

Table of Contents

1. ENTANDO 5.0.0 RELEASE NOTES	1
1.1. ABSTRACT	1
1.2. PREFACE	1
1.3. GOAL OF ENTANDO 5	1
1.4. ABOUT ENTANDO 5	1
1.5. WHAT'S NEW WITH ENTANDO 5	1
1.5.1. KNOWN ISSUES	2
1.5.2. VERSIONING INFORMATION	3
1.5.3. SOFTWARE REQUIREMENTS	3
1.5.4. LEGAL NOTICE	3
1.5.5. LINKS AND RESOURCES	3
2. INSTALLING ENTANDO APP BUILDER 5.0	3
2.1. PREREQUISITES	4
2.1.1. INSTALLED PREREQUISITES:	4
2.1.2. SPACE AND HARDWARE REQUIREMENTS	4
2.2. CREATING A NEW ENTANDO PROJECT	4
2.3. LAUNCHING THE ENTANDO APPLICATION	4
2.3.1. QUICK START USING DERBY AND JETTY	4
2.4. SETTING UP A DATABASE:	5
2.5. DEPLOYING TO DOCKER USING S2I (SOURCE TO IMAGE)	5
2.5.1. PREREQUISITES:	5
2.6. USING OPENSIFT	6
2.6.1. PREREQUISITE:	6
2.7. LAUNCHING THE APP BUILDER	6
2.7.1. BUILD FROM SOURCE	6
CLONE AND SET UP	7
DEPLOY	7
2.7.2. BUILD THE MAPP ENGINE FROM SOURCE	7
INTEGRATE THE BPM FORM	8
3. USING ENTANDO APP MANAGER WITH THE RED HAT PROCESS APPLICATION MANAGER	9
3.1. INTRODUCTION	9
3.2. QUICK START FROM ZIP	9
4. CUSTOM APP BUILDING WITH ENTANDO	11
4.1. INTRODUCING THE ENTANDO APP BUILDER	11
4.1.1. ENTANDO ADMIN CONSOLE VERSION 5.0	11
4.2. APP BUILDER OVERVIEW	11
4.3. DASHBOARD	12
4.4. CREATING AND MANAGING PAGES WITH PAGE DESIGNER	12

4.4.1. USING THE PAGE TREE	12
CREATING A NEW PAGE	13
4.4.2. CONFIGURING A PAGE	14
4.5. UX PATTERNS: MANAGING WIDGETS, PAGE MODELS, AND FRAGMENTS	14
4.5.1. USING WIDGETS	14
4.5.2. USING FRAGMENTS	15
ADDING A NEW FRAGMENT	15
4.5.3. USING PAGE MODELS	16
CREATING A PAGE MODEL	16
4.6. INTEGRATING WITH RED HAT PROCESS AUTOMATION MANAGER (PAM)	17
4.7. MANAGING USERS	18
4.7.1. WORKING WITH USERS	19
CREATING A ROLE	19
CREATING A GROUP	19
CREATING A USER	20
SETTING A USER'S AUTHORIZATIONS	21
4.7.2. RESTRICTING ACCESS	21
4.8. PRESENTING DATA	22
4.8.1. STRUCTURING WITH DATA TYPES	22
4.9. CONFIGURING ENTANDO	23
4.9.1. CATEGORIES	23
4.9.2. LABELS AND LANGUAGES	23
4.9.3. DATABASE	23
4.9.4. FILE BROWSER	23

1. ENTANDO 5.0.0 RELEASE NOTES

1.1. ABSTRACT

This document contains release notes for Entando 5.0.

1.2. PREFACE

These release notes highlight new features in Entando 5.0, list features that are in technical preview, and include any known issues.

1.3. GOAL OF ENTANDO 5

The goal of Entando 5.0 is to fully engage our platform with the Modern Applications revolution by designing for a cloud-based infrastructure, adapting our architecture to be more distributed, and fully supporting DevOps methodologies.

1.4. ABOUT ENTANDO 5

Entando 5 represents a fundamental update to the Entando infrastructure to support modern application development processes including: new options for clustering, cloud native deployment, out of the box support for deploying apps in OpenShift, new UI technologies, and an updated widget development process.

These release notes provide a technical introduction to the changes included in Entando 5. Check out our Installation guide, Getting Started guide, or our blog for examples and more on putting Entando 5 to work.

1.5. WHAT'S NEW WITH ENTANDO 5

- Infrastructure/Platform
 - Hybrid hosting environment
 - Entando 5 supports deployment in both public and private cloud configurations including deployment using Docker and OpenShift
 - Cloud native clustering using a JSR 107 compliant cache for scaling in the cloud
 - Entando 5 introduces the ability to back an installation of the MApp Engine-engine with a JSR 107 cache for multinode high availability (HA) configuration
 - A reference implementation using Infinispan is provided in the entando-components project
 - Container deployment
 - Deploy directly to OpenShift using images published to the Red Hat catalog
 - Deploy using Docker images provided on DockerHub

- Install an Entando project using provided Source to Image (S2I) Docker images
- New public facing REST APIs
 - Entando 5 exposes a new set of REST APIs for interacting with App Builder assets
- Decoupled application builder and micro engine for updated deployment options
 - This release decouples the application builder's front end from the Entando engine allowing the engine and App Builder to be deployed separately. The updated application builder is a ReactJS based application that uses REST APIs and can be deployed and managed separately from the MApp Engine
- Upgrade to Spring 5
- Security updates We have performed security scans as part of the Entando 5 software release to ensure that we are using secure libraries and practicing secure coding practices within our application. We continue to use automated tools and manual security reviews to monitor and evaluate the security of the Entando platform. Notable updates include:
 - Various library version upgrades to match OWASP security scan results at time of release
 - Software updates to support SonarQube security scans and improve software security
 - The Entando core has been updated to Spring 5.0.5
- Processes
 - Supports DevOps processes
 - First class support for deploying Entando projects on Docker and OpenShift
 - New archetypes to deploy to OpenShift and Docker using Fabric8
 - Created Docker base images for quick extension and deployment
 - Manage full CI/CD life cycle
 - Created reference implementation of Entando Docker images in Jenkins to create, deploy, and manage an application built using Entando in a CI/CD environment
 - Decoupled CMS from MApp Engine
 - The CMS capability available in Entando 4.3.2 has been decoupled from the core MApp Engine implementation and will be available as a plugin to Entando 5 in a future release
 - New component development experience
 - The Entando Component Generator (formerly edo) has been updated to use a configuration file to more easily select different types of components to generate

