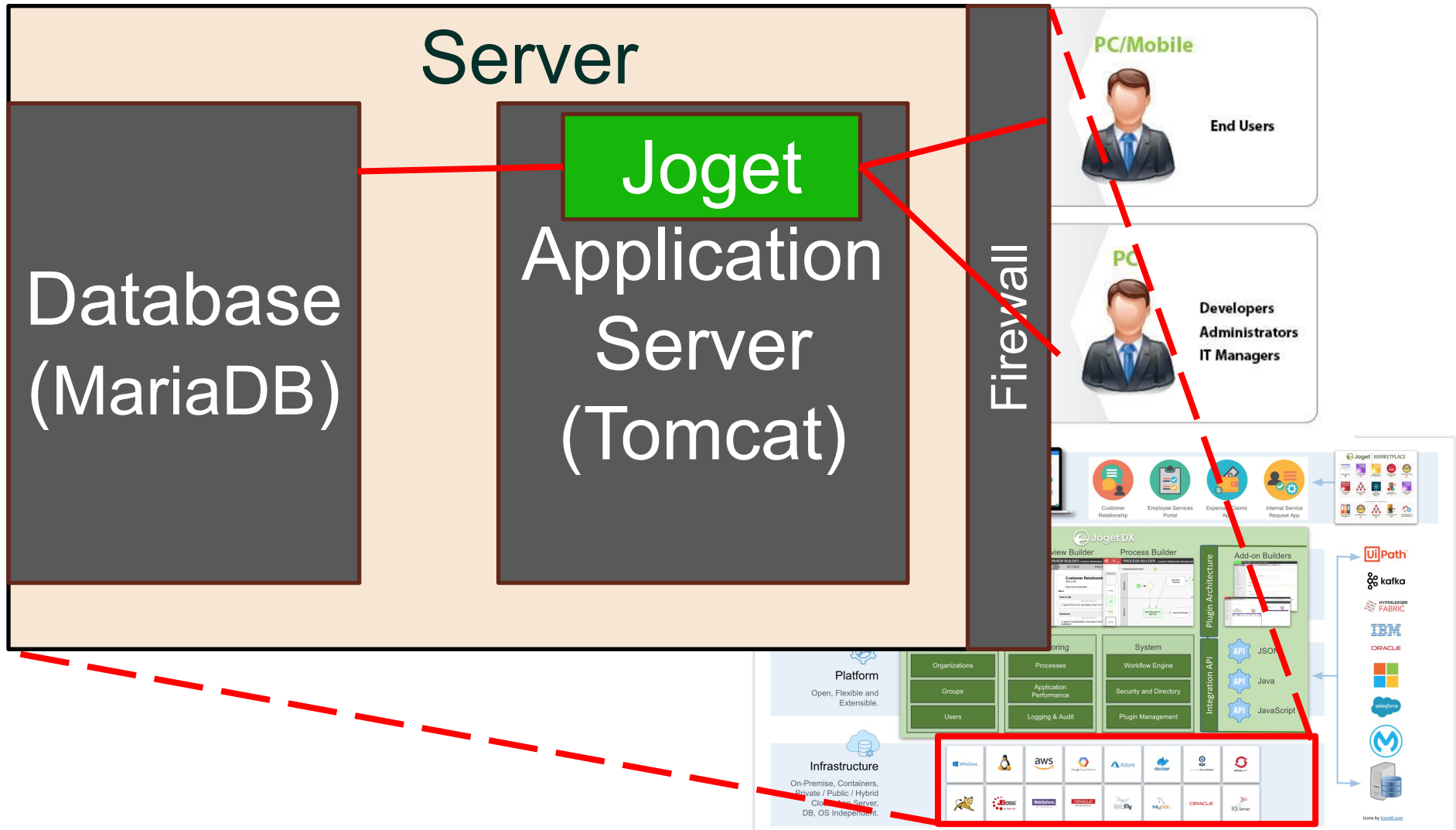


Icons by [icons8.com](https://icons8.com)

# Plugin Types



# The Typical Stack





# Chapter Review

---

- General understanding on how Joget is hosted.

