

---

**Joget DX**

# Best Practices on Application Building

 <http://facebook.com/jogetworkflow>

 <http://twitter.com/jogetworkflow>

# Prerequisites

---

1. Understand all major components of Joget.

# Content

---

1. Application Building Best Practices
2. Performance



# Chapter 1

## Application Building Best Practices

# Application Building Best Practices

---

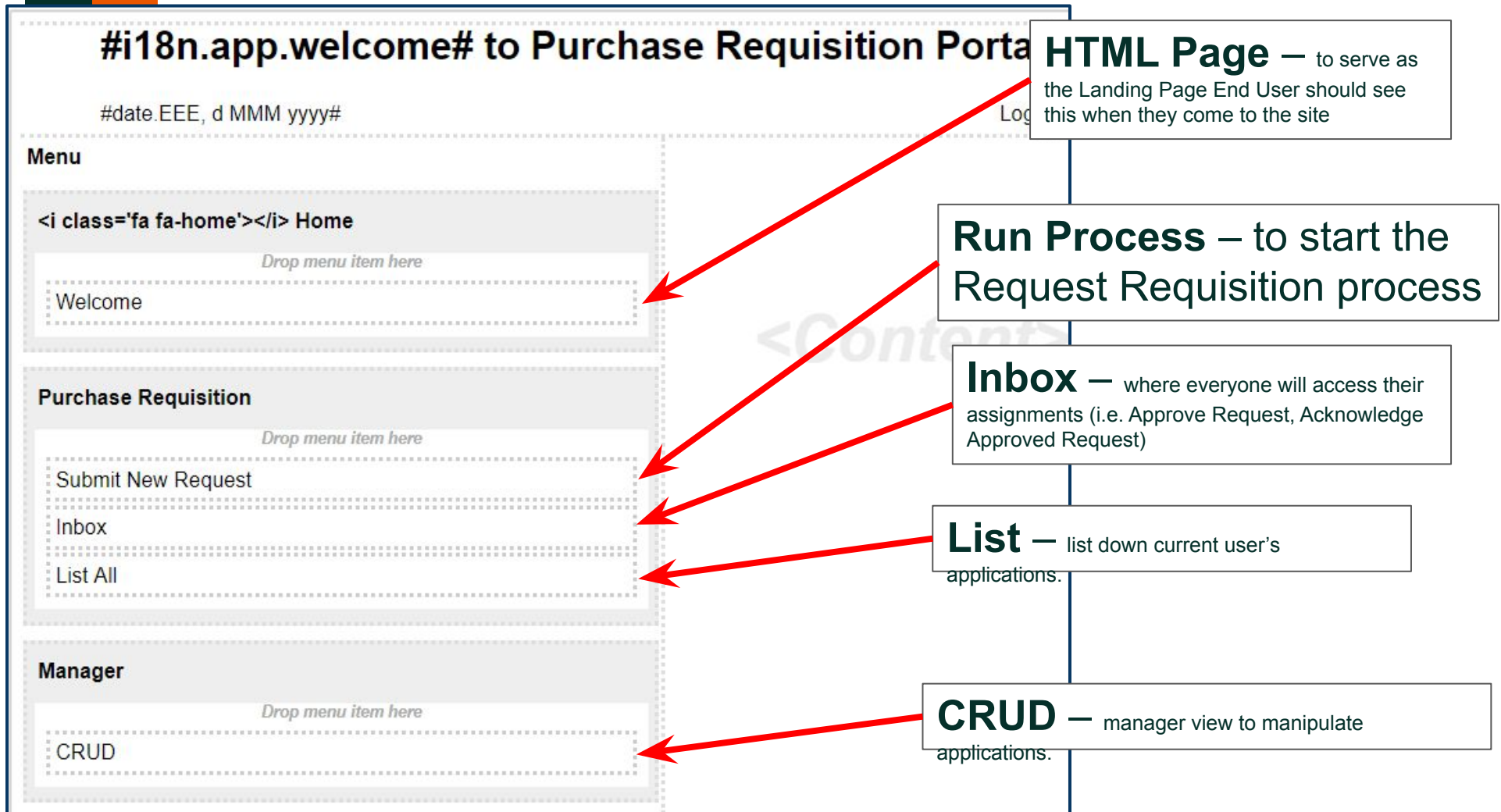
1. Draw a Sitemap
2. Segregate Userview by Roles
3. Iterative Build Process
4. Presence Indicator
5. Naming Convention
6. Taggings
7. Beanshell Coding
8. App Versioning
9. Process or Data
10. Defaults
11. Notes
12. Comments
13. Chrome Groups

# Draw a Sitemap

---

- Before you start designing your app, try to visualize on all these components would work together.
  - Process
  - Form
  - List
  - Userviewby using the Userview as the centerpiece of the planning.

# Draw a Sitemap – Purchase Requisition App






# Segregate Userview by Roles

If you have multiple roles and a lot of userview menu elements, it is best to segregate the userview by roles.

Benefits:

- Clear cut Permission Management
- Performance (CRUD / Inbox Count, etc)

## USERVIEW

-  App Admin Userview
-  Approver Userview
-  Requester Userview



# Iterative Build Process

---

- Build as you go.
- Establish the most basic layer first.
- Modularize.
- Design, configure and test individual module independently when possible.
- Increase complexity between each iteration.
- Teamwork is good but....
  - Must work in the same server/instance.
  - Must work on separate item/sections at any point of time.
  - Communication and expectation must be set clearly between each team member.
  - Assign a App champion to oversee and delegate tasks.

# Iterative Build Process

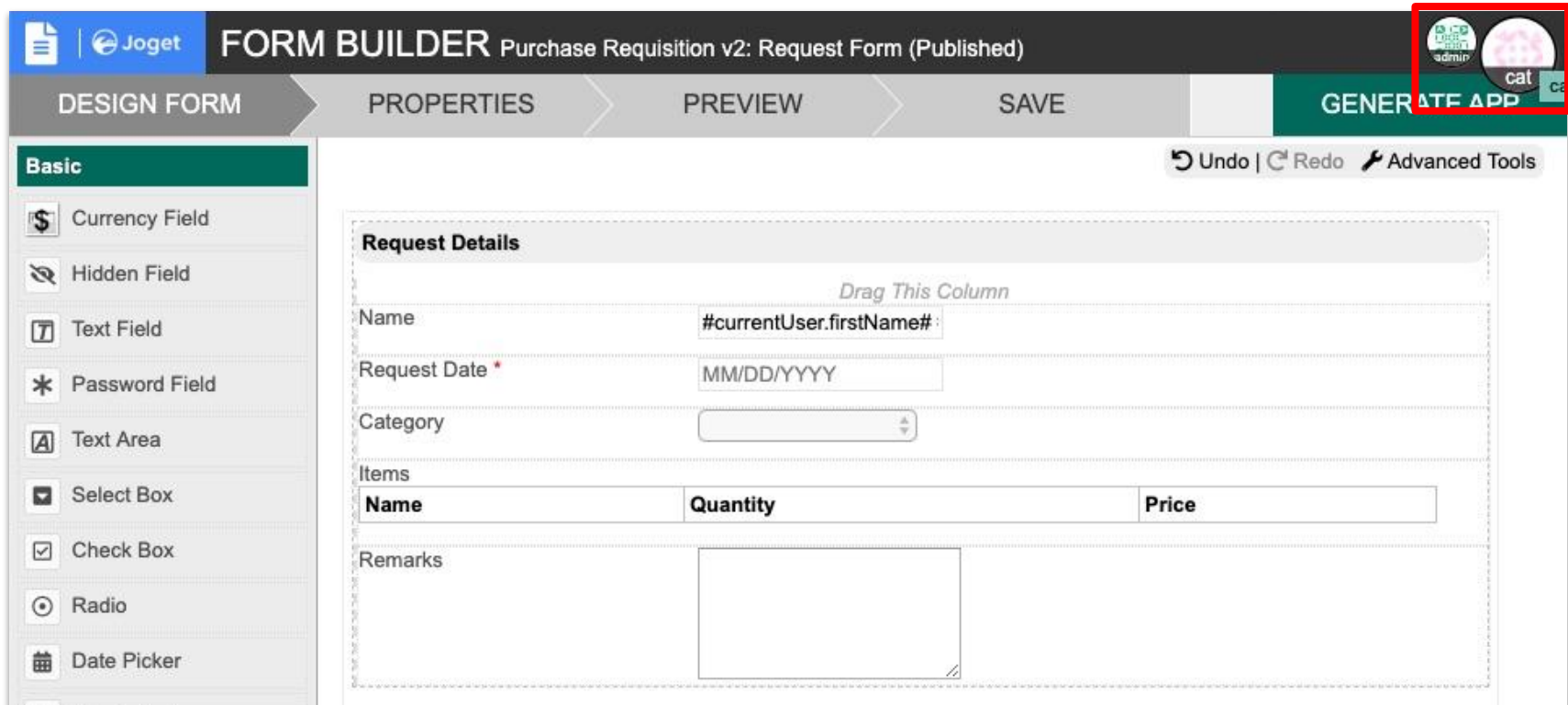
Incrementally go through each step thoroughly before going to the next...

