

Dynamic Workflow – Server Side

USER GUIDE

Nathan Nguyn

V1.1

Table of Contents

[Overview 2](#_Toc22147577)

[Disclaimer 2](#_Toc22147578)

[Features 2](#_Toc22147579)

[Deployment Instructions 3](#_Toc22147580)

[Configuration 4](#_Toc22147581)

[Server 4](#_Toc22147582)

[Enterprise 4](#_Toc22147583)

[Features 4](#_Toc22147584)

[Integration Key 5](#_Toc22147585)

# Overview

Dynamic workflows allow users to specify the next participants within an agreement. It replicates the workflow created in workflow designer by creating a web form version. Both internal and external users can initiate an agreement using the selected workflow.

# Disclaimer

This application is currently on version 2.1. There is a known issue that has been submitted to JIRA for participant groups. The feature is currently there, but the functionality has a bug. Once this issue is resolved, a patch will be issued out.

# Features

|  |  |
| --- | --- |
| Features | Description |
| Dynamic Routing | Allows user to provide the next participant associated with the workflow. |
| Hide All Workflow w/ Predefined Recipients | Allows the application to hide workflows with predefined recipients. |
| Hide Target Workflows w/ Predefined Recipients | Specify which workflows you want to only hide predefined recipients. |
| CC Recipient | Allow users to provide CC recipients. |
| Hide All Workflow w/ Predefined CC Recipients | Allow the application to hide workflows with predefined CC recipients. |
| Hide Target Workflows w/ Predefined Recipients | Specify which workflows you want to only hide predefined CC recipients. |
| Document Names | Allow users to change the agreement name. |
| Messages | Allow users to provide a message to attach to the agreement. |
| Upload Documents | Allow the sender to upload additional documents if enabled in workflow. |
| Sender Input Fields | Allow workflow sender input fields to be replicated over into the form. |
| Passwords | Allow users to provide an open password to an agreement |
| Completion Deadline | Allows user to provide an expiration date on the agreement with a max date of 180 days. |
| Reminders | Allow the user to provide a frequency reminder on the agreement. |
| Recipient Groups | Allows you to have the ability to dynamically route to recipient groups.  \*As of version 1.1, this feature is blocked please see disclaimer. |
| CSS Theme | Easily change the theme by adjusting CSS variables . |

# Deployment Instructions

This is a server-side version of the Dynamic Workflow application. You must host and deploy this application for it to work. There are many methods of deploying an application. In this example, I will be deploying it to Digital Ocean. Feel free to choose whichever platform/services and use this documentation as a guideline.

#### Digital Ocean

1. Click create and select droplet
2. Choose an image. We will be going with Ubuntu 18.04.3 LTS x64
3. Choose a plan that best fits your business needs. For demo purposes, I will be choosing a Standard plan at $5/mo
4. Select your preferred datacenter region
5. Select any additional options
6. Choose an SSH for the authentication method
7. Create an SSH key for your droplet
8. Choose a hostname
9. Click create droplet
10. SSH into your server

#### Node.js

1. Refresh your local package index with the following command
   1. sudo apt update
2. Install Node.js from repositories
   1. sudo apt install nodejs
3. Install node packaging manager
   1. sudo apt install npm

#### Importing Source File

1. There are two options to import your source file.
   1. Github Repo
      1. git clone <url to your repo>
   2. FileZilla
      1. Use FileZilla to transfer application onto the server

#### Install Dependencies

1. You will need to install all dependencies associated with this application
   1. cd Dynamic\_Workflow
   2. npm install

#### Installing nodemon (Optional)

1. Use nodemon as a dev node on production. \*We will use pm2 as our process manager
   1. npm install -g nodemon
   2. npm start dev \*Use only for development

#### Default port 80 (Optional)

1. To run application on default port 80 install lib2cap-bin
   1. sudo apt-get install lib2cap-bin
   2. sudo setcap cap\_net\_bind\_service=+ep `readlink -f \`which node\``

#### Install pm2

1. Install pm2 to and run application in the background
   1. npm install pm2 -g
   2. Pm2 start ~/Dynamic\_Workflow/server.js

# Configuration

There is a sample configuration file in config/config.sample.yaml that needs to be copied to config/config.yaml and modified for your environment.

There are three main components within the configuration file.

## Server

* host: The host address to your Sign Console.
* endpoint: This is the API endpoint

## Enterprise

* Integration: This is your integration key for Adobe Sign. Please see the integration key section.

## Features

* Hide\_predefined: This turns on the feature to hide predefined recipients in your workflow.
  + Yes: Turn on
  + No: Turn off
* Hide\_Workflow\_List: A list to target specific workflows to hide predefined recipients. Please see configuration template in config/config.yaml
* Hide\_cc: This turns on the feature to hide predefined CC recipients in your workflow.
  + Yes: Turn on
  + No: Turn off
* Hide\_CC\_Workflow\_List: A list to target specific workflows to hide predefined cc recipients. Please see configuration template in config/config.yaml

# Integration Key

You will be required to create an integration key with limited scopes for this application and add the integration key inside the config.yaml file. [Click Me](https://helpx.adobe.com/sign/kb/how-to-create-an-integration-key.html)

