



PORTAL FACTORY 1.0

Esigate Module Documentation

Rooted in Open Source CMS, Jahia's Digital Industrialization paradigm is about streamlining Enterprise digital projects across channels to truly control time-to-market and TCO, project after project.

Jahia Solutions Group SA

9 route des Jeunes,
CH-1227 Les acacias
Geneva, Switzerland

<http://www.jahia.com>

Summary

- 1 Overview.....3
 - 1.1 About Esigate3
 - 1.2 About this module.....3
- 2 Installation and setup4
 - 2.1 Requirements4
 - 2.2 Installation on Digital Factory4
 - 2.2.1 Method 1 (a link to internet is mandatory).....4
 - 2.2.2 Method 2 (this can be done even if the Digital Factory server is not linked to internet) ..5
- 3 Esigate usages and principles.....6
 - 3.1 Esigate overview6
- 4 Module configuration8
- 5 Aggregate an external application10
 - 5.1 Using Esi tags in the external application.....10
 - 5.2 Without changes in the external application10
- 6 Step by step12
 - 6.1 Declaring the provider12
 - 6.2 Inserting the application into a web page14
 - 6.3 Direct links to aggregated apps.....15

1 Overview

1.1 About Esigate

Esigate is an Open Source framework that aims to aggregate content or applications from various sources in a non-intrusive fashion at server side level. This approach is very complementary to Digital Factory's content-based and very granular model. To know more about Esigate, please read <http://www.esigate.org/>

1.2 About this module

The Esigate module is part of the [Portal Factory offer](#) and is available and supported for Digital Factory Enterprise distribution only.

This module provides an implementation of the Esigate proxy filter, which is a servlet filter. This filter, using Esigate technology, allows to map external applications and use ESI includes to aggregate them in Digital Factory pages. This permits a seamless integration whatever the underlying technology stack.

2 Installation and setup

2.1 Requirements

- An installed version of Digital factory 7.001
- A web project created on the platform
- The Portal Factory Package (<http://store.jahia.com/contents/modules-repository/packages/Portal.html>)

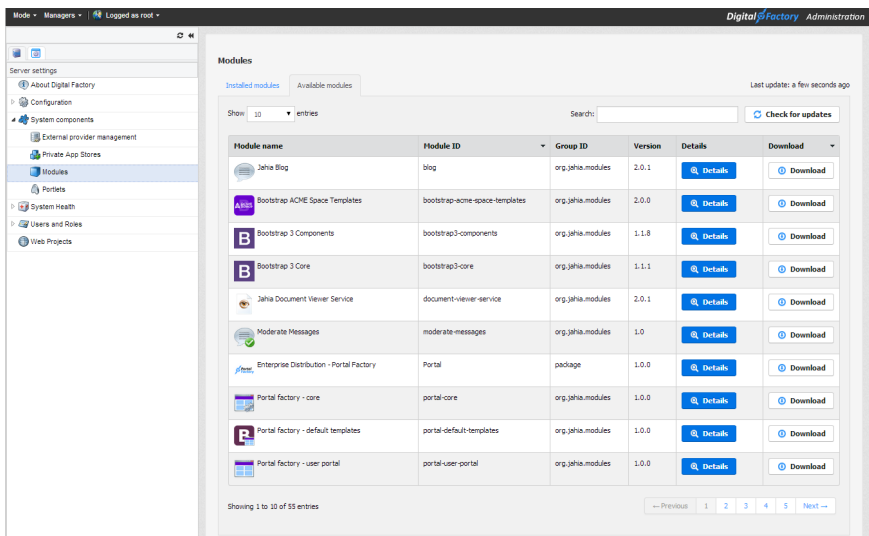
2.2 Installation on Digital Factory

The Esigate module is part of the Portal-Factory package. This package can only be installed on an Enterprise Distribution of Digital factory 7.001 and is supported by Jahia through a specific offer.

Installing the Esigate module is straight-forward and can be achieved in several ways

2.2.1 Method 1 (a link to internet is mandatory)

Logon to your Digital Factory platform; go into *Administration > System Components > Modules > Available modules*. Find the Portal factory entry in the list and click on the download button.



Or

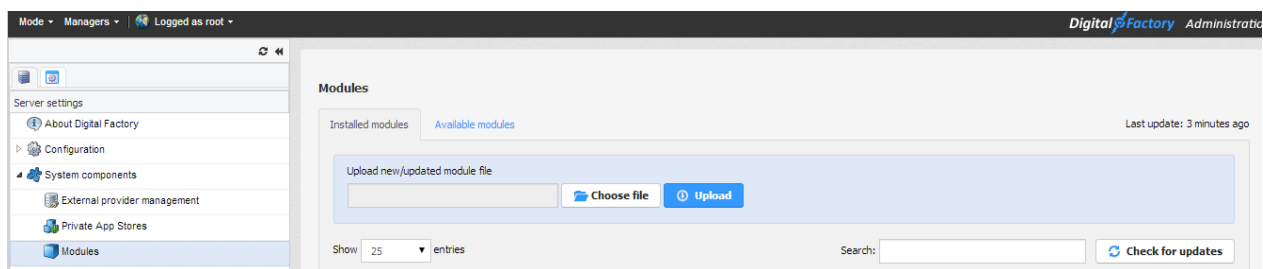
2.2.2 Method 2 (choose this method if the Digital Factory server does not have an internet connection)

From a connected computer (not the Digital Factory server if it doesn't have an internet connection), navigate to the Jahia Public App Store (<http://store.jahia.com/home.html>) and browse the repository to find the desired module.