1. Process Design
  - Verify that all possible routes flow as intended to.
2. Form Design
  - Build and test with and without being part of the process.
  - Define table names with ERD in mind.
3. Form with Process
  - With form and process individually tested and verified, then only add Form layer on top of process layer.
4. Datalist
  - With all Form and Process verified and tested, then only design the Datalist for reporting purpose.
5. Userview
  - Should come last depending on the flow of design you have. With everything tested, one can now add in permission control across all the layers.

Once a cycle is completed, consider versioning it to save your work before moving to next new cycle.

# Presence Indicator

- Real-time presence will check to see if someone else is viewing or editing a Joget Component while you have it open  
- Helps avoid conflicts and promote collaboration.



The screenshot shows the Joget Form Builder interface for a 'Purchase Requisition v2: Request Form (Published)'. The interface includes a top navigation bar with 'DESIGN FORM', 'PROPERTIES', 'PREVIEW', 'SAVE', and 'GENERATE APP' buttons. A presence indicator is visible in the top right corner, showing a red box around the 'admin' and 'cat' user avatars. The main workspace displays a form titled 'Request Details' with the following fields:

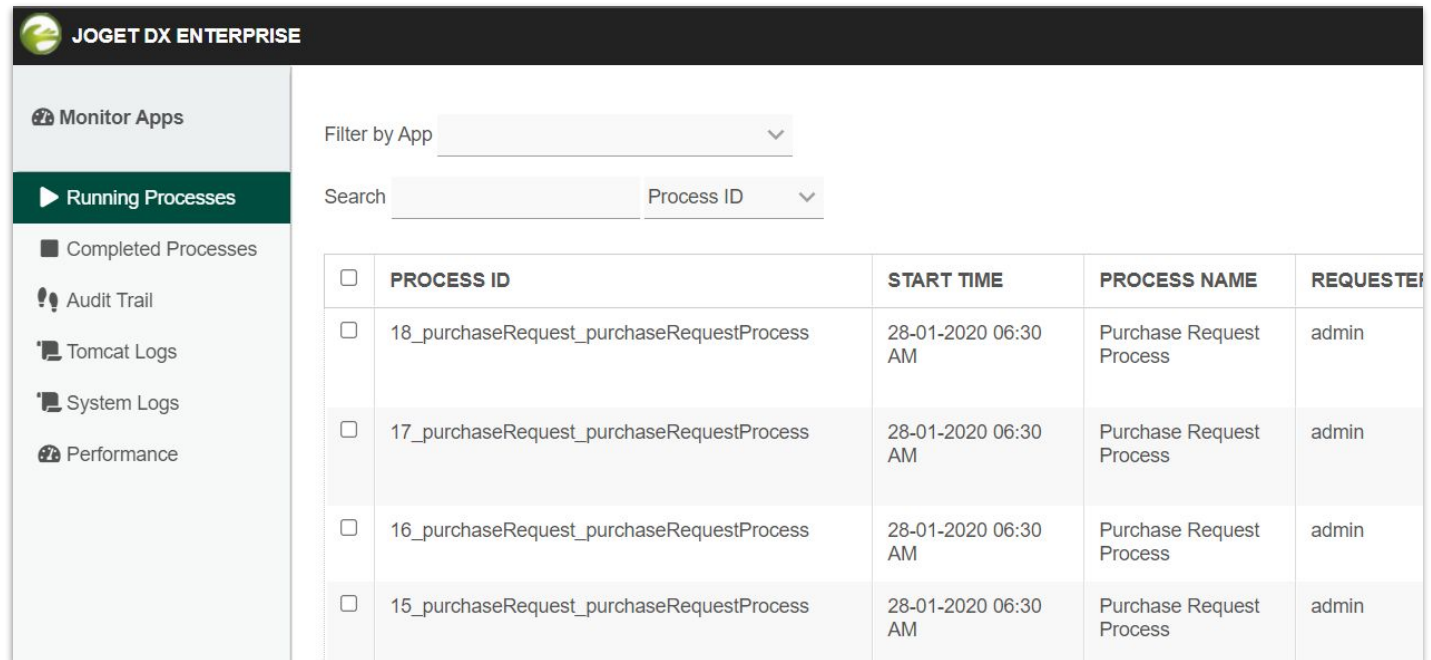
- Name: #currentUser.firstName#
- Request Date: MM/DD/YYYY
- Category: [Dropdown]
- Items:
 

Name	Quantity	Price
[Empty table body]		
- Remarks: [Text Area]

The left sidebar contains a 'Basic' section with various form components: Currency Field, Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, and Date Picker. The top right corner of the interface includes 'Undo', 'Redo', and 'Advanced Tools' buttons.

# Naming Convention

- Use camelCase when appropriate. Particularly:-
  - App ID
  - Process ID
  - Activity ID



The screenshot shows the 'Running Processes' section of the Joget DX Enterprise interface. It features a sidebar with navigation options: Monitor Apps, Running Processes (selected), Completed Processes, Audit Trail, Tomcat Logs, System Logs, and Performance. The main area has a search bar and a dropdown menu set to 'Process ID'. Below is a table with the following data:

<input type="checkbox"/>	PROCESS ID	START TIME	PROCESS NAME	REQUESTER
<input type="checkbox"/>	18_purchaseRequest_purchaseRequestProcess	28-01-2020 06:30 AM	Purchase Request Process	admin
<input type="checkbox"/>	17_purchaseRequest_purchaseRequestProcess	28-01-2020 06:30 AM	Purchase Request Process	admin
<input type="checkbox"/>	16_purchaseRequest_purchaseRequestProcess	28-01-2020 06:30 AM	Purchase Request Process	admin
<input type="checkbox"/>	15_purchaseRequest_purchaseRequestProcess	28-01-2020 06:30 AM	Purchase Request Process	admin

Because Joget uses underscore ( \_ ) as the separator in the naming of Process Instance ID.

# Naming Convention

---

- Use **camelCase** for the rest as well to maintain consistency.
  - Form ID
  - Datalist ID
  - Userview ID
  - Workflow Variable

# Naming Convention

---

- Use **snake\_case** when dealing with **database** related fields.  
Particularly:-
  - Form Element ID.
  - Form Table Name.
- Define a **prefix** for **Form Table Name**
  - **All apps'** form data are stored in the same database.
  - Prevent other App from writing into other App's set of tables.