1.5.1. KNOWN ISSUES

- The ability to associate attributes with User Profiles is not implemented in the decoupled App Builder
 - Workaround: use the legacy application screens to add attributes to user profiles
 - Anticipated fix in first patch release
- Manually editing and saving Data Types and Data Models in the decoupled App Builder has errors

- Workaround: use the legacy application screens to edit Data Types
- Anticipated fix in future patch release
- If you move the homepage to a child of itself in the page tree it can result in browser and application stability problems
- Anticipated fix in future patch release
 - The PAM Datatable and Task List widgets only function in the legacy App Builder

1.5.2. VERSIONING INFORMATION

Table 1. Jars and artifacts included in this release:

Artifact	Version
entando-core	5.0.0
entando-archetypes	5.0.0
entando-ux-packages	5.0.0
entando-vfs-manager	5.0.0
app-builder	5.0.0
MApp-Engine-admin	5.0.0

1.5.3. SOFTWARE REQUIREMENTS

- Java 1.8+
- npm 5.6.0+ (for ReactJS front ends)
- Maven 3.0.5+
- Ant 1.8.0+

1.5.4. LEGAL NOTICE

- LGPL v2.1 License for the Core
- MIT License for the components

1.5.5. LINKS AND RESOURCES

- Explore the code on GitHub: <https://github.com/entando>
- See our Getting Started guide: <http://docs.entando.com/#getting-started>
- See our Custom App Building guide: http://docs.entando.com/#_custom_app_building_with_entando
- Contribute to the software: <https://central.entando.com/en/community.page>

2. INSTALLING ENTANDO APP BUILDER 5.0

2.1. PREREQUISITES

2.1.1. INSTALLED PREREQUISITES:

- Java 1.8+
- npm 5.6.0+ (for ReactJS front ends)
- Maven 3.0.5+ (including Maven Central)
- Ant 1.8.0+
- Git (to build from source)

2.1.2. SPACE AND HARDWARE REQUIREMENTS

Entando is built to be a very light framework requiring limited runtime resources. A deployment of Entando can have a footprint as small as 58 megabytes. When extending Entando in a development setting, sizing for development environments that add to an Entando install is up to individual user requirements.

2.2. CREATING A NEW ENTANDO PROJECT

You can set up an Entando application via a Maven archetype that will create the project and structure needed to develop and add to an application.

Steps:

1. Open your command line and navigate to the directory that will house your new project.
2. Run the following command:
`mvn archetype:generate -Dfilter=entando-archetype-webapp-generic`
3. Select Option 1, displayed as follows:
1: remote > org.entando.entando:entando-archetype-webapp-generic (Generic web app Archetype for Entando: an agile, modern and user-centric open source web app like platform.)
4. If prompted for a version, select 5.0.0.
5. Enter values for groupId, artifactId, version, and package. These values go into the Maven setup in the new project.
6. Open the directory created by the Maven archetype in Step 2. The directory will have the same name as the value you entered for artifactId in Step 5. This directory contains all of the structure necessary to start the Entando MApp-Engine and execute the application. See Launching the Application section below for instructions on starting the newly created project.

2.3. LAUNCHING THE ENTANDO APPLICATION

2.3.1. QUICK START USING DERBY AND JETTY

Once you have created an Entando Application, you can launch it using Jetty and an embedded Derby database.

To do so, run the following command inside your project:

```
mvn clean jetty:run
```

Once started the logs will print the following messages:

```
[INFO] Started SelectChannelConnector@0.0.0.0:8080
```

```
[INFO] Started Jetty Server
```

```
[INFO] Starting scanner at interval of 5 seconds
```

The logs will indicate that your application is running. Follow the steps in Launching the App Builder section to launch the App Builder.

2.4. SETTING UP A DATABASE:

You can configure a newly created Entando application to connect to a database as its backing store. Derby is the default configuration in archetype-created applications, but you can change the default to a different value. Open the Filter Properties file in `src/main/filters` and enter the appropriate environment and database configuration.

To connect the MApp Engine to a database server:

1. In your database server, create a user for the application.
2. Create two databases. Use strings to name the databases similar to:
`<your-project-name>Port`
`<your-project-name>Serv`
3. Give the user from Step 1 permission to create, read, and write.
4. Update the appropriate Filter Properties file in `src/main/filters` to use the configuration for the database properties. For example, on a macOS, you would update `+ filter-development-unix.properties`.
5. Set the user, database, and password for the values created in Steps 1 and 2.
6. Launch the application.

NOTE

When launching with the `mvn jetty:run` command, Jetty Entando will automatically create the table structure required to run the application. This can be used to instantiate an empty database when the target deployment is an app server such as JBoss or Tomcat.

2.5. DEPLOYING TO DOCKER USING S2I (SOURCE TO IMAGE)

2.5.1. PREREQUISITES:

- Installed and running docker instance
- S2I command installed (<https://github.com/openshift/source-to-image>)
- An Entando App created using the steps in Creating a New Entando Project can be run directly

in Docker using the Entando S2I images. Images are provided for Wildfly 11, JBoss EAP X, and Tomcat.

To use the Docker S2I images, you must have already set up a database. See the Setting up a Database section for details.

By default, the app deployed in Docker will connect to a Postgres database to persist resources created using the App Builder and by the MApp Engine. In the app created from the archetype, update the properties in: **<your application>/.s2i/environment** to point to the user and databases created in the Setting up a Database section.