Download the Portal Factory Package (jar file)

Go onto your Digital Factory platform > *Administration* > *System components* > *Modules*

Upload the jar file (transferred from your connected computer to the disconnected server) using the file upload form (top of the page), which is similar to copy the file manually in *WEB-INF/var/modules* but more convenient.



3 Esigate usages and principles

3.1 Esigate overview

Esigate acts as a proxy server by retrieving items contained between ESI tags coming from different sources to aggregate them into one new single html page

To do that, Esigate needs

- An HTML page that contains an esi:fragment tag (the fragment is identified by a name)
- An application that contains an esi:include tag to pull the page containing the fragment
 - o this esi:include need to contain it self a esi:replace tag (this tag targets the fragment by specifying its name)

Esigate insert the html content contained in the esi:replace (the application) into the esi:fragment (the page) and “decorates” the application with the html page to, finally, deliver a new page with its own URL.

Aggregating simple html content items into a Digital Factory page is something that is already possible in various ways pretty easily without Esigate then the Esigate framework has not been integrated for that particular purpose. What is possible with Esigate is to proxyify not only static HTML but dynamic HTML coming from an application and use that application as if it was called directly, but nicely revamped in another context.

The role of Esigate is to aggregate all the items, rewrite on the fly all the links and paths if necessary, then serve result to the users, becoming the new frontal application.

Let’s take an example to make things more clear. Imagine that

- You have a Digital Factory web site (then in java / jsp) with a fresh nice look and feel
- You have an old but very popular PHP forum on another server, with its own look and feel, etc.
- You don’t want to stop your old forum that works well and start with a new one in java, you don’t want to change the users authentication system, recreate or move users accounts, you don’t even want or can touch anything in that good old forum application not even the css style sheets, you just want to continue to use it, but revamped with the look and feel of

your new website and display it in a page with the new look and feel and managed in Digital Factory so you authors can use DF to update it, etc.

Then using Esigate can help you by

- Declaring an Esigate provider
- Pulling the forum on one side
- Pulling a Digital Factory page on the other side and use it as a kind of template
- Mixing both into a new full html page accessible to your users.