# Naming Convention

---

- Not to be confused with Form Table Name on the previous slide, **Form Name** should be named with **process prefix** and **numbering** if it is bounded to a process.
- Example:-

Form Name
R-1-Submit Request
R-2-Approve Request
R-3-Acknowledge Approved Request
QR-1-Submit Quick Request
QR-2-Acknowledge Approved Quick Request

# Taggings

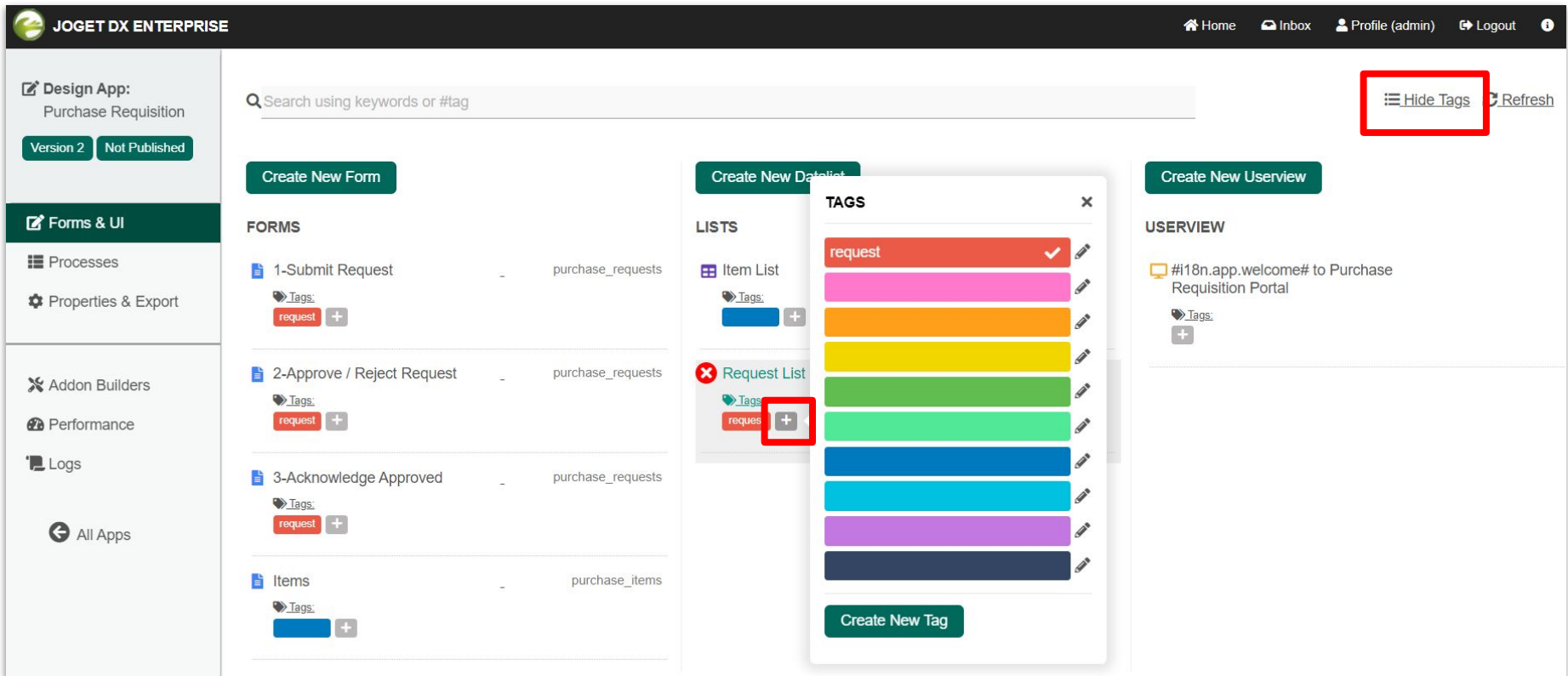
---

- Taggings and Labels are an extra layer of visual data and organization to the Joget components.
- View organized tags at a glance.
- Can also be used in the search filter.



# How to Add Tag?

- In **Design App**, click on **Show Tags** and click on **+** sign to add color tags and add labels to the color tags.



The screenshot displays the Joget DX Enterprise interface. At the top, the navigation bar includes 'Home', 'Inbox', 'Profile (admin)', and 'Logout'. The main content area is divided into several sections:

- Design App:** Purchase Requisition, Version 2, Not Published.
- Forms & UI:** Processes, Properties & Export.
- Addon Builders:** Performance, Logs.
- All Apps:** A back arrow icon.

The central area shows a search bar and a 'Hide Tags' button (highlighted with a red box). Below this, there are three main sections: 'FORMS', 'LISTS', and 'USERVIEW'. The 'FORMS' section lists several forms, each with a 'Tags' field and a '+' sign. The 'LISTS' section shows a 'Request List' with a '+' sign (highlighted with a red box). The 'TAGS' modal window is open, showing a list of color tags with the word 'request' entered in the first tag. A '+' sign is also highlighted in red in the modal. The 'USERVIEW' section shows a user view for '#i18n.app.welcome# to Purchase Requisition Portal' with a 'Tags' field and a '+' sign.

# Use Search Filter With #tag

- Use Search filter with prefix # to search the relevant tag.
- If no text were defined in the tag, you can search filter by color name.

Q #request

**Create New Form**

FORMS

- 1-Submit Request - purchase\_requests
  - Tags: request +
- 2-Approve / Reject Request - purchase\_requests
  - Tags: request +
- 3-Acknowledge Approved
  - Tags: request +

**Create New Datalist**

LISTS

- Request List
  - Tags: request +

**Create New Userview**

USERVIEW

Q #blue

**Create New Form**

FORMS

- Items - purchase\_items
  - Tags: [blue] +

**Create New Datalist**

LISTS

- Item List
  - Tags: [blue] +

**Create New Userview**

USERVIEW

# Beanshell Coding

---

- Do not use **System.out.println** as it will not be displayed in **Monitor > System Logs** or **joget.log**
- Use *LogUtil* method instead (see **LogUtil.java**)

## Examples:

```
LogUtil.debug(className,message)
```

```
LogUtil.error(className,throwableException,message)
```

```
LogUtil.info(className,message)
```

```
LogUtil.warn(className,message)
```

## Reference:

<https://dev.joget.org/community/display/DX7/Managing+Log+Files>

# Version Control - App Versioning

---

- Between each **development** cycle, it is recommended to create a new App version before moving on to the next iteration.
- Plan your release to the production.
- **Beware:**
  - Each “New Version” and “Import App” would increase the App Version count in your environment.
  - Production server’s app version **!=** Dev server’s app version.

# Version Control

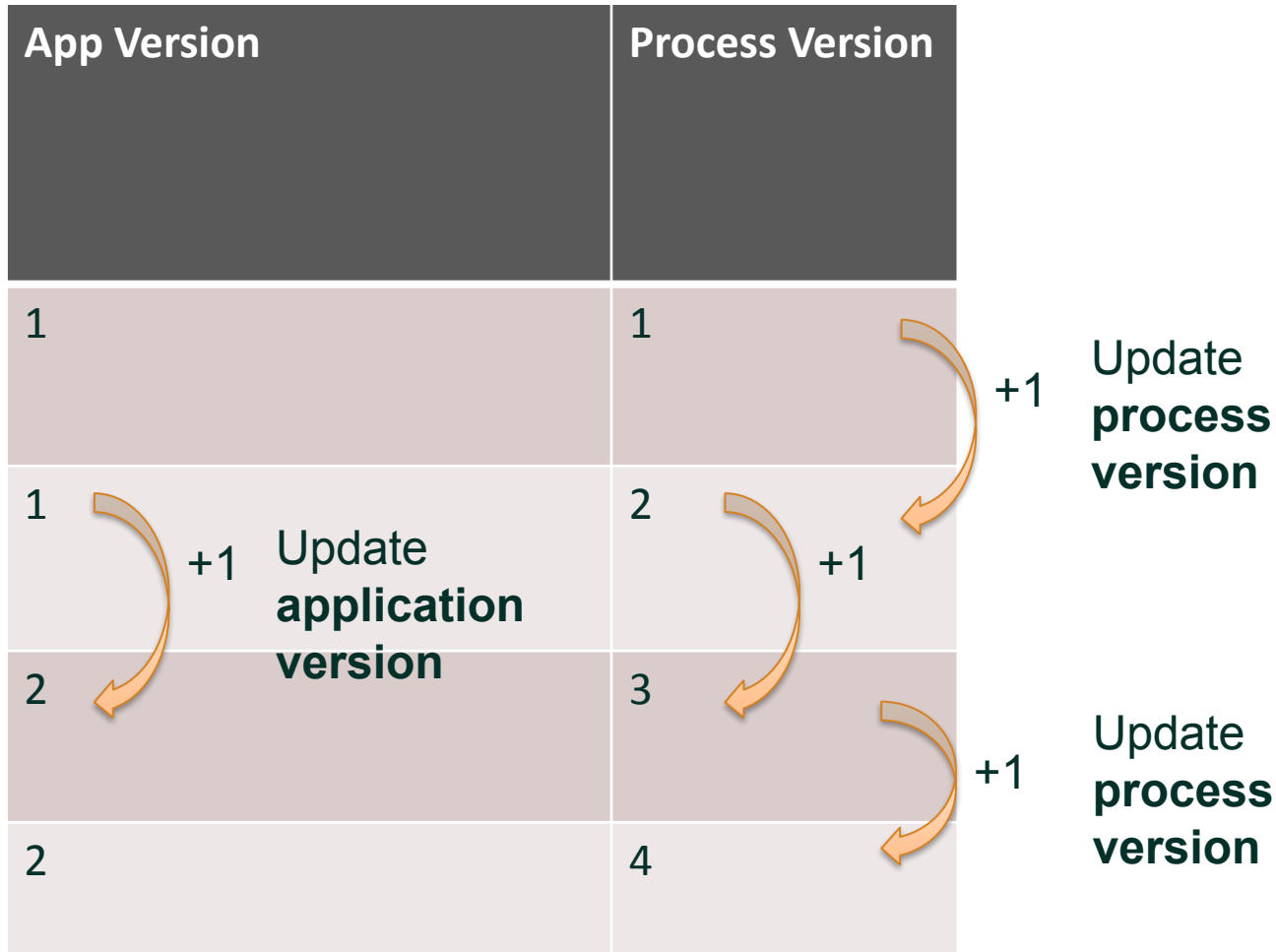
App Version	Process Version
1	1
1	2
2	3
2	4