After configuring the database:

1. Pull in the Docker image using the following command:

```
docker pull entando_wildfly_s2i_5.0.0
```

2. Build the image using S2I using the command to build and deploy a Docker app in Docker:

```
s2i build <path or URL of your project> entando_wildfly_s2i_5.0.0 <your image name>
```

Where:

- **<path or URL of your project>** is the path to your project or a URL to access the code. The build for this project will be invoked and the resulting war file deployed to the app server in the image
- **Entando_wildfly_s2i_5.0.0** is the name of the base docker image provided by Entando
- **<your image name>** is the name for this docker image

2.6. USING OPENSIFT

2.6.1. PREREQUISITE:

- Installed minishift environment

You can configure an Entando app to launch directly into OpenShift using a pre-configured archetype. Create an OpenShift-ready project using the following command:

```
mvn archetype:generate -Dfilter=entando-archetype-webapp-openshift
```

2.7. LAUNCHING THE APP BUILDER

2.7.1. BUILD FROM SOURCE

Prerequisites:

- git
- npm
- node

CLONE AND SET UP

Enter the following commands in your command line:

1. `git clone https://github.com/entando/frontend-common-components.git`
2. `cd frontend-common-components.git`
3. `npm install`

NOTE | The `npm install` command installs npm dependencies

DEPLOY

Enter the following commands in your command line:

1. `npm run lint`
Runs the linter. It fails if linting rules are not matched.
2. `npm run coverage`
Runs unit tests. It fails if a unit test fails, or if the minimum coverage threshold is not met.
3. `npm run import-plugins`
Compiles and imports Entando plugins.
4. `npm run build`
Compiles the project and creates the build directory.
5. `npm run build-full`
Runs `npm run lint`, `npm run coverage`, `npm run import-plugins` and `npm run build`

Development

`npm start`

Starts the application in dev mode (local web server).

Using Docker

1. Pull in the docker image:
`docker pull entando/app-builder-5.0.0`
2. Run the image. Example docker command:
`docker run -it --rm -d -p 5000:5000 -e DOMAIN=http://localhost:8000/my-app appbuilder-5.0.0`
Where:
 - **DOMAIN=** is the url of a running instance of the MApp Engine. The App Builder uses the REST APIs in the engine to create and manage the application

2.7.2. BUILD THE MAPP ENGINE FROM SOURCE

To download the latest source code:

1. Open your terminal and create an empty directory for your project:
`mkdir ~/my_new_project`
2. Move to the new directory

```
cd ~/my_new_project
```

3. Clone the following repositories IN ORDER: entando-core, entando-components, entando-archetypes, entando-ux-packages projects:
 - a. Entando-core:

```
git clone https://github.com/entando/entando-core
```
 - b. Entando-components:

```
git clone https://github.com/entando/entando-components
```
 - c. Entando-archetypes:

```
git clone https://github.com/entando/entando-archetypes
```
4. Install, IN ORDER, the entando-core, entando-components, entando-archetypes projects:
 - a. cd entando-core

```
mvn clean install -DskipTests
```
 - b. cd entando-components

```
mvn clean install -DskipTests
```
 - c. cd entando-archetypes

```
mvn clean install -DskipTests
```
5. Complete the download by following the steps from the Creating a New Entando Project section.

NOTE

The command to use the artifacts you have installed locally with an additional switch on the archetype command is:

```
mvn archetype:generate -Dfilter=entando-archetype-webapp-generic  
-DarchetypeCatalog=local
```

INTEGRATE THE BPM FORM

1. From the tabs along the top, choose Integrations > Red Hat BPM Config > Add Integration.

[custom integrations] | *custom_integrations.png*

2. In the Connection section, add all server information. For example:

- General Settings section:
 - **Active:** On
 - (Optional) **Debug:** On
- Connection section:
 - **Name:** Ansible Server
 - **Host name:** ansible.serv.run
 - **Schema:** http
 - **Port:** 8080
 - **Webapp name:** kie-server
 - **Username:** username for the process server administrator
 - **Password:** password for the process server administrator

- **Conn. Timeout (in millis):** 500

3. Click **Test Connection** to verify a valid the configuration, then click **Save**.
4. Open the Page Tree by choosing the Page Designer tab > Page Tree.
5. Find your new page in the Page Tree and click its **More Options** button > Configure.
6. Search for the “BPM-form by Data Type” widget and drag it to the Central Bar Center frame.
7. Click the **Preview** button to verify the correct appearance.
8. Click **Publish**.

3. USING ENTANDO APP MANAGER WITH THE RED HAT PROCESS APPLICATION MANAGER

3.1. INTRODUCTION

The "Entando_5.0.0_PAM_App_Manager" zip file provides an offline distribution of the Entando App Manager for supported Red Hat customers. The assets here provide all tools necessary for creating new Entando applications that are integrated with the Red Hat Process Automation Suite.

Major and minor versions of Entando releases are also available on Maven Central. Patch releases are only available to supported customers. The open source version of the software is available on GitHub at: <https://github.com/entando>.

You can find documentation, downloads, and more information at <https://central.entando.com/>.

3.2. QUICK START FROM ZIP

1. Go to <https://central.entando.com> and click the **Download** button in the Welcome To Entando Central > Download section.
2. In the "Download Licensed Entando Packages" section, click **Get This**. Acknowledge the End User License Agreement, click **Download**, and select a download location for the zip file "Entando_5.0.0_PAM_App_Manager."

PRODUCTS & DOWNLOADS

DISTRIBUTIONS

ENTERPRISE LICENSED

COMMUNITY
DISTRIBUTIONS

DOWNLOAD LICENSED ENTANDO PACKAGES

MApp Engine + App Builder + PAM Plugin + Example Project
Entando App Manager v5.0.0 for Red Hat PAM v7.0.0

GET THIS



3. Unzip Entando_5.0.0_PAM_App_Manager in a directory of your choosing (for example /opt/entando).
4. Open a command line in the unzipped folder and go to the Entando project directory by entering the following command:

```
cd entando-pam-sample
```

5. Enter the following command to start the Entando application:

```
mvn clean jetty:run -Djetty.port=8081
```

This process can take several minutes to complete and will seem to pause a few moments between some steps. The process is complete after the command line prints the following lines:

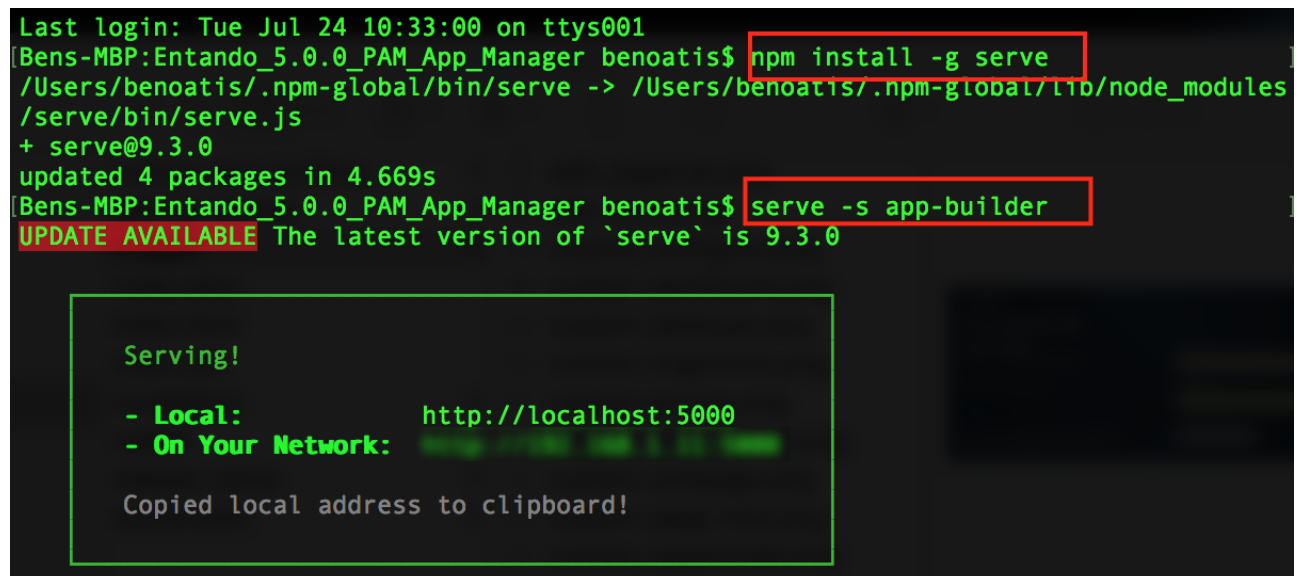
```
[INFO] Started Jetty Server
[INFO] Starting scanner at interval of 5 seconds
```

NOTE

The `Djetty.port=8081` argument removes conflicts between the MApp Engine and any other program running on port 8081, such as Wildfly or EAP.

6. On a new command line, start the App Builder by going to the Entando_5.0.0_PAM_App_Manager directory and running the following commands **as an administrator**:

```
npm install -g serve
serve -s app-builder
```



```
Last login: Tue Jul 24 10:33:00 on ttys001
[Bens-MBP:Entando_5.0.0_PAM_App_Manager benoatis$ npm install -g serve
/Users/benoatis/.npm-global/bin/serve -> /Users/benoatis/.npm-global/lib/node_modules
/serve/bin/serve.js
+ serve@9.3.0
updated 4 packages in 4.669s
[Bens-MBP:Entando_5.0.0_PAM_App_Manager benoatis$ serve -s app-builder
UPDATE AVAILABLE The latest version of `serve` is 9.3.0

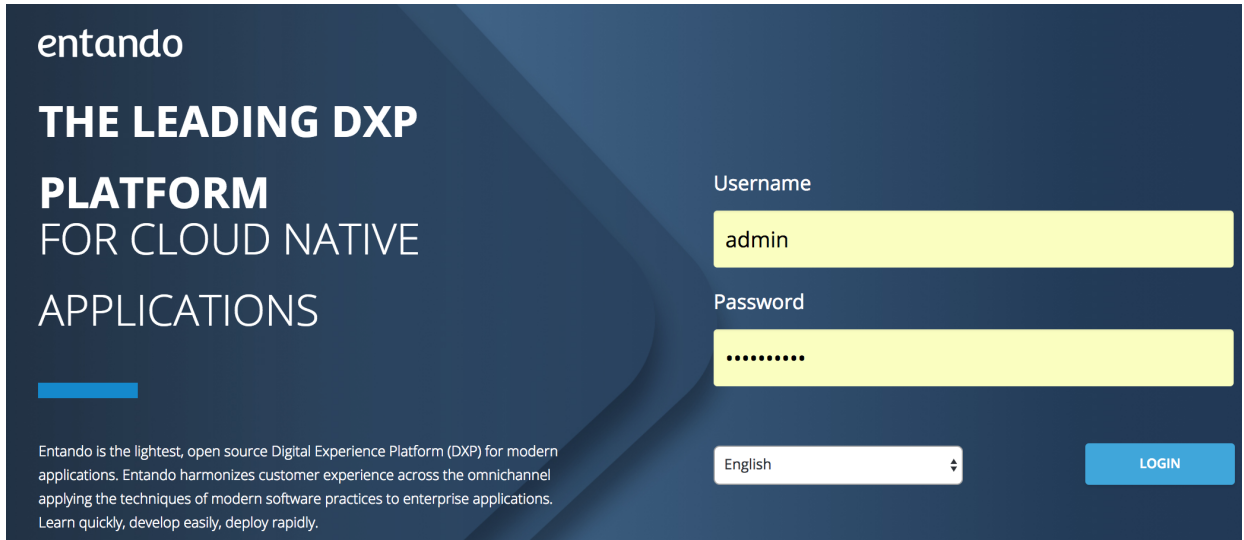
Serving!

- Local: http://localhost:5000
- On Your Network: http://192.168.1.100:5000

Copied local address to clipboard!
```

7. Wait thirty seconds for the App Builder to start, then open a browser and go to <http://localhost:5000>.
8. Log in to the App Builder with the following credentials:

- Username: admin
- Password: adminadmin

The image shows the Entando login interface. On the left, the 'entando' logo is at the top, followed by the text 'THE LEADING DXP PLATFORM FOR CLOUD NATIVE APPLICATIONS'. Below this is a blue horizontal bar and a paragraph of text: 'Entando is the lightest, open source Digital Experience Platform (DXP) for modern applications. Entando harmonizes customer experience across the omnichannel applying the techniques of modern software practices to enterprise applications. Learn quickly, develop easily, deploy rapidly.' On the right side, there is a login form with two input fields: 'Username' containing 'admin' and 'Password' containing a masked password '.....'. Below the password field is a language selector dropdown menu showing 'English'. A blue 'LOGIN' button is positioned to the right of the dropdown.