---

# Chapter 2

## Basic Database Management (MariaDB)

# Typical Platform Requirement

Apache Tomcat

MariaDB

JDK



We are going to inspect the Datasource

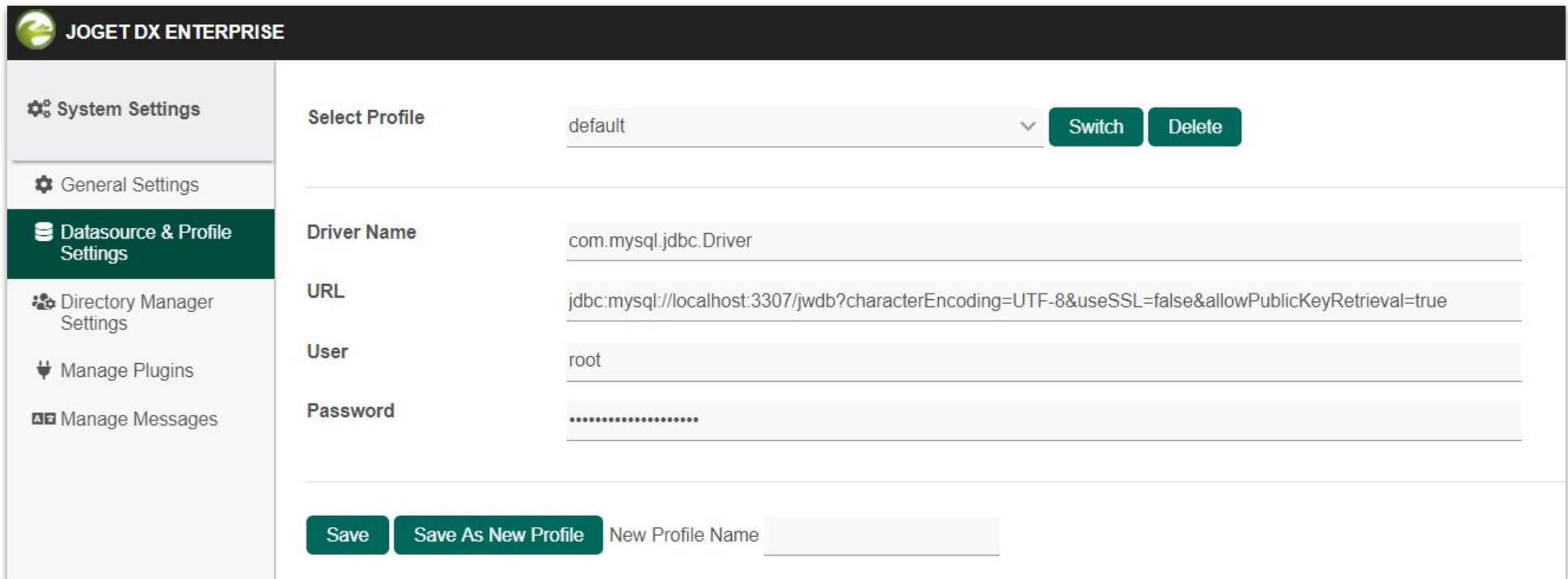
# Datasource Profile

---

- Joget supports configuration of \***multiple** datasource profiles, but only **1** profile can be activated at any point of time.

# Datasource Profile Configuration

- The Datasource Profile can be located in **Settings -> Datasource & Profile.**



The screenshot shows the 'Datasource & Profile Settings' configuration page in the Joget DX Enterprise application. The interface includes a sidebar with navigation options and a main configuration area with several input fields and buttons.

**Navigation Sidebar:**

- System Settings
- General Settings
- Datasource & Profile Settings**
- Directory Manager Settings
- Manage Plugins
- Manage Messages

**Main Configuration Area:**

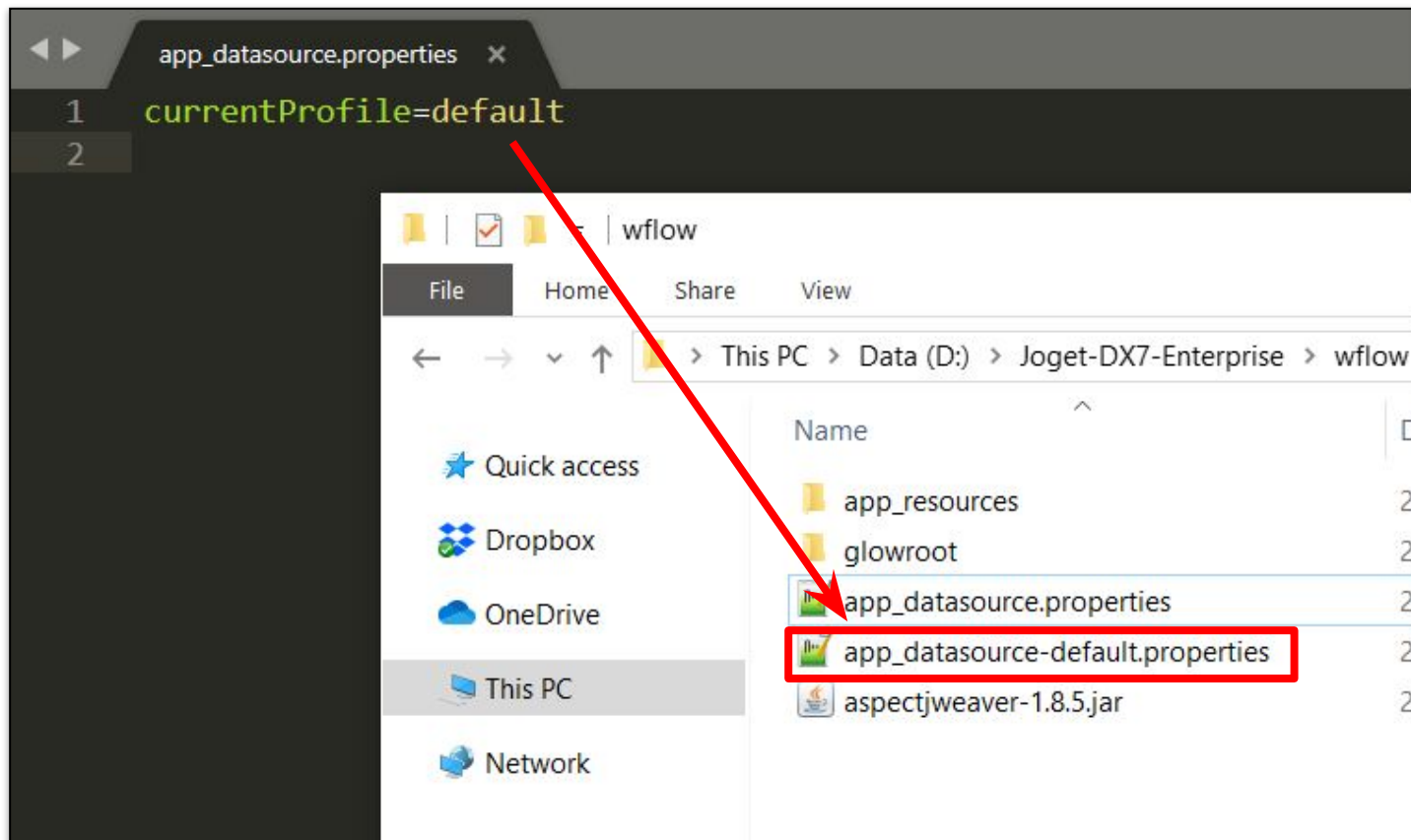
- Select Profile:** A dropdown menu showing 'default' with a downward arrow, accompanied by 'Switch' and 'Delete' buttons.
- Driver Name:** A text input field containing 'com.mysql.jdbc.Driver'.
- URL:** A text input field containing 'jdbc:mysql://localhost:3307/jwdb?characterEncoding=UTF-8&useSSL=false&allowPublicKeyRetrieval=true'.
- User:** A text input field containing 'root'.
- Password:** A password input field with masked characters (dots).

**Bottom Action Bar:**

- 'Save' button
- 'Save As New Profile' button
- 'New Profile Name' text input field

# Datasource Profile Configuration in File System

- The Datasource Profile can also be located in the **wflow** folder.



# Optional Exercise – Setting Up New Database

---

- Assuming that our current installation of Joget is connected to the MariaDB database named “jwdb”
- We are going to:
  1. Create a new database named “wflowdb”
  2. Create a new datasource profile that uses “wflowdb”
  3. Switch Joget’s datasource profile to use “wflowdb”
- Joget DX will automatically initialize any new empty databases upon setup (no datasource profile found).