+1 Update process version

+1 Update application version

+1

+1 Update process version



# Version Control - GIT

- Joget DX has in-built Git version-control system.
- You may use it to track changes made to your app.
- Find out more in Module 12.

Joget-DX7-Enterprise > wflow > app\_src > versionControl > versionControl\_1 >









Name	Date modified	Type
.git	23-Dec-19 3:15 PM	File folder
forms	23-Dec-19 3:15 PM	File folder
userviews	23-Dec-19 3:15 PM	File folder
appConfig.xml	23-Dec-19 3:15 PM	XML Documer
appDefinition.xml	23-Dec-19 3:15 PM	XML Documer
package.xpdl	23-Dec-19 3:15 PM	XPDL File

# Version Control - GIT

- Ensure to use **named user** during development, it will make it more meaningful to track changes

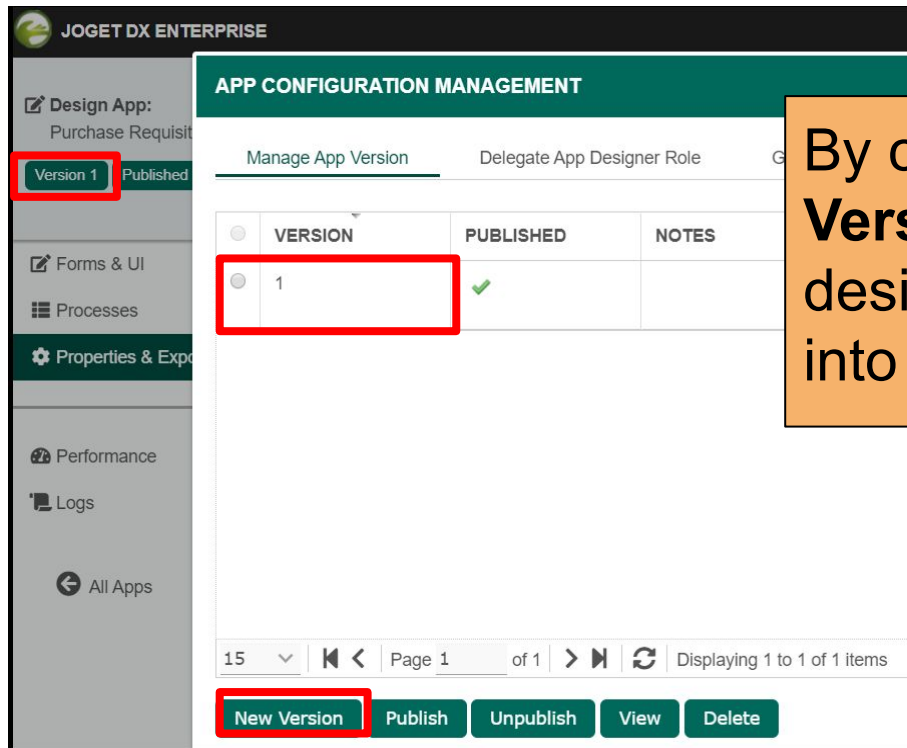
## Commits

Search commits  🔍 All branches ▾

Author	Commit	Message
 clark	<a href="#">80ac032</a>	Update app definition training. Up... <a href="#">🔗 training_3</a>
 clark	<a href="#">1ec58da</a>	Add form request. Add list list_requ... <a href="#">🔗 training_3</a>
 clark	<a href="#">3807a2f</a>	Initial commit for training_3 <a href="#">🔗 training_3</a>
 cat	<a href="#">c5dec58</a>	Add list list_request. Add userview ... <a href="#">🔗 training_2</a>
 cat	<a href="#">1def325</a>	Add form request. Update app con... <a href="#">🔗 training_2</a>
 cat	<a href="#">c2382cb</a>	Initial commit for training_2 <a href="#">🔗 training_2</a>
 BCC	<a href="#">fad7997</a>	Update form request. <a href="#">🔗 training_1</a>
 BCC	<a href="#">d6d4554</a>	Add form request. <a href="#">🔗 training_1</a>

# How to Update Application Version?

1. App Control Panel > Versions > Select version > New Version.



By creating a **New Version**, the App design will be **cloned** into the new version.

Online Reference:

<https://dev.joget.org/community/display/DX7/App+Versioning+and+Publishing>



# How to Update Application Version?

---

## 2. Import App

- By importing the app into a Joget server, the Application Version will **increase by 1** over the existing version already in the server.

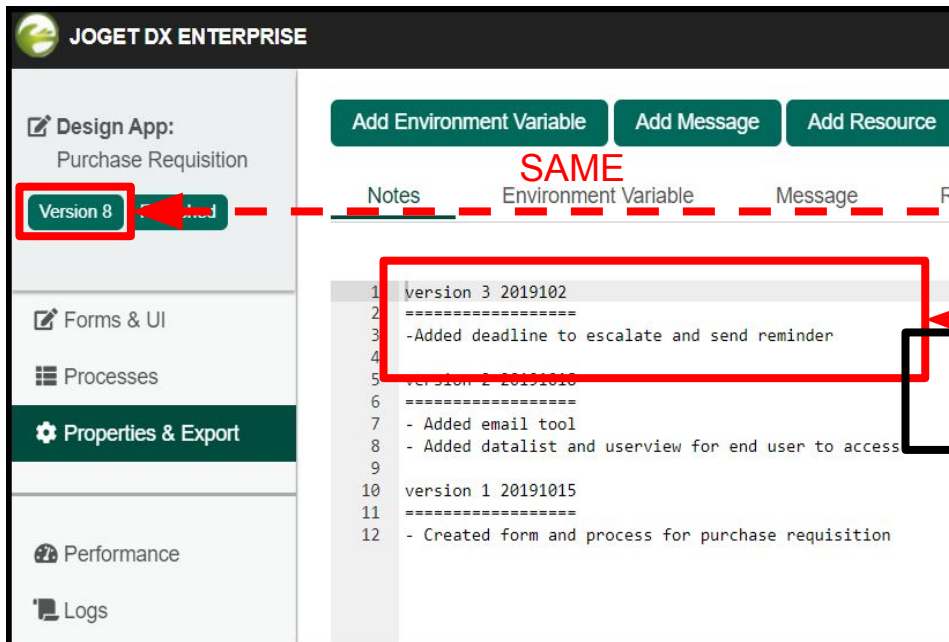
### **What does this means?**

When you are dealing with the same app across different Joget servers, you may end up with different app version in each server but with exact same app design.