4. CUSTOM APP BUILDING WITH ENTANDO

4.1. INTRODUCING THE ENTANDO APP BUILDER

The App Builder is Entando's application to help you integrate, configure and customize any kind of modern web application, regardless of your programming proficiency. The App Builder provides an Admin Console, a visual environment, that collects elements for configuring your web application and managing your activities.

The App Builder is a ReactJS-based application that uses REST APIs to communicate with the Entando Core.

This document describes the elements and functionalities of the Admin Console.

4.1.1. ENTANDO ADMIN CONSOLE VERSION 5.0

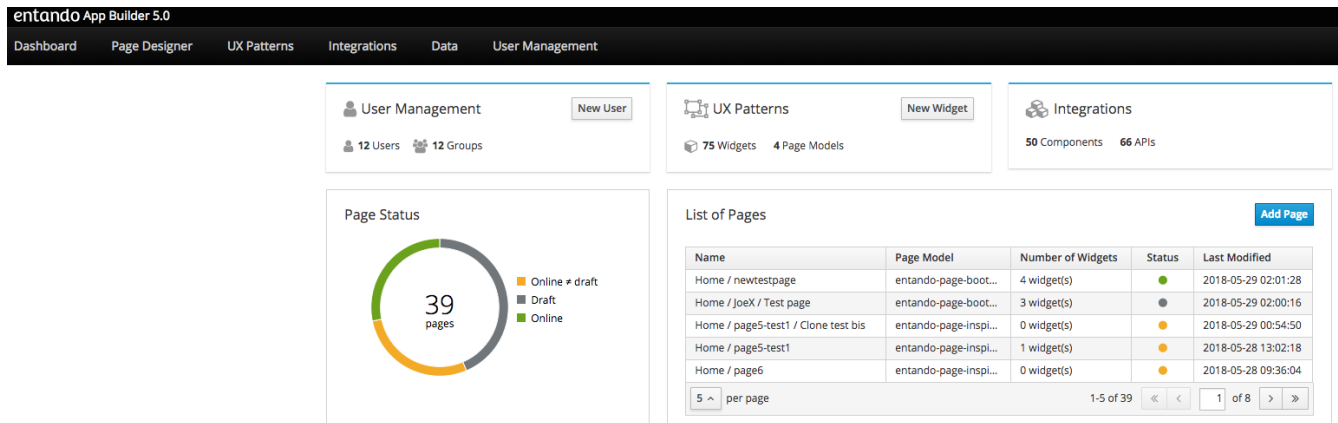
With Entando version 5.0 the Admin Console has been redesigned and renewed with the adoption of the PatternFly framework and guidelines. PatternFly is built on top of Bootstrap3 and provides an enterprise grade UX look and feel based on research and UX best practices.

4.2. APP BUILDER OVERVIEW

The Entando main view is the entry point for every administration activity. On the up side of the screen there is an horizontal menu, which gives access to all Entando's functionalities, while on the down-side of the screen, the Dashboard provides an overview of Entando status and a quick access to some functionalities.

4.3. DASHBOARD

The dashboard is divided into distinct areas.



Dashboard areas:

1. Page Status (gives an overview of the status of the pages: Online, Online /= Draft and Draft)
2. List of Pages (gives an overview of the list of all Pages: Name, configured Page Model, Number of configured Widgets, Status, Last Modified)
3. Add Page (possibility to add a new Page)
4. User Management (quick access to Users list and Group list)
5. Add User (possibility to add a new User)
6. UX Patterns (gives an overview of the number of configured Widgets and Page Models)
7. Add Widget (possibility to add a new Widget)
8. Integrations (gives an overview of the number of integrated components and APIs)

4.4. CREATING AND MANAGING PAGES WITH PAGE DESIGNER

The Page Designer section deals with all page management-related activities.

The Page Designer section offers the following subsections:

1. Page Tree
2. Page Configuration
3. Page Settings

4.4.1. USING THE PAGE TREE

The Page Tree subsection shows the hierarchical structure of the pages and their statuses, and lets you perform all page management actions.

Dashboard
Page Designer
UX Patterns
Integrations
Data
User Management
Configuration

Page Designer > Page Tree

Search

Page Code

Page Code

Search

Add

Page tree	Status	Displayed in menu	Actions
<div> <div>Home</div> </div>	●	Yes	⋮
<div> <div>page5-updated</div> </div>	●	Yes	⋮
<div> <div>Inspinia</div> </div>	●	Yes	⋮
<div> <div>EngTitle</div> </div>	●	Yes	⋮
<div> <div>Service</div> </div>	●	No	⋮
<div> <div>page9 - updated back</div> </div>	●	Yes	⋮
<div> <div>page6-updated</div> </div>	●	Yes	⋮

CREATING A NEW PAGE

To add a new page:

1. From the tabs along the time, choose the Page Designer > Page Tree > **Add**.
2. Enter a name in both the English and Italian fields.
3. (Optional) Enter a code. By default, Entando generates a code similar to the page title, but you can manually enter a different code. The field accepts alphanumeric characters and underscores, with a character limit of 30.

NOTE

Once you save the page, you will no longer be able to edit the page's Code field.

4. Select an Owner Group and a Join Group from the drop-down menus in the Page groups section. Owner Group members can administer the page. Join Group members can view the page.
5. Select from the Page Tree section the page or folder that will be the parent of the new page.
6. Select a Page Model from the Page Model drop-down menu. The Page Model determines the layout of your new page. See the [Using Page Models](#) section below.
7. Set additional options from the Settings section as needed.
 - **Displayed in menu:** toggles the page's presence in the menu on or off.
 - **SEO:** sets toggles on or off optimization of the page for Search Engine Optimization (SEO)
 - **Charset:** specifies the character encoding to use. Selectable between utf8 and iso-88591
 - **Mime Type:** identifies the type of content on the page
8. After you have made your selections, click **Save and Configure** to open the Configuration section. Learn more in the [Configuring a Page](#) section below.