# 1. Setup Empty Wflowdb

- Assuming MariaDB is installed in **C:\joget\mariadb**.
- Run command prompt.

```
cd C:\Joget\mariadb\bin
mysql -u root
mysql> create database wflowdb;
Query OK, 1 row affected (0.05 sec)
mysql> exit;
Bye
mysql -u root wflowdb < C:\Joget\data\jwdb-empty.sql
```

Import to this database

Location of the SQL import file



## 2. Verify Creation of Wflowdb

---

- Verify existence of new database.

```
mysql -u root
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| jwdb |
| wflowdb |
| performance_schema |
+-----+
5 rows in set (0.00 sec)
mysql> exit;
Bye
```

## 3. Create A New Datasource Profile

---

- Login to **Joget** as Admin user.
- Navigate to **System Settings > Datasource & Profile Setting**.
- Amend the URL field and change “jwdb” to “wflowdb” (without quotation marks).
- Specify a new profile name – “wflow”.
- Click on the “Save As New Profile” button.

# 3. Create A New Datasource Profile

**JOGET DX ENTERPRISE**

- System Settings
- General Settings
- Datasource & Profile Settings**
- Directory Manager Settings
- Manage Plugins
- Manage Messages

Select Profile: default

Driver Name: com.mysql.jdbc.Driver

**URL: jdbc:mysql://localhost:3307/wflowdb?characterEncoding=UTF-8&useSSL=false&allowPublicKeyRetrieval=true**

User: root

Password: .....

## 4. Compare The Changes in File System

---

- The value of **currentProfile** in `/wflow/app_datasource.properties` is changed to **wflow**.
- A **new file**, `/wflow/app_datasource-wflow.properties` is created, with configurations to **wflow** datasource.

# Database User / Password Changed?

---

- Open to edit

`/wflow/app_datasource-profileName.properties` file in text editor.

- To update database username, amend the values for **workflowUser**.
- To update database password, amend the values of **workflowPassword**.
- Restart Joget server (or Apache Tomcat) for changes to take effect.

# Discussion

---

- Can we use other Application Server container other than Tomcat? (e.g: JBoss, Glassfish, etc...)
- Can we use other Database Systems other than MariaDB?
- Must we use RDBMS?
- How does Joget talk to MariaDB?
- How do users access Joget?
- Can we separate Application Server and Database Server?

# Chapter Review

---

- General understanding on how Joget connects to the Database system.

---

# Chapter 3

## Basic Application Server Management (Tomcat)



# Typical Platform Requirement

Apache Tomcat

MariaDB

JDK



MariaDB



We are going to inspect Tomcat

# Joget Application Files

---

- In Apache Tomcat, there are 1 Joget web application file to be noted:
  - {Tomcat}/webapps/**jw**.war

# Updating Joget

- Essentially, just replace the **.war** files with the latest available version.
- You can update your Joget by following these steps:
  1. **Stop** Joget.
  2. **Delete** "**jwt**" folder and "**jwt.war**" in "[Joget Installation Directory]/[tomcat]/webapps".
  3. **Delete** "**jwt**" folder in "[Joget Installation Directory]/[tomcat]/work/Catalina/localhost".
  4. **Place** your newly downloaded "**jwt.war**" in "[Joget Installation Directory]/[tomcat]/webapps".
  5. **Start** Joget.

# wflow.home Directory

---

- The location where the following Joget physical files are stored:
  - Datasource and profile configurations.
  - XML definition and HTML rendering of forms designed using Form Builder.
  - Plugins.
  - Graphical image and thumbnail of all processes.
  - Uploaded Files Attachments.

# Where is wflow.home Directory?

- Open **joget-start.bat** using text editor, and look for the **JAVA\_OPTS** parameters.
- wflow.home is configured using the **-Dwflow.home** option.
- default **wflow** folder location is in the root of Joget folder.

```
9 REM Start Tomcat
10 set JAVA_HOME=.\jre11.0.2
11 set CATALINA_HOME=.\apache-tomcat-8.5.41
12 set JAVA_OPTS=-Xmx768M -Dwflow.home=./wflow/ -javaagent:./wflow/aspectjweaver-1.8.5.jar
13 REM set JAVA_OPTS=-XX:MaxPermSize=128m -Xmx1024M -Xdebug -Xnoagent -Djava.compiler=NONE
  -javaagent:./wflow/aspectjweaver-1.8.5.jar -javaagent:./wflow/glowroot/glowroot.jar
14 ECHO == Starting Tomcat from %CATALINA_HOME% ==
15 ECHO.
16 %CATALINA_HOME%\bin\catalina.bat run
```

# Backup

---

- To backup Joget, you should include:
  - Use database dump as a snapshot:
    - `mysqldump -u root wflowdb > backup.sql`
  - Backup entire `wflow.home` directory.

# Log Files

---

- When Joget is running in console, all log messages are shown on the console window.
- Log messages from **Apache Tomcat** are written to `{Tomcat}/logs/catalina.yyyy-MM-dd.log`.
- Log messages from **Joget** are written to `{Tomcat}/logs/localhost.yyyy-MM-dd.log`.
- To customize the logging behaviors, edit `{Tomcat}/conf/logging.properties` file.

# What is a Stack Trace?

```
Start Joget Server
ERROR 02 Sep 2010 10:19:51 org.hibernate.util.JDBCExceptionReporter - Cannot create PoolableConnectionFactory (Access denied for user 'root'@'localhost')
ERROR 02 Sep 2010 10:19:51 org.springframework.web.context.ContextLoader - Context initialization failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'setupSessionFactory' defined in class path resource [commonsAp
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1336)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:471)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.run(AbstractAutowireCapableBeanFactory.java:409)
    at java.security.AccessController.doPrivileged(Native Method)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:380)
    at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:264)
    at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:220)
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:261)
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:185)
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:164)
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:423)
    at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:729)
    at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:381)
    at org.springframework.web.context.ContextLoader.createWebApplicationContext(ContextLoader.java:255)
    at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:199)
    at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:45)
    at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:3843)
    at org.apache.catalina.core.StandardContext.start(StandardContext.java:4342)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:791)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:771)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:525)
    at org.apache.catalina.startup.HostConfig.deployDescriptor(HostConfig.java:627)
    at org.apache.catalina.startup.HostConfig.deployDescriptors(HostConfig.java:553)
    at org.apache.catalina.startup.HostConfig.deployApps(HostConfig.java:488)
    at org.apache.catalina.startup.HostConfig.start(HostConfig.java:1149)
    at org.apache.catalina.startup.HostConfig.lifecycleEvent(HostConfig.java:311)
    at org.apache.catalina.util.LifecycleSupport.fireLifecycleEvent(LifecycleSupport.java:117)
    at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1053)
    at org.apache.catalina.core.StandardHost.start(StandardHost.java:719)
    at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1045)
    at org.apache.catalina.core.StandardEngine.start(StandardEngine.java:443)
```