# App Version Across Different Servers

- When you are dealing with the same app across different Joget servers, you may end up with different app version in each server but with exact same app design.

## Development Server



JOGET DX ENTERPRISE

Design App: Purchase Requisition

Version 8

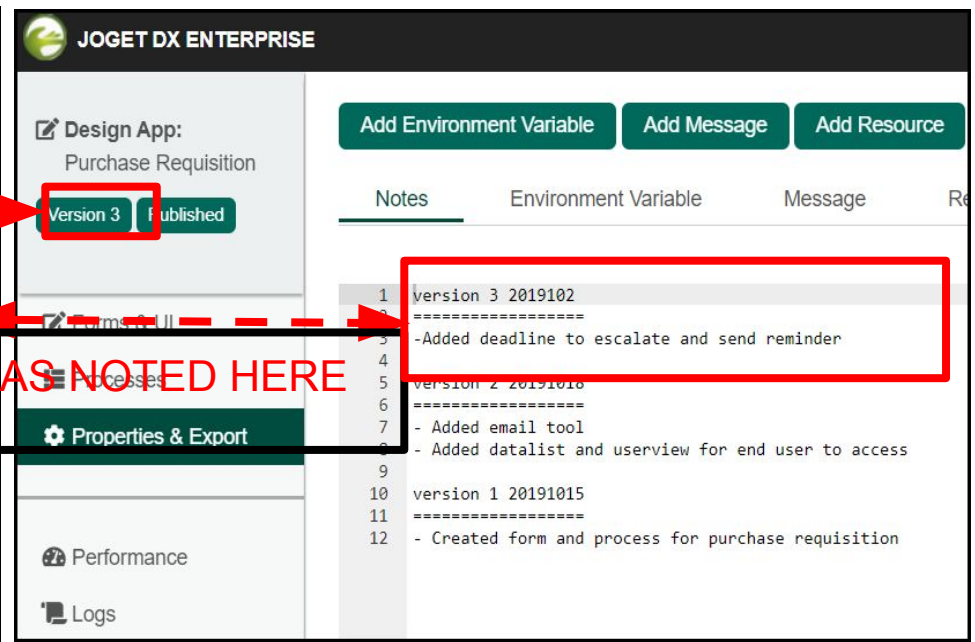
Notes Environment Variable Message

version 3 2019102  
-----  
- Added deadline to escalate and send reminder

version 2 20191018  
-----  
- Added email tool  
- Added datalist and userview for end user to access

version 1 20191015  
-----  
- Created form and process for purchase requisition

## Production Server



JOGET DX ENTERPRISE

Design App: Purchase Requisition

Version 3 published

Notes Environment Variable Message

version 3 2019102  
-----  
- Added deadline to escalate and send reminder

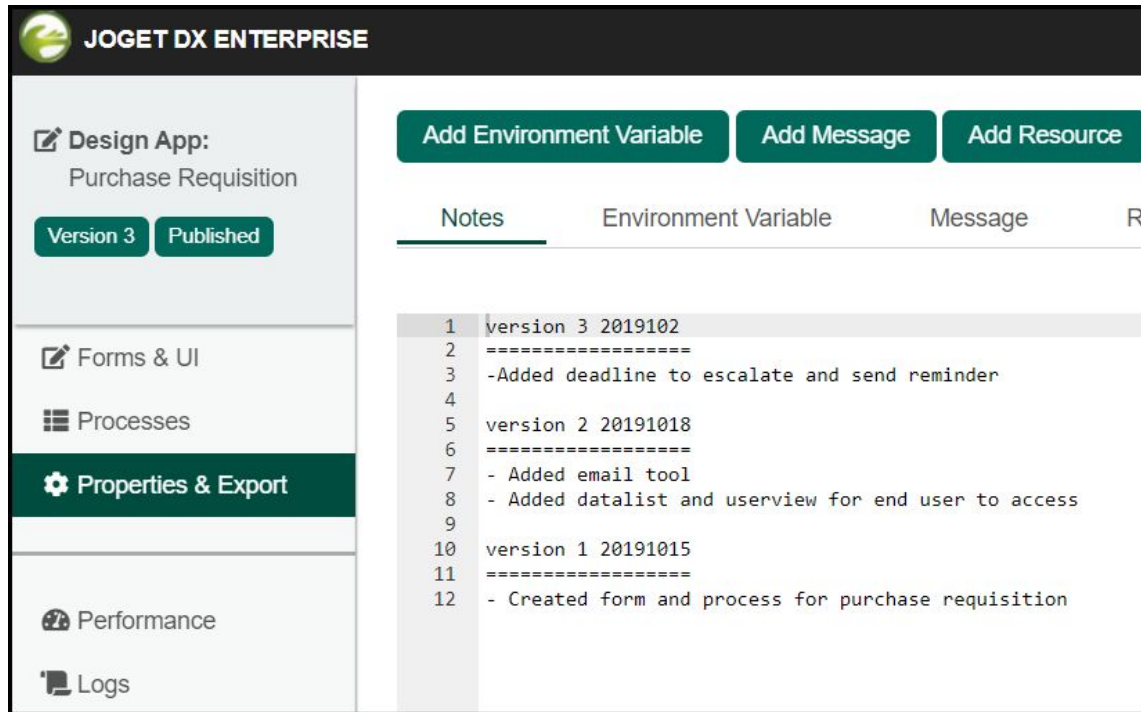
version 2 20191018  
-----  
- Added email tool  
- Added datalist and userview for end user to access

version 1 20191015  
-----  
- Created form and process for purchase requisition

AS NOTED HERE

# Keeping Track of App Design Across Different Servers

- With the nature of increment of the last app version when an app is imported in, it is imperative to keep track of the “real” app version (app design).
- Make use of **Notes** in app’s properties.



The screenshot displays the 'JOGET DX ENTERPRISE' interface. On the left, a sidebar contains navigation options: 'Design App: Purchase Requisition' (with 'Version 3' and 'Published' buttons), 'Forms & UI', 'Processes', 'Properties & Export' (highlighted), 'Performance', and 'Logs'. The main area shows three buttons: 'Add Environment Variable', 'Add Message', and 'Add Resource'. Below these, a tabbed interface is visible with 'Notes', 'Environment Variable', 'Message', and 'Resource' tabs. The 'Notes' tab is active, showing a list of notes:

Line	Content
1	version 3 2019102
2	=====
3	-Added deadline to escalate and send reminder
4	
5	version 2 20191018
6	=====
7	- Added email tool
8	- Added datalist and userview for end user to access
9	
10	version 1 20191015
11	=====
12	- Created form and process for purchase requisition