Add

Info

* Required fields

en Title *	<input type="text" value="English title"/>
it Title *	<input type="text" value="Italian title"/>
Code *	<input type="text" value="Code"/>
Select page placement *	<div>Page tree</div> <div>> Home</div>

Page groups

* Required fields

Owner Group *	<input type="text" value="Choose an option"/>
Join Group *	<input type="text" value="Choose an option"/>

To edit an existing page:

1. Open the Page Tree and find the page to edit.
2. Click the **Actions** (three vertical dots) button in line with the page and select "Edit."

4.4.2. CONFIGURING A PAGE

The Page Configuration section is where you assign widgets to the frames defined by the Page Model you applied to your page. The available widgets are in the right pane. Apply widgets by dragging and dropping them to a frame of your choice.

This is also where you publish or unpublish the page or preview the layout prior to publishing.

Click the **Info** button for a summary of options you assigned when creating the page, e.g., Title and Owner Group.

Integrated Delivery

Info

Preview

Restore

Unpublish

Publish

On-the-fly page

No

Default widget applied

Sidebar 1XYZ	Top Bar 1	Top Bar 2	Top Bar 3	Top Bar 4
Sidebar 2	Left	Center	Right	
Sidebar 3	Full 1			
Sidebar 4	Full 2	Main Frame		
Sidebar 5	Content left	Content right		

Pages

WIDGETS

Search

User

+

Lorem ipsum is simply dummy text of the printing and typesetting industry changed by Selenium

+

News - Latest News

+

title

Kiebpmp

+

BPM Case actions

+











BPM Case chart

4.5. UX PATTERNS: MANAGING WIDGETS, PAGE MODELS, AND FRAGMENTS

The UX Patterns section provides settings options for your general UX Patterns.

4.5.1. USING WIDGETS

A Widget is an independent software component configured and executed within your application's page. Widgets provide UX/UI blocks that deliver a specific function.

Dashboard Page Designer UX Patterns Integrations Data User Management Configuration			
Widget ?			
New			
user			
Name	Code	used	Actions
 Lorem Ipsum is simply dummy text of the printing and typesetting industry changed by Selenium	Lorem Ipsum is simply dummy text	2	
 News - Latest News	NWS_Latest	3	
 title	abc	5	
jpkiebpm			
Name	Code	used	Actions
 BPM-Case actions	bpm-case-actions	3	
 BPM-Case chart	bpm-case-chart	3	

The following are the categories of widgets:







- **User Widgets:** user-generated widgets created via the Admin Console
- **Custom Widgets:** widgets designed to meet specific applications needs such as Red Hat PAM widgets
- **Stock Widgets:** default widgets provided by the App Builder to perform common tasks
- **Specific Application Widgets:** widgets that apply only to a specific tool in Entando, e.g., the installation of CMS module provides Content Management related widgets
- **Plugins Widgets:** widgets provided by the installed plugins

NOTE

Each widget's Action menu differs depending on the type, including no available actions at all.

4.5.2. USING FRAGMENTS

Fragments are single portions of front-end interface used in widgets (including plugins) and Page Models. They are are HTML blocks that include Freemarker instructions by using specific tags that you replace with appropriate values each time you use the fragment. Entando then renders the widgets using the fragments. The FRAGMENTS section lets you manage existing fragments or add new ones.

Dashboard Page Designer UX Patterns Integrations Data User Management Configuration			
UX Patterns > Fragments			
Fragments ?			
List Settings			
<div> <div>Search</div> <div> <div>Code</div> <div>Code</div> </div> <div> <div>Widget Type</div> <div>All</div> </div> <div> <div>Plugin</div> <div>All</div> </div> <div>Search</div> </div>			
New			
Code	Widget Type	Plugin	Actions
-			
collaboration_entryidea		jpcollaboration	
collaboration_fastEntryidea		jpcollaboration	
collaboration_idea_find		jpcollaboration	
collaboration_idea_tags		jpcollaboration	
collaboration_statistics		jpcollaboration	

ADDING A NEW FRAGMENT

To add a new fragment:

1. From the tabs along the top, choose the UX Patterns tab > Fragments > **New**.

2. Enter a Code value. This value represents the fragment's unique identifier and appears in the Name column of the Fragments table.

NOTE Once you save the fragment, you will no longer be able to edit its Code field.

4.5.3. USING PAGE MODELS

The Page Model is the look and feel of an application's page. It defines the structure of the page using frames or areas of the page that contain widgets.

Page models have two configuration elements:

- **JSON Configuration:** formatted text that defines the frame structure of the page.
- **Template:** the look and feel and the real layout of the page. In other words, the template defines where to show a frame and how it will be rendered. Templates use Velocity Template Language (VTL) to provide a simple and clean way to incorporate dynamic content into a web page. It supports statements and directives, references and comments, access to exposed java objects, and methods.

CREATING A PAGE MODEL

To create a new page model:

1. From the tabs along the top, select UX Patterns > Page Models > **Add**.
2. Assign the page model a code and a name. Codes represent the page model's unique identifier. The name can be more human readable. For example:
Code: *entando-page-purple*
Name: *Purple - BPM*
3. In the JSON Configuration field, enter JSON content to structure the page. For example:

```
{
  "frames": [
    {
      "pos": 0,
      "descr": "Top bar 1",
      "mainFrame": false,
      "defaultWidget": null,
      "sketch": {
        "x1": 0,
        "y1": 0,
        "x2": 2,
        "y2": 0
      }
    }
  ]
}
```

NOTE The JSON content must be a single object defining "frames." In other words, the content you enter must go within the `frames: []` object.

See the Getting Started with Entando App Builder document for a complete example.

4. In the Template field, enter XML formatting to determine where each defined frame belongs and how to render it. See for a complete example.
5. When finished, click **Save**.