# What is a Stack Trace?

---

- When an error is thrown, a stack trace will depict a sequence of events executed in the code level, which can precisely suggest the point where an exception is caught.
- Stack trace could also suggest meaningful error message to help troubleshooting.
- In the example above, indicative error message is shown before stack trace:

```
ERROR 02 Sep 2019 10:19:51 org.hibernate.util.JDBCExceptionReporter -  
Cannot create PoolableConnectionFactory (Access denied for user  
'roo'@'localhost' (using password: NO))
```

# Troubleshooting

---

- When you're seeking for help on troubleshooting, copy the whole stack trace (**all log messages printed at the same date time**) and share it out. This can help the troubleshooter to have a better idea on the error.

# Memory Allocation

- Open **joget-start.bat** using text editor, and look for the **JAVA\_OPTS** parameters.
- Memory allocation is configured using the **-Xmx** option.

```
9 REM Start Tomcat
10 set JAVA_HOME=.\jre11.0.2
11 set CATALINA_HOME=.\apache-tomcat-8.5.41
12 set JAVA_OPTS=-Xmx768M -Dwflow.home=./wflow/ -javaagent:./wflow/aspectjweaver-1.8.5.jar
13 REM set JAVA_OPTS=-XX:MaxPermSize=128m -Xmx1024M -Xdebug -Xnoagent -Djava.compiler=NONE
   -javaagent:./wflow/aspectjweaver-1.8.5.jar -javaagent:./wflow/glowroot/glowroot.jar
14 ECHO == Starting Tomcat from %CATALINA_HOME% ==
15 ECHO.
16 %CATALINA_HOME%\bin\catalina.bat run
```

# Optimize Tomcat

---

- Edit {Tomcat}/conf/server.xml using text editor.
- Look for the **HTTP/1.1** connector configuration:
- Try to add MaxThreads="" to set your preferred maximum thread count.

```
<Connector port="8080" protocol="HTTP/1.1"  
          connectionTimeout="20000"  
          maxThreads="2000"  
          compression="on"  
          useSendfile="false"  
          redirectPort="8443" />
```

- NOTE: One size does not fits all. Every environment need to be fine tuned accordingly. Read more at

<https://dev.joget.org/community/display/DX7/Joget+Clustering+and+Performance+Testing+on+AWS>

# Changing Apache Tomcat HTTP Port

- Edit {Tomcat}/conf/server.xml using text editor.
- Look for the **HTTP/1.1** connector configuration:

```
<Connector port="8080" protocol="HTTP/1.1"
```

- Change the port number to your preferred one, and restart Joget Server (Apache Tomcat).

# Optional Exercise – Setting up SSL

---

- Assuming that you are running Joget with the default bundled Tomcat.
- We are going to:-
  1. Generate a key store file.
  2. Configure Tomcat to support SSL.

Reference:

<https://dev.joget.org/community/display/DX7/Setting+Up+SSL+on+Tomcat>

# 1. Generate a key store file

---

- First of all, we will need to generate a key store file. You may want to generate it with or without a SSL certificate purchased from your SSL certificate provider. This is an example on generating one by ourselves.

# 1. Generate a key store file

```
C:\Program Files\Java\jdk\bin>keytool -genkey -alias tomcat -keyalg RSA
Enter keystore password: password
Re-enter new password: password
What is your first and last name?
  [Unknown]: Robert
What is the name of your organizational unit?
  [Unknown]: home
What is the name of your organization?
  [Unknown]: home
What is the name of your City or Locality?
  [Unknown]: SF
What is the name of your State or Province?
  [Unknown]: CA
What is the two-letter country code for this unit?
  [Unknown]: US
Is CN=Robert, OU=home, O=home, L=SF, ST=CA, C=US correct?
  [no]: yes
Enter key password for <tomcat>
      (RETURN if same as keystore password): password
Re-enter new password: password
```



## 2. Configure Tomcat to support SSL

---

- Make sure that your server is not running. Open up `\apache-tomcat\conf\server.xml`, uncomment and edit the following lines accordingly.

## 2. Configure Tomcat to support SSL

- **Port:** 8443 to 443 (If you intend to browse to <https://yourDomain> instead of <https://yourDomain:8443>)  
**keystoreFile:** Path to the .keystore file  
**keystorePass:** The password defined earlier

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
      This connector uses the JSSE configuration, when using APR,
the
      connector should be using the OpenSSL style configuration
described in the APR documentation -->