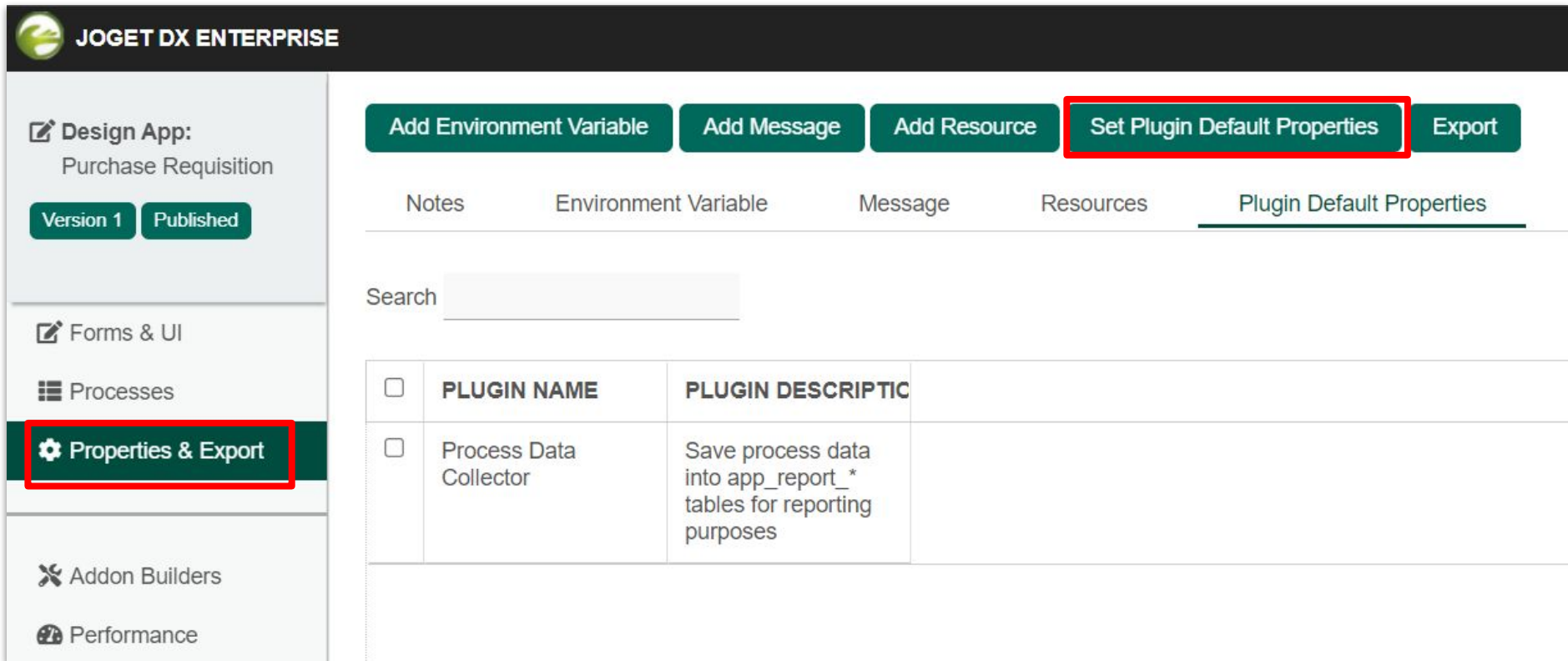
# Process or Data

---

- Do you really need a Process?
- Are you merely doing CRUD?

# Defaults

- To minimize maintenance, one may consider to use...
  - Plugin Default Properties.



The screenshot shows the Joget DX Enterprise interface. The top navigation bar includes the Joget logo and the text 'JOGET DX ENTERPRISE'. On the left sidebar, the 'Properties & Export' menu item is highlighted with a red box. The main content area features a top navigation bar with buttons for 'Add Environment Variable', 'Add Message', 'Add Resource', 'Set Plugin Default Properties' (highlighted with a red box), and 'Export'. Below this, there are tabs for 'Notes', 'Environment Variable', 'Message', 'Resources', and 'Plugin Default Properties' (which is the active tab). A search bar is located below the tabs. The main content area displays a table with the following data:

<input type="checkbox"/>	PLUGIN NAME	PLUGIN DESCRIPTIC
<input type="checkbox"/>	Process Data Collector	Save process data into app_report_* tables for reporting purposes

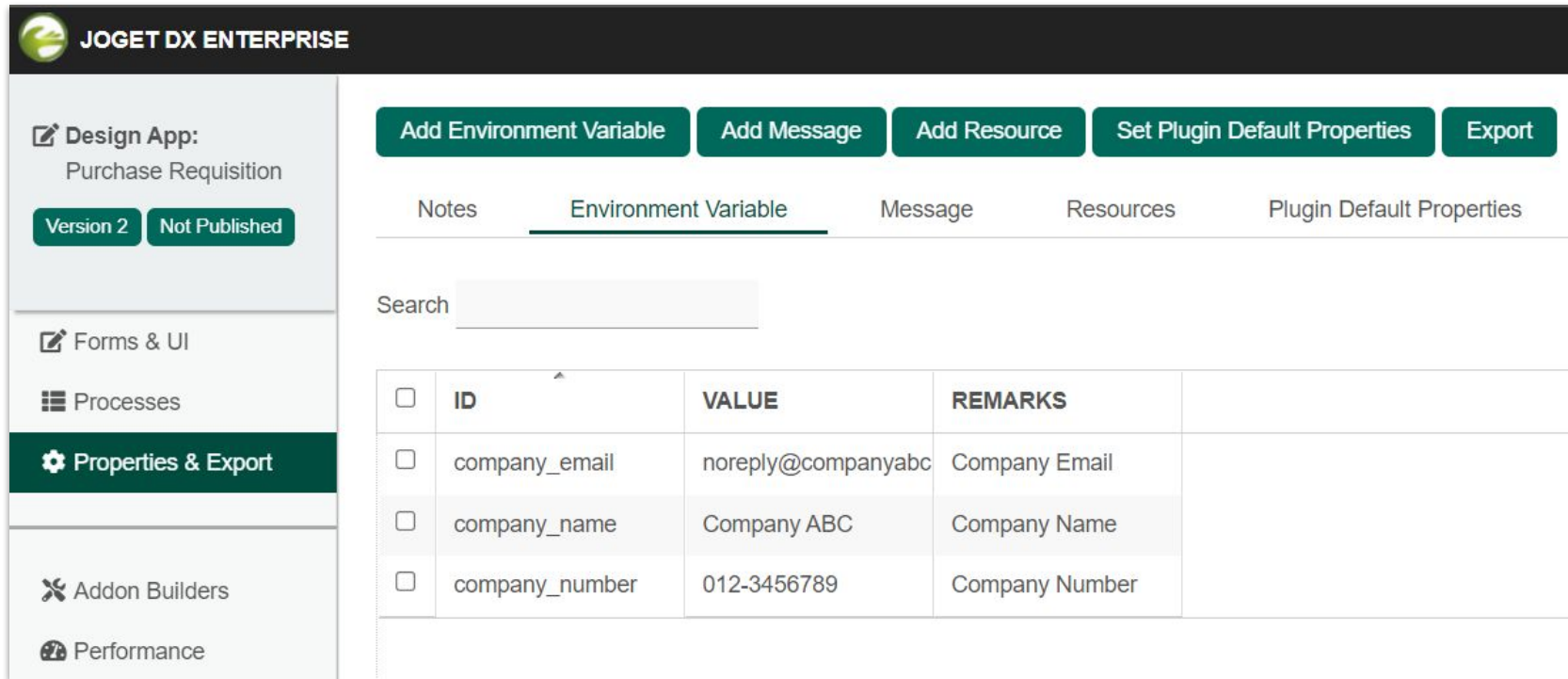
# About Plugin Default Properties...

---

- It's good to.... only apply/add in changes after each plugin is tested individually, to steer clear of configuration mistake(s) or bugs.

# Defaults

- To minimize maintenance, one may consider to use...
  - Environment Variable, to factorize frequently used values



**JOGET DX ENTERPRISE**

Design App: Purchase Requisition  
Version 2 Not Published

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

Add Environment Variable Add Message Add Resource Set Plugin Default Properties Export