DashboardPage DesignerUX PatternsIntegrationsDataUser ManagementConfiguration

Page Designer > Page Models > Edit

Edit

Code * ⓘentando-page-bootstrap-hero

Name * ⓘBootstrap - Hero Unit

JSON Configuration *

```
1 {
2   "frames": {
3     {
4       "pos": 0,
5       "descr": "Navbar 1",
6       "mainFrame": false,
7       "defaultWidget": {
8         "code": "entando-widget-language_choose",
9         "properties": null
10      },
11       "sketch": {
12         "x1": 0,
13         "y1": 0,
14         "x2": 11,
15         "y2": 0
16      }
17     }
18   }
19 }
```

Template *

```
1 <#assign wp=JspTaglib["/ape-core"]>
2 <#assign c=JspTaglib["http://java.sun.com/jsp/jstl/core"]>
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6   <meta charset="utf-8" />
7   <title>
8     <#wp.currentPage param="title" /> - <#wp.i18n key="PORTAL_TITLE" />
9   </title>
10  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
11  <meta name="description" content="" />
12  <meta name="author" content="" />
13  <link rel="icon" href="#<#wp.info key="systemParam" paramName="applicationBaseURL" />favicon.png" type="image/png" />
14  <!-- Le W3C's shim, for IE6-8 support of HTML5 elements -->
15  <!--[if lt IE 9]>
```

4.6. INTEGRATING WITH RED HAT PROCESS AUTOMATION MANAGER (PAM)

Entando integrates with Red Hat's Process Automation Manager (formerly Red Hat JBoss BPM Suite), a middleware platform for creating cloud-native business automation applications and microservices. It enables enterprise business and IT users to document, simulate, manage, automate and monitor business processes and policies. It is designed to empower business and IT users to collaborate more effectively, so business applications can be changed more easily and quickly.

DashboardPage DesignerUX PatternsIntegrationsDataUser ManagementConfiguration

Red Hat PAM Config

Test AllAdd Configuration

SERVER

Ansible server

Active ✓

Debug Disabled

Version

Address `http://ansible.serv.run:8080/zie-server`

Capabilities

KieServer BRM BPM BPM-LU BRP

To integrate with Red Hat PAM:

1. From the tabs along the top, choose Integrations > Red Hat BPM Config > **Add Integration**.
2. In the Connection section, add all server information. For example:

17

- General Settings section:
 - **Active:** On
 - (Optional) **Debug:** On
- Connection section:
 - **Name:** Ansible Server
 - **Host name:** ansible.serv.run
 - **Schema:** http
 - **Port:** 8080
 - **Webapp name:** kie-server
 - **Username:** username for the process server administrator
 - **Password:** password for the process server administrator
 - **Conn. Timeout (in millis):** 500

3. Click **Test Connection** to verify a valid the configuration, then click **Save**.
4. Open the Page Tree by choosing the Page Designer tab > Page Tree.
5. Find your new page in the Page Tree and click its **More Options** button > Configure.
6. In the right pane, search for the “BPM-form by Data Type” widget and drag it to a frame established by your page model.
7. Click the **Preview** button to verify the correct appearance.
8. Click **Publish** once your page is complete.

4.7. MANAGING USERS

The Entando App Builder assigns access rights using combinations of roles and groups. You can manage all activities and privileges for roles and groups through the User Management tab.

User management definitions:

- **Users:** perform operations and access information. Users are characterized by a Profile Type that identifies the individual, such as name and email address. The actions that a user can take in an Entando program derive from combinations of Roles and Groups.
- **Roles:** assigned sets of permissions, e.g., content editing rights, or user profile editing rights. Roles are mutually independent, meaning one user can be associated with multiple roles.
- **Groups:** assigned sets of users, with each user given a role within the group. One user can belong to multiple groups. For example, Lynne is a member of the Content Editing group and has the role of Administrator, giving her the ability to perform any task on pages that the Content Editing group is assigned to. She’s also a member of the Newsletter group with the role of Editor, meaning she can edit content, but does not have full access to the pages the Newsletter group is assigned to.
- **User Restrictions:** perform access restrictions on the users, e.g., months availability since the last access, or months validity of the user password, etc.

- **Profiles Types:** perform different user's profile types and creation of the new profile types.

TIP

For first time projects, consider creating profiles from largest to smallest. For example:

1. **Groups:** are containers of users. Groups grant or restrict access to information and services independently from the Roles of the single Users. For example, all members of the Page Editors group would have access to pages assigned to Page Editors
2. **Roles:** assign responsibilities within the group, e.g., group Administrators of the Page Editors group have full administrative access to the page, while Newsletter Managers in the Page Editors group can only manage newsletter functions
3. **Users:** represent an entity with the ability to perform operations and access information. The actions allowed for a certain user are given by the combination of the Roles and Groups.

4.7.1. WORKING WITH USERS

CREATING A ROLE

To create a new role:

1. From the tabs along the top, select User Management > Roles > **Add**.
2. Enter a name.
3. (Optional) Enter a code. The code is the Role's unique identifier. By default, Entando generates a code similar to the page title, but you can manually enter a different code. The field accepts alphanumeric characters and underscores, with a character limit of 30.

NOTE

Once you save the role, you will no longer be able to edit the page's Code field.

4. In the Permissions section, toggle to "On" the access rights all users assigned to this role will possess. For example, public users may have no options toggled while power users may have the "All Permissions" option toggled to On.
5. Click **Save**. To edit a Role after saving, return to the Roles screen. In the Roles table, select the Actions menu (three vertical dots) for the role to change and select "Edit."

Dashboard Page Designer UX Patterns Integrations Data User Management Configuration

UX Patterns > Roles > Edit

Edit

* Required Fields

Name * backoffice

Code * backoffice

Permissions

Content Editing	OFF	User Profile Editing	OFF	User Editing	ON
Access to Administration Area	ON	Comment Editing	OFF	Comment Moderate	OFF
Rating Editing	OFF	Newsletter Configuration	OFF	Newsletter management	OFF
Gestione Web Dynamic Forms	OFF				

Save

CREATING A GROUP

To create a new group:

1. From the tabs along the top, select User Management > Groups > **Add**.
2. Enter a name.
3. (Optional) Enter a code. The code is the Group's unique identifier. By default, Entando generates a code similar to the page title, but you can manually enter a different code. The field accepts alphanumeric characters and underscores, with a character limit of 30.