<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
          maxThreads="150" scheme="https" secure="true"
          clientAuth="false" sslProtocol="TLS"
          keystoreFile="C:/Users/Robert/.keystore"
          keystorePass="password" />
```

## 2. Configure Tomcat to support SSL

---

- Start your server.
- You may now surf to your Joget at <https://yourDomain/jw> or <https://yourDomain:8443/jw>, depending on your configuration.

# Chapter Review

---

- General understanding on how Joget is “hosted” runs under application server – Tomcat.



# Chapter 4

## Web Log Viewer

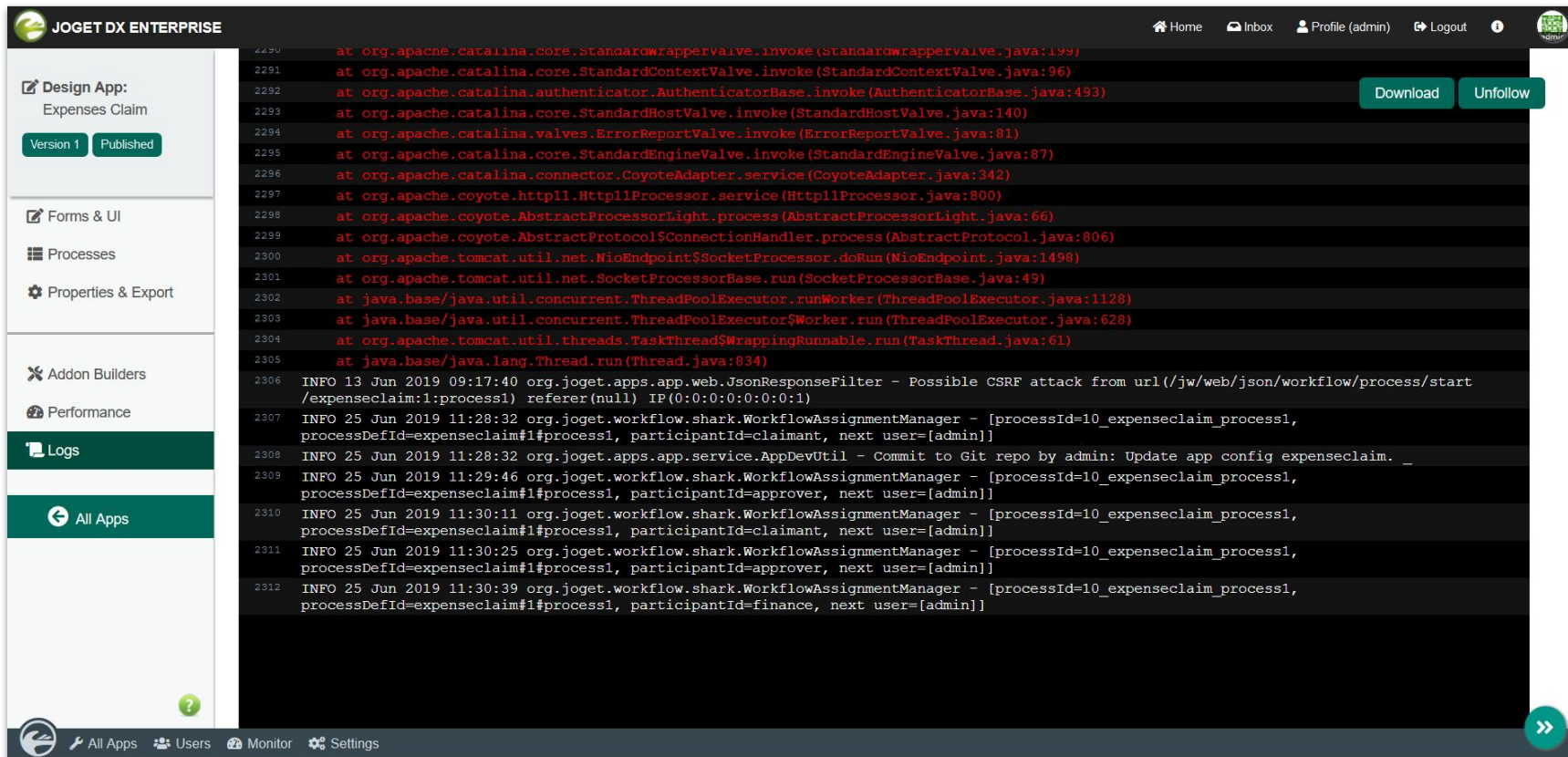
# Accessing Log using Web Log Viewer

---

- New DX Feature.
- Web App Log Viewer enables the administrators to view the logs on the web console for viewing and finding errors.

# App specific Log

- Displays log information and errors related to the specific App.



**JOGET DX ENTERPRISE**

Home | Inbox | Profile (admin) | Logout

**Design App:**  
Expenses Claim  
Version 1 | Published

Forms & UI  
Processes  
Properties & Export  
Addon Builders  
Performance  
**Logs**

All Apps

Download | Unfollow