Notes Environment Variable Message Resources Plugin Default Properties

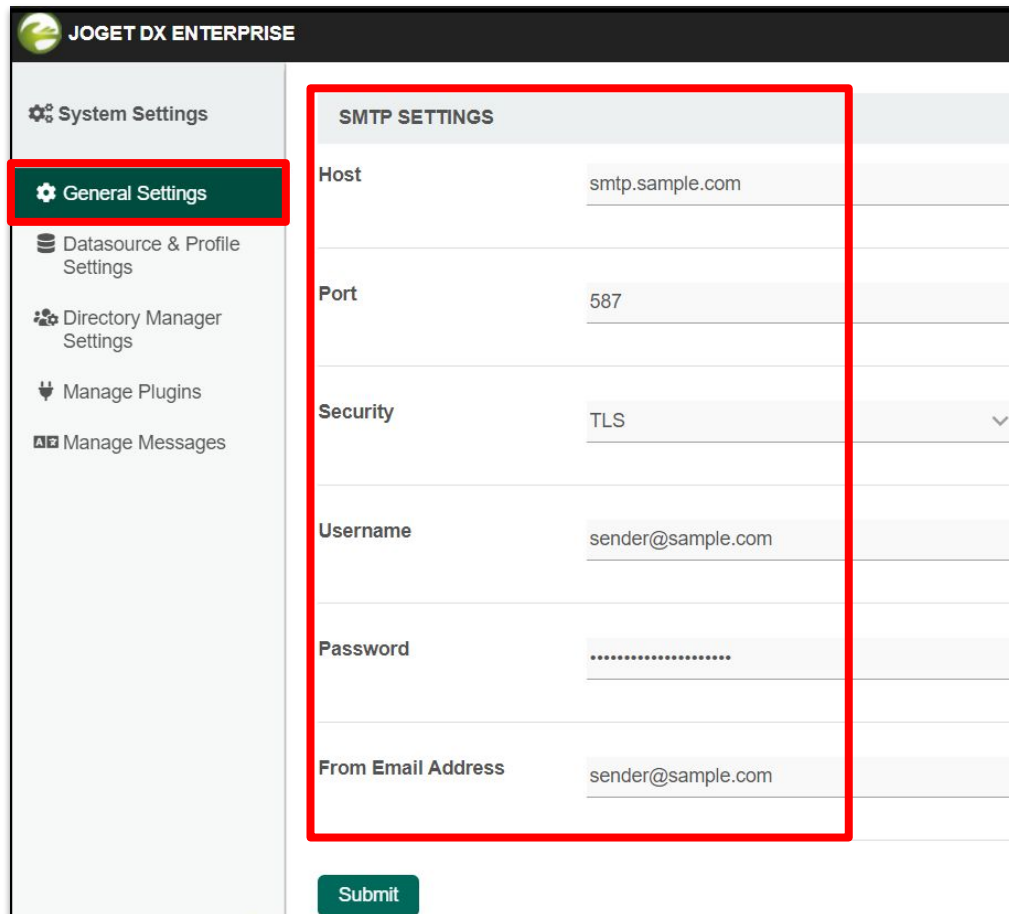
Search

<input type="checkbox"/>	ID	VALUE	REMARKS
<input type="checkbox"/>	company_email	noreply@companyabc	Company Email
<input type="checkbox"/>	company_name	Company ABC	Company Name
<input type="checkbox"/>	company_number	012-3456789	Company Number

Refer to our hash variable KB to learn how to use environment variables

# Defaults

- To minimize maintenance, one may consider to use...
  - Platform-level email settings.



The screenshot shows the 'JOGET DX ENTERPRISE' interface. On the left is a sidebar with navigation options: 'System Settings', 'General Settings' (highlighted with a red box), 'Datasource & Profile Settings', 'Directory Manager Settings', 'Manage Plugins', and 'Manage Messages'. The main content area is titled 'SMTP SETTINGS' and contains a form with the following fields:

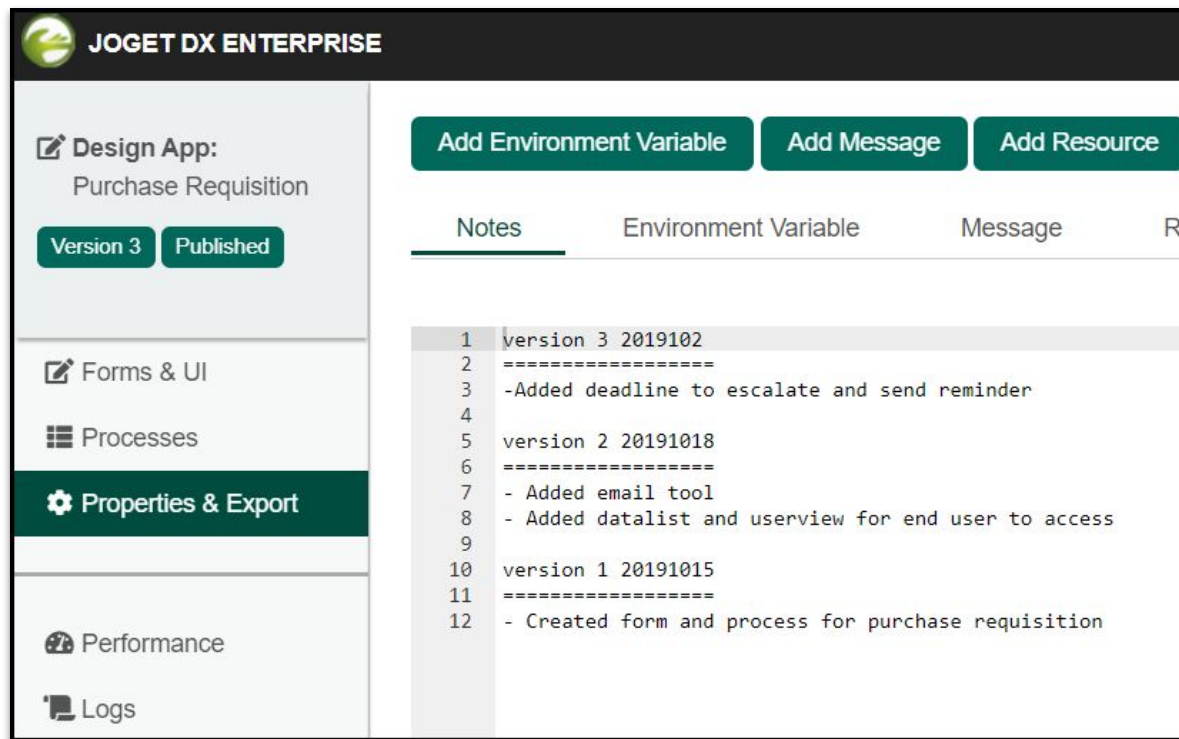
SMTP SETTINGS	
Host	smtp.sample.com
Port	587
Security	TLS
Username	sender@sample.com
Password	.....
From Email Address	sender@sample.com

A red box highlights the entire SMTP settings form area. At the bottom of the form is a 'Submit' button.



# Notes

- Use this feature to jot down important notes for your apps.
- Example: keep track for app development changes.



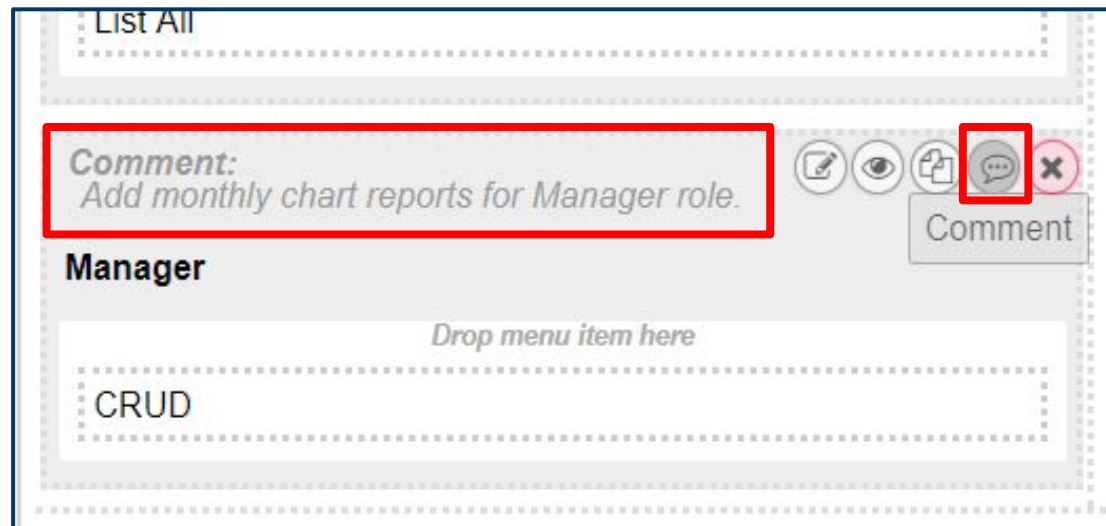
The screenshot displays the 'JOGET DX ENTERPRISE' interface. On the left sidebar, the 'Design App' section is active, showing 'Purchase Requisition' with 'Version 3' and 'Published' status. Below this are 'Forms & UI', 'Processes', 'Properties & Export' (highlighted), 'Performance', and 'Logs'. The main content area has three buttons: 'Add Environment Variable', 'Add Message', and 'Add Resource'. Below these buttons are tabs for 'Notes', 'Environment Variable', 'Message', and 'Resource'. The 'Notes' tab is selected, showing a list of notes with line numbers 1 through 12. The notes are organized into three sections: 'version 3 2019102', 'version 2 20191018', and 'version 1 20191015', each with a separator line of dashes.