NOTE

Once you save the group, you will no longer be able to edit the group's Code field.

To edit a Group after saving, return to the Groups screen. In the Groups table, select the Actions menu (three vertical dots) for the group to change and select "Edit."

Dashboard

Page Designer

UX Patterns

Integrations

Data

User Management

Configuration

User Management > Groups

Groups

?

Add

Name	Code	Actions
Administrators	administrators	<div>⋮</div>
BenTest	benetest	<div>⋮</div>
Bpm Admin	bpm_admin	<div>⋮</div>
Bpm Appraiser	bpm_appraiser	<div>⋮</div>
Bpm Broker	bpm_broker	<div>⋮</div>
Bpm Manager	bpm_manager	<div>⋮</div>
cicci	cicci	<div>⋮</div>

CREATING A USER

To create a new user:

1. From the tabs along the top, choose User Management > Users > **Add**.
2. In the Username field, enter

NOTE

The username must be between 8-20 characters. You can use numbers, underscores, special characters, and upper or lowercase letters.

3. Enter a password.
4. From the Profile Type drop-down menu, select "Default User Profile."
5. Toggle the Status to "On" to ensure that individuals can sign in with this user's credentials.
6. Click **Save** to create the user and return to the Users screen.

Dashboard
Page Designer
UX Patterns
Integrations
Data
User Management
Configuration

User Management > Users

Users

Search

Username

Users

All

User with a profile

User without a profile

Search

Add

Username	Full Name	Email	Status	Actions
admin			✓ Active	⋮
a_new_one			✗ Not active	⋮
davide			✓ Active	⋮
LenderBen2			✓ Active	⋮
Supervisor			✓ Active	⋮
testtest_1	test test	test@test.it	✓ Active	⋮
testtest_2			✓ Active	⋮

SETTING A USER'S AUTHORIZATIONS

- From the table on the Users screen, find your new user profile and click **More Options** > Manage authorizations for: *your profile name*. Set the following options:
 - User Group:** select a group to assign this user to. You can assign a user to as many groups as you want.
 - User Role:** select the role this user should have in the group. You can only assign a user to one role per user group.

The User Group/Role pairing will appear in the Authorizations table at the top of the screen.

- Click **Add** and then click **Save**.

4.7.2. RESTRICTING ACCESS

Entando lets you set restrictions on users (see Figure below), specifically a shelf life for passwords and a maximum time between log ins before the password expires.

Setting user restrictions:

- From the tabs along the top, choose User Management > User Restrictions.
- Toggle the Password Always Active option.
 - On - the password never expires and users have unlimited time between logins
 - Off - activates the password limitation fields, allowing you to set expirations
- Toggle the Enable Gravatar integration (Avatar of users) option.
 - On - enables management of the user's avatar
 - Off - disables Gravatar integration

DashboardPage DesignerUX PatternsIntegrationsDataUser Management

UsersRolesGroupsProfile typesUser Restrictions

User Management > User Restrictions

User Restrictions

Password Expiration

Password Always Active

ON

Number of months the password is valid for, after last access

0

month(s)

Number of months the password is valid for

0

month(s)

Avatar

Enable Gravatar Integration (Avatar of users)

ON

4.8. PRESENTING DATA

Data Types are collections of attributes. They are used to gather and manage data coming from external sources, like APIs or the IoT. This abstraction layer allows Entando to perform core features like searching on external data.

Data Models are used to agnostically render data coming from data types, ensuring that the same data can be represented differently, such as with forms, tables, or graphs. By separating the definition and presentation of data, the Entando App Builder gives you the ability to display the same data in a context- and user-aware way without requiring a data source or integration.

To manage Data Types:

1. Create a Data Type.
2. Create one or more Data Models per Data Type.
3. Publish the data on web pages with one or more widgets. The Data section allows you to manage the Data.

4.8.1. STRUCTURING WITH DATA TYPES

The Data Types section lists all available Data Types, their attributes, and is where you can create a new data structure.

Dashboard Page Designer UX Patterns Integrations Data User Management Configuration			
Data > Data Types			
Data Type ?			
New			
Name	Code	Status	Actions
Doc Demo 1	DMO		i
Data Type 1	DTI		i
Doc Demo 1	Dmo		i
fff	FFF		i
Venerdi	VEN		i
whatever	ccc		i
data model 5	dmx		i
10 ^ per page 1-7 of 7 << < 1 of 1 > >>			

To create a new data type:

1. From the tabs along the top, choose Data > Data Type > **New**.
2. Enter a Code. The code must be three letters.
3. Enter a Name.
4. Click **Save**. This opens the Attributes section where you specify OGNL validation.

4.9. CONFIGURING ENTANDO

The Configuration section (far right of the toolbar, gives you access to Entando general duty functionalities, such as configuring Categories, Labels, Configuration Reload, Database backup, file browsing and groups management.

4.9.1. CATEGORIES

Categories are dividers and parent folders for pages in your Entando page tree. Add a new category by clicking the **Add** button.

4.9.2. LABELS AND LANGUAGES

The Languages tab displays a list of all configured languages in the system. You can add as many different languages to the system as you need by selecting a language from the Select Language drop down menu and clicking **Add**. You can then include the appropriate vocabulary for any term in the System Labels section.

The System Labels tab displays system labels that apply to the entire Entando application. By default, system labels are available in English and Italian. If you add additional languages in the Languages tab, you can enter the appropriate word for a label in your choice language.

4.9.3. DATABASE

The Database section lets you back up your entire Entando's database by clicking the **Create Backup** button. The backed up files become available from the Protected folder of the File Browser section once the backup process completes.

4.9.4. FILE BROWSER

The File Browser section lets you browse files, upload, create text files and folders, and access database backups from two distinct areas of the file-system:

- **Public:** /project_name/src/main/webapp/resources
- **Protected:** /project_name/src/main/webapp/protected

NOTE | Backed up database files are in the Protected section by default.