```

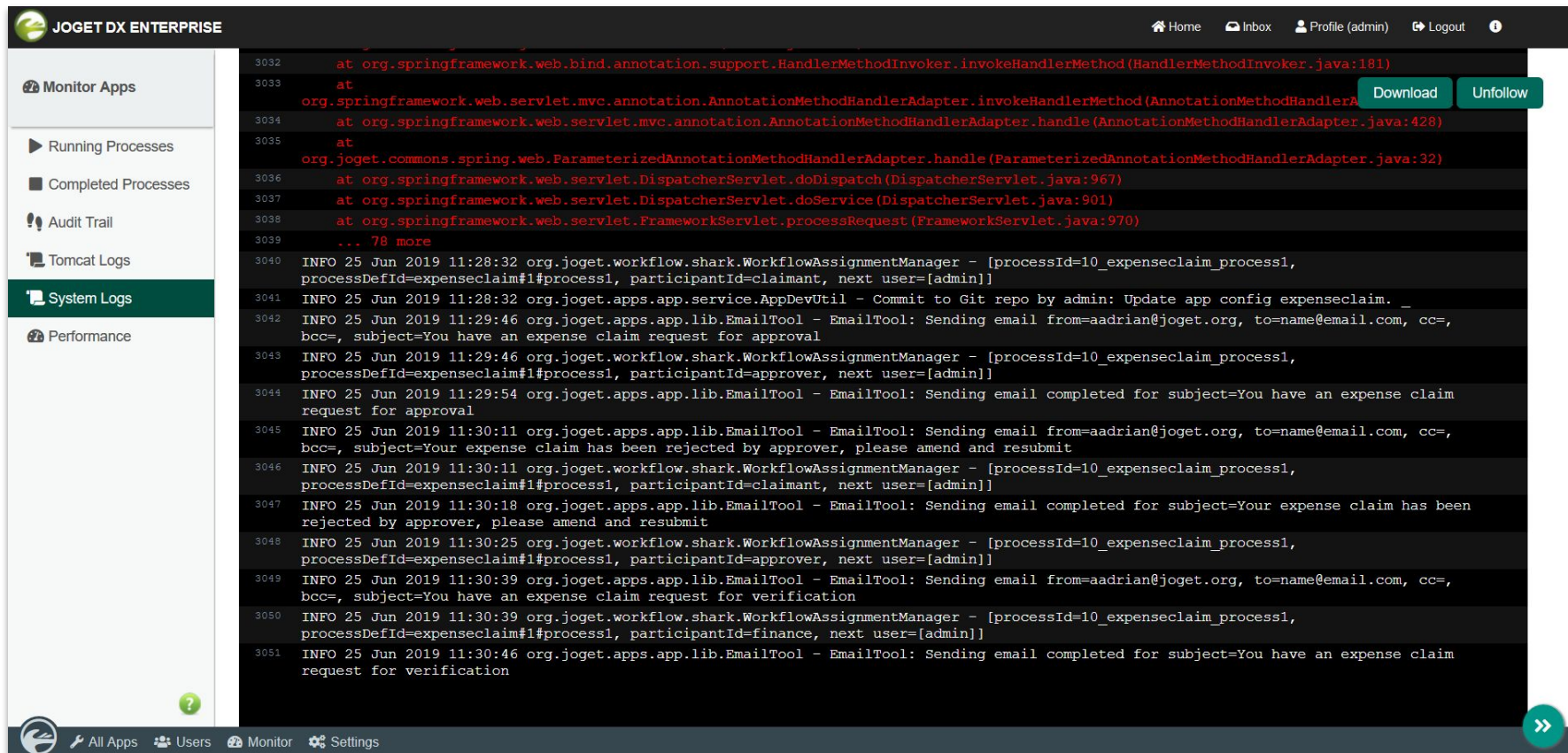
2290 at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
2291 at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
2292 at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:493)
2293 at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:140)
2294 at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81)
2295 at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
2296 at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:342)
2297 at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:800)
2298 at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66)
2299 at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:806)
2300 at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1498)
2301 at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
2302 at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128)
2303 at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
2304 at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
2305 at java.base/java.lang.Thread.run(Thread.java:834)
2306 INFO 13 Jun 2019 09:17:40 org.joget.apps.app.web.JsonResponseFilter - Possible CSRF attack from url(/jw/web/json/workflow/process/start
/expenseclaim:1:process1) referer(null) IP(0:0:0:0:0:0:1)
2307 INFO 25 Jun 2019 11:28:32 org.joget.workflow.shark.WorkflowAssignmentManager - [processId=10_expenseclaim_process1,
processDefId=expenseclaim#1#process1, participantId=claimant, next user=[admin]]
2308 INFO 25 Jun 2019 11:28:32 org.joget.apps.app.service.AppDevUtil - Commit to Git repo by admin: Update app config expenseclaim.
2309 INFO 25 Jun 2019 11:29:46 org.joget.workflow.shark.WorkflowAssignmentManager - [processId=10_expenseclaim_process1,
processDefId=expenseclaim#1#process1, participantId=approver, next user=[admin]]
2310 INFO 25 Jun 2019 11:30:11 org.joget.workflow.shark.WorkflowAssignmentManager - [processId=10_expenseclaim_process1,
processDefId=expenseclaim#1#process1, participantId=claimant, next user=[admin]]
2311 INFO 25 Jun 2019 11:30:25 org.joget.workflow.shark.WorkflowAssignmentManager - [processId=10_expenseclaim_process1,
processDefId=expenseclaim#1#process1, participantId=approver, next user=[admin]]
2312 INFO 25 Jun 2019 11:30:39 org.joget.workflow.shark.WorkflowAssignmentManager - [processId=10_expenseclaim_process1,
processDefId=expenseclaim#1#process1, participantId=finance, next user=[admin]]