Notes	Environment Variable	Message	Resource
1 version 3 2019102			
2 -----			
3 -Added deadline to escalate and send reminder			
4			
5 version 2 20191018			
6 -----			
7 - Added email tool			
8 - Added datalist and userview for end user to access			
9			
10 version 1 20191015			
11 -----			
12 - Created form and process for purchase requisition			

# Comments

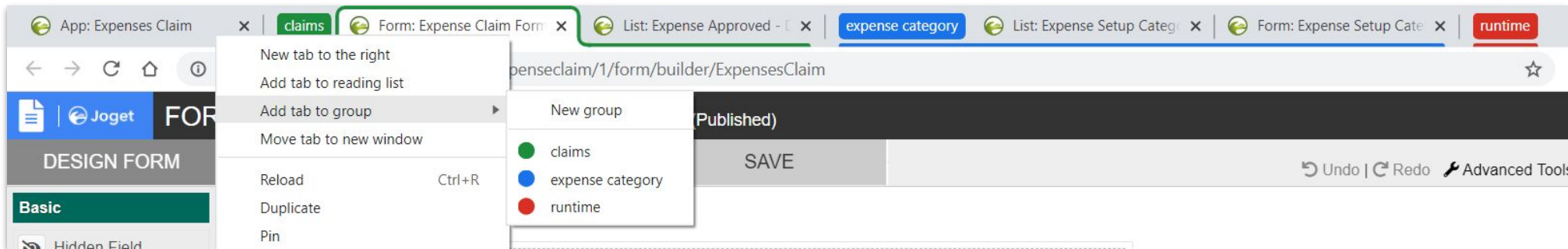
---

- Use this feature to comment on a form section/userview category. This can aid comprehension in a collaborative development team.
- Example: take note of the intention of a section or category/pending edits/notable info/etc.



# Chrome Browser - Add Tab to Group Function

- If you use chrome browser, try use the add tab to group feature to stay organized, avoid confusion and open relevant tabs only



---

# Chapter 2

## Performance

# About Performance...

---

- Performance varies **wildly** from one to another depending on factors such as:-
  - Total number of users.
  - Maximum expected concurrent users.
  - Number of apps running on the platform.
  - Complexity of each of the apps.
  - Amount of data generated in each app.
  - Network infrastructure.
  - etc...

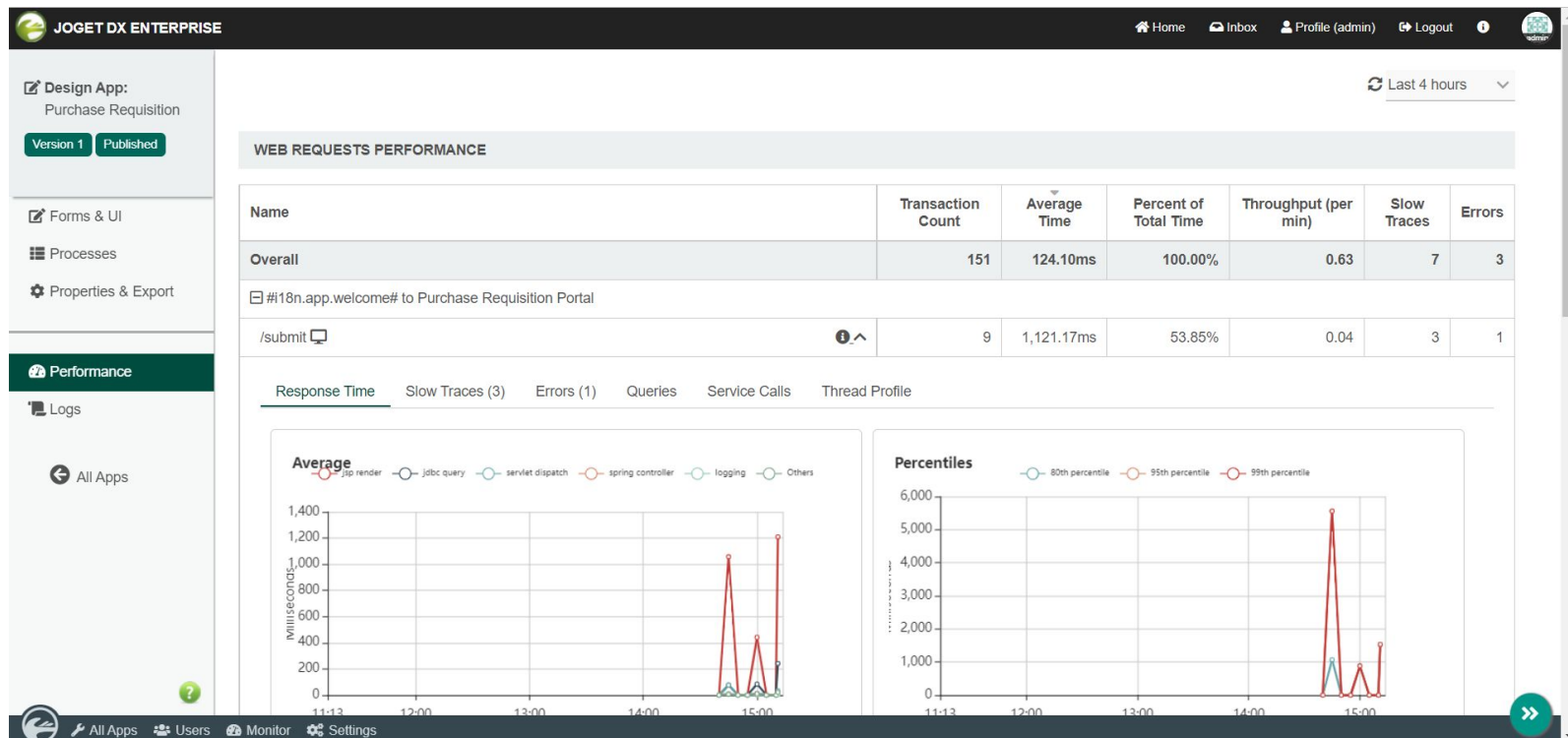
# Server sizing

---

- Since each app differs to one another:
  - There's no one-size-fits-all
  - If its too small = poor performance
  - if its too big = waste of resources
- Run stress test / load test to measure its capability
- Check APM & Performance Analyzer (DB connection leaks) to ensure it can sustain the load and does not crash the app / database / server

# Performance Monitoring

- Any resource usage (e.g. JDBC connections in Beanshell scripts or plugins, IO, etc) must be closed in a try-finally block to prevent leaks.
- Monitor using **APM** to monitor usage patterns across different apps.



# App Development DEVOPS style

---

- Use APM to continuously monitor slow traces and errors during development
- Continuously monitor existing pages for performance improvements and regressions.



# Scripting

---

- **Scripting**
  - Beanshell scripts are good for **rapid** development, but for large deployments, it may be slow.
  - Consider developing/porting to a plugin.

# Integration

---

- Any integration (e.g. SQL queries, REST calls, etc) may be slow, so optimize them.

# Database Indexing

---

- Form Data are stored with the prefix “**app\_fd\_**”.
- Columns are saved in **text/varchar** to maintain flexibility.
- Add indexing as needed, depending on use case.
- MySQL/MariaDB related: Enable slow query logging to track and improve slow queries.

# Caching

---

- Since v6, all userview elements can be cached to reduce resource-intensive server/database operations (for non time-sensitive elements only).
- BE CAREFUL when selecting caching scope.
- Visit <https://dev.joget.org/community/display/DX7/Performance+Improvement+with+Userview+Caching> to find out more.

# More...

---

- Visit

<https://dev.joget.org/community/display/DX7/Deployment+Best+Practices> for more deployment best practices and tips.

# Module Review

---

1. Application Building Best Practices
2. Performance

# Stay Connected With Joget

---

- [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)