```

All Apps | Users | Monitor | Settings

# System Logs

- Displays system-wide log information and errors inclusive of all Apps.
- Admin Bar > Monitor > System Logs.



The screenshot shows the 'System Logs' section of the Joget DX Enterprise application. The interface includes a top navigation bar with 'Home', 'Inbox', 'Profile (admin)', and 'Logout' options. A left sidebar contains navigation items: 'Monitor Apps', 'Running Processes', 'Completed Processes', 'Audit Trail', 'Tomcat Logs', 'System Logs' (highlighted), and 'Performance'. The main content area displays a list of log entries with line numbers 3032 through 3051. The logs show a sequence of events related to an expense claim process, including commit operations, email sending, and workflow assignments. A 'Download' button and an 'Unfollow' button are visible at the top right of the log list. The bottom of the interface features a footer with 'All Apps', 'Users', 'Monitor', and 'Settings' icons, along with a green arrow button on the right.



# What can you do with the Web App Log Viewer?

---

- Download the log file by clicking the **Download** button.
- Click **Follow** button to display newly added lines from a log file in real-time on the web console.



# Chapter Review

---

- General understanding on how to access log via Web Log Viewer.

---

# Chapter 5

## Application Performance Monitoring (APM)

# What is APM?

---

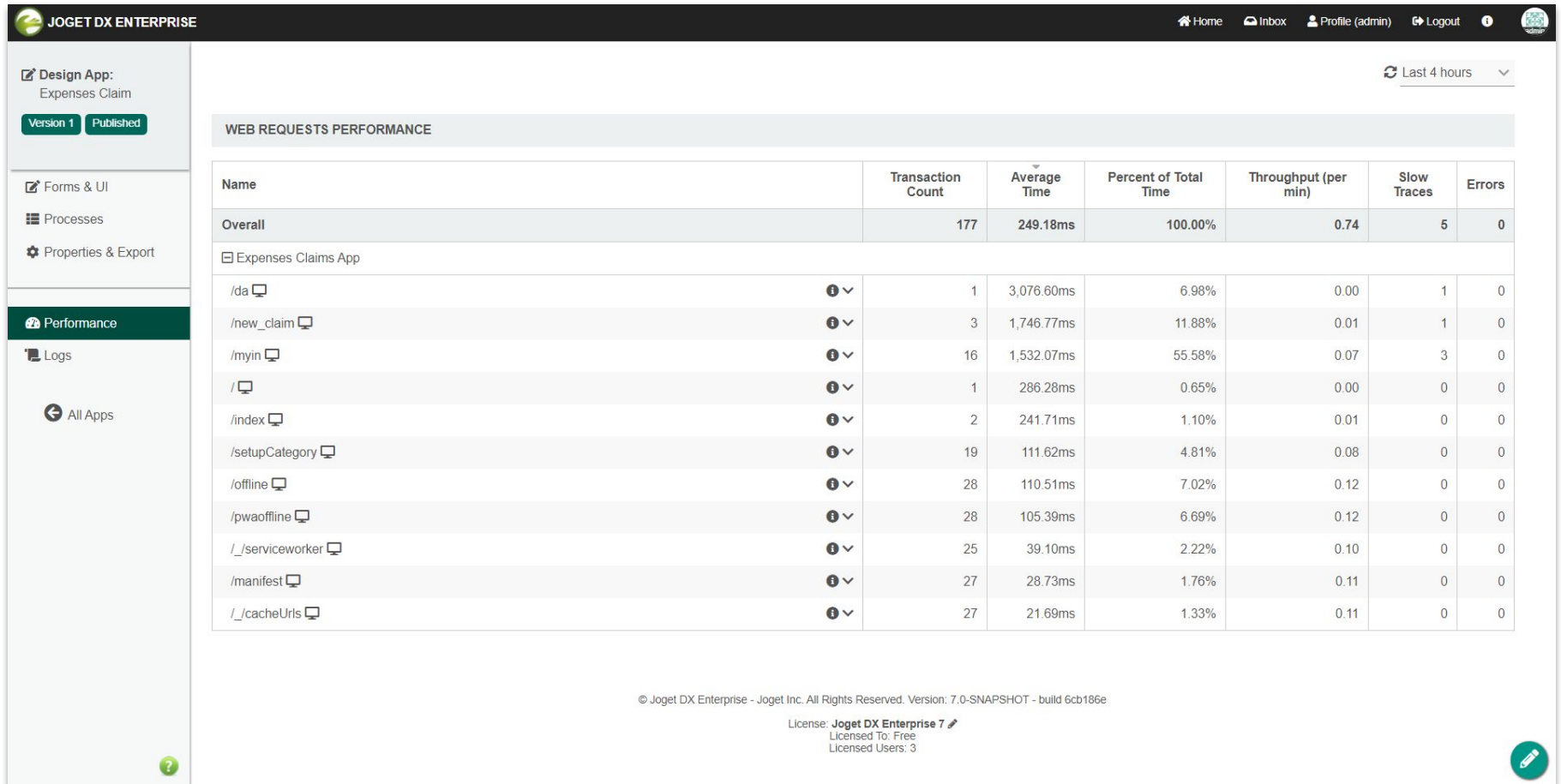
- New DX Feature.
- Built-in feature that automatically monitors system and application performance during runtime.
- Alert notification can be configured when user-defined threshold have exceeded based on various metrics.

Reference:

<https://dev.joget.org/community/display/DX7/Application+Performance+Management>

# App-level Performance Monitoring

- Can be accessed by selecting the respective App > Design App > Performance.



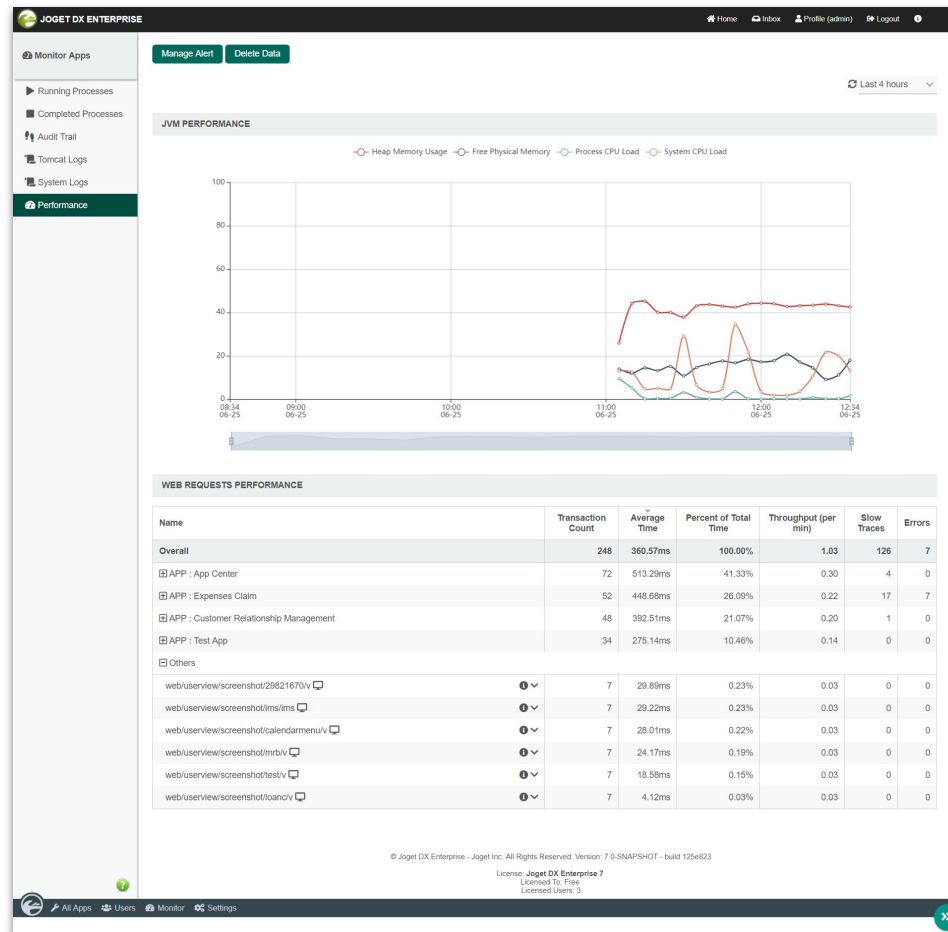
The screenshot shows the 'Performance' section of the Joget DX Enterprise interface for the 'Expenses Claim' app. The 'WEB REQUESTS PERFORMANCE' table displays the following data:

Name	Transaction Count	Average Time	Percent of Total Time	Throughput (per min)	Slow Traces	Errors
<b>Overall</b>	177	249.18ms	100.00%	0.74	5	0
<b>Expenses Claims App</b>						
/da	1	3,076.60ms	6.98%	0.00	1	0
/new_claim	3	1,746.77ms	11.88%	0.01	1	0
/myin	16	1,532.07ms	55.58%	0.07	3	0
/	1	286.28ms	0.65%	0.00	0	0
/index	2	241.71ms	1.10%	0.01	0	0
/setupCategory	19	111.62ms	4.81%	0.08	0	0
/offline	28	110.51ms	7.02%	0.12	0	0
/pwaoffline	28	105.39ms	6.69%	0.12	0	0
/_serviceworker	25	39.10ms	2.22%	0.10	0	0
/manifest	27	28.73ms	1.76%	0.11	0	0
/_cacheUrls	27	21.69ms	1.33%	0.11	0	0

© Joget DX Enterprise - Joget Inc. All Rights Reserved. Version: 7.0-SNAPSHOT - build 6cb186e  
 License: Joget DX Enterprise 7  
 Licensed To: Free  
 Licensed Users: 3

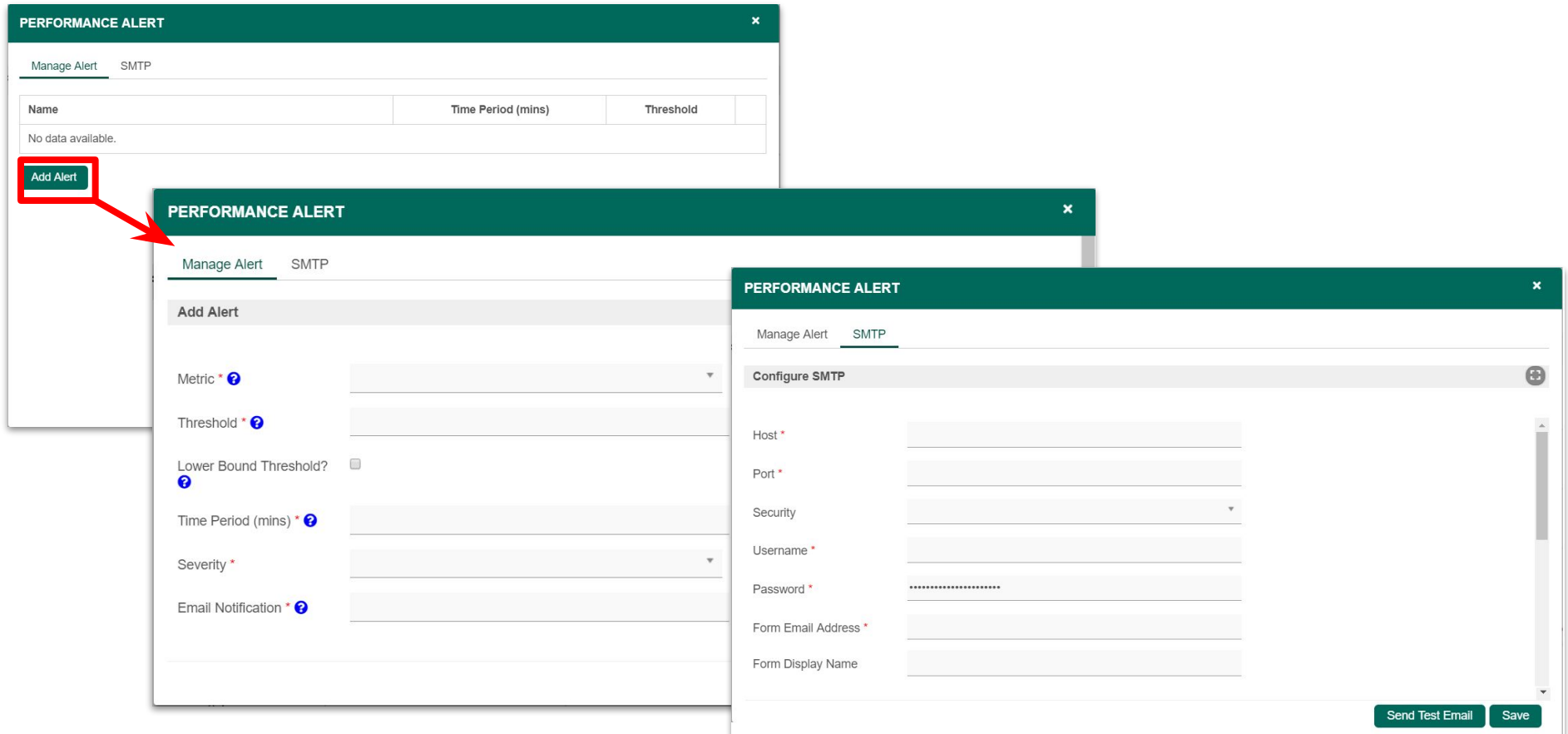
# Platform-level Performance Monitoring

- Can be accessed via Admin Bar > Monitor > Performance.



# Manage Alerts

- At the platform-level, Admin can configure alert email notifications based on a selection of metrics.



The screenshot displays the 'PERFORMANCE ALERT' management interface. It shows three overlapping windows:

- Top Window:** 'PERFORMANCE ALERT' with tabs for 'Manage Alert' and 'SMTP'. It contains a table with columns 'Name', 'Time Period (mins)', and 'Threshold'. The table is empty with the text 'No data available.' below it. A red box highlights the 'Add Alert' button.
- Middle Window:** 'PERFORMANCE ALERT' with tabs for 'Manage Alert' and 'SMTP'. It shows the 'Add Alert' form with fields for: Metric, Threshold, Lower Bound Threshold? (checkbox), Time Period (mins), Severity, and Email Notification.
- Bottom Window:** 'PERFORMANCE ALERT' with tabs for 'Manage Alert' and 'SMTP'. It shows the 'Configure SMTP' form with fields for: Host, Port, Security, Username, Password, Form Email Address, and Form Display Name. At the bottom right, there are 'Send Test Email' and 'Save' buttons.

# Chapter Review

---

- General understanding on how APM works in Joget.



# Module Review

---

1. Typical stack for Joget.
2. Basic Database Management (MariaDB).
3. Basic Application Server Management (Tomcat).
4. Web Log Viewer.
5. Application Performance Monitoring.

# Recommended Further Learning

---

- Best Practices on Application Building
- Server performance tuning and hardening.
- Database server performance tuning.

# Stay Connected With Joget DX

---

- [www.joget.org](http://www.joget.org)
- [community.joget.org](http://community.joget.org)
- [twitter.com/jogetworkflow](https://twitter.com/jogetworkflow)
- [facebook.com/jogetworkflow](https://facebook.com/jogetworkflow)
- [youtube.com/jogetworkflow](https://youtube.com/jogetworkflow)