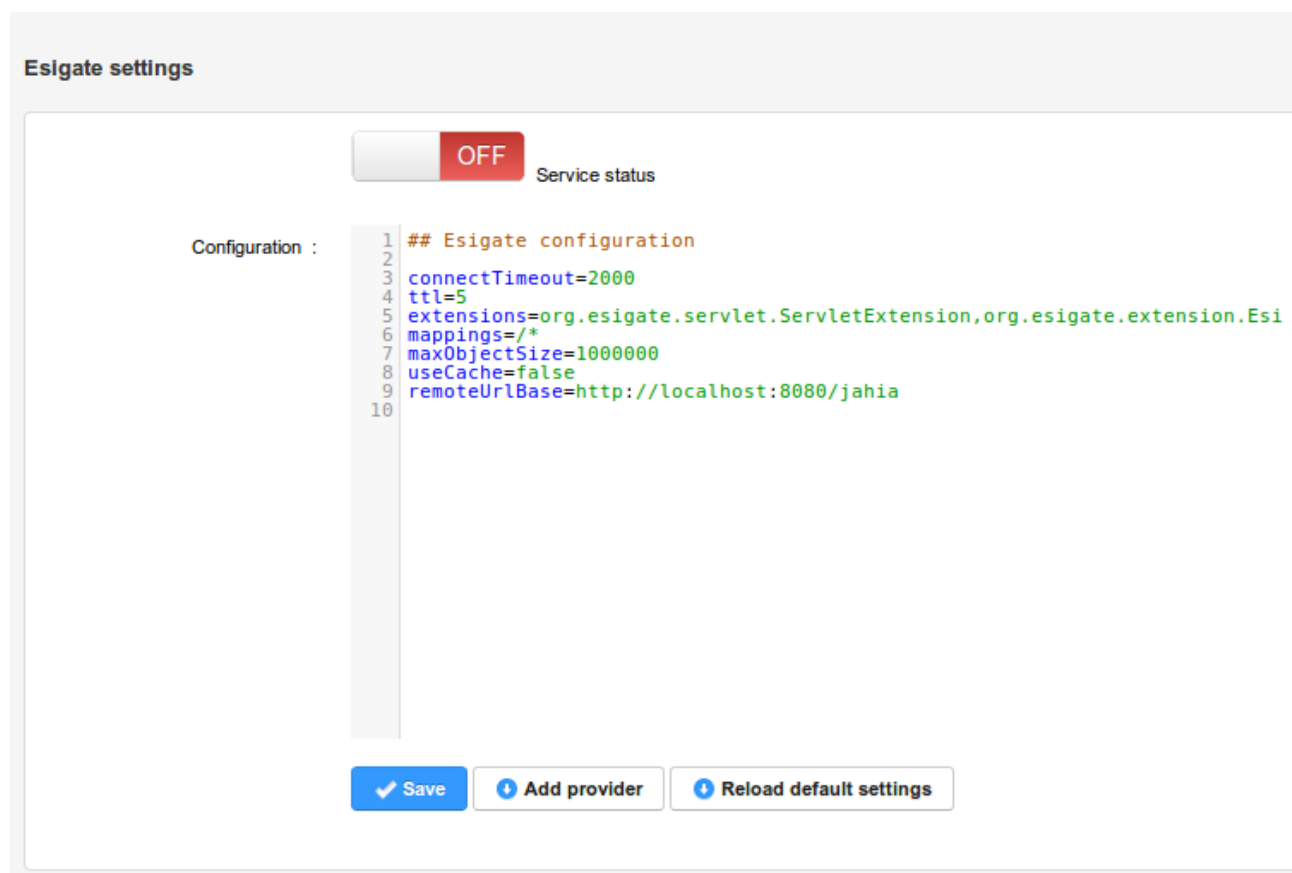
4 Module configuration

Once the module is installed, a new entry appears in *Administration > Server Settings*.

This entry is named *Portal Factory* and contains the different administration panels relative to the Portal Factory components.

Click on *> Esigate settings*

The following panel appears



The screenshot shows the 'Esigate settings' panel. At the top, there is a 'Service status' toggle switch currently set to 'OFF'. Below this is a 'Configuration' section with a text area containing the following properties:

```
1 ## Esigate configuration
2
3 connectTimeout=2000
4 ttl=5
5 extensions=org.esigate.servlet.ServletExtension,org.esigate.extension.Esi
6 mappings=/*
7 maxObjectSize=1000000
8 useCache=false
9 remoteUrlBase=http://localhost:8080/jahia
10
```

At the bottom of the configuration area, there are three buttons: 'Save' (with a checkmark icon), 'Add provider' (with a plus icon), and 'Reload default settings' (with a refresh icon).

In this panel you can:

- **Start/stop** the filter (top button)
- Edit manually the detailed configuration of the ESI Filter. The Esigate configuration parameters are documented in the Esigate documentation (<http://www.esigate.org/reference.html#esigate.properties>). There are no specifics or restrictions due to Jahia Digital Factory. In order to be able to use Esigate immediately, a

minimal configuration is generated when the module is deployed. A default provider is automatically created regarding to your context on the Digital Factory application. Note that no input controls are performed when the configuration file is edited manually.

- Add providers in the configuration file using a form, by clicking the “Add provider” button.
- Restore default configuration (this will remove all providers declared or other changes made manually) by clicking on the “Reload default settings” button.

5 Aggregate an external application

There are two methods to aggregate an application with Esigate

5.1 Using Esi tags in the external application

The first method consists to add

- Add Esi tags (`esi:include` and `esi:replace`) in the HTML of your external application to specify what exact part(s) of the HTML generated by the application will be aggregated and in which fragment list. To tag your old application, please refer to the Esigate documentation

In that case, all you have to do at the Digital factory level is to

- Declare the application as a provider (see below)
- Insert an Esi component in the Digital factory page that will be used to embed the app (the Esi component is just a classic content list surround with an `esi:fragment` tag taking the system name of the component as fragment name)
- Insert a link in your site that will lead to the aggregated page delivered by Esigate.

This method is interesting if you don't want to clip the entire application for instance, and just a part of the html that is generated (for instance to remove old headers and footers, you'll just put esi tags around the body of your application).

5.2 Without modifying the external application

The second method does not require modifications on the external application, not even adding esi tags. You just have to:

- Declare the application as a provider
- Set a default page include (the page that will be used as "decorator")
- Set a default Esi fragment name (where the external application will appear in the page)
- Add an Esi component in the Digital factory Page that will be used to embed the app reusing the fragment name as component name. (the Esi component is just a classic

content list surround with an esi:fragment tag taking the system name of the component as fragment name)

- Insert a link in your site that will lead to the aggregated page delivered by Esigate.

This method requires less work and applies to apps that are completely external, even the ones on which you have no control or where you can't change the html source code for any reason. The counterpart is that the complete application will be aggregated by Esigate, including headers, footers or side columns. In other words, it's perfect for small applications dedicated to one specific purpose and that does not come with a full and complex look and feel or page structure. In that case you'll be up and running in minutes.

6 Step by step

For the purpose of the explanation we will use a very simple webapp, deployed in the same tomcat under the context /sample-contact-webapp that we will integrate into a Digital Factory page through Esigate.

The webapp can be retrieved at: <https://github.com/jkevan/sample-contact-webapp> and built with maven (mvn install) if you want to repeat the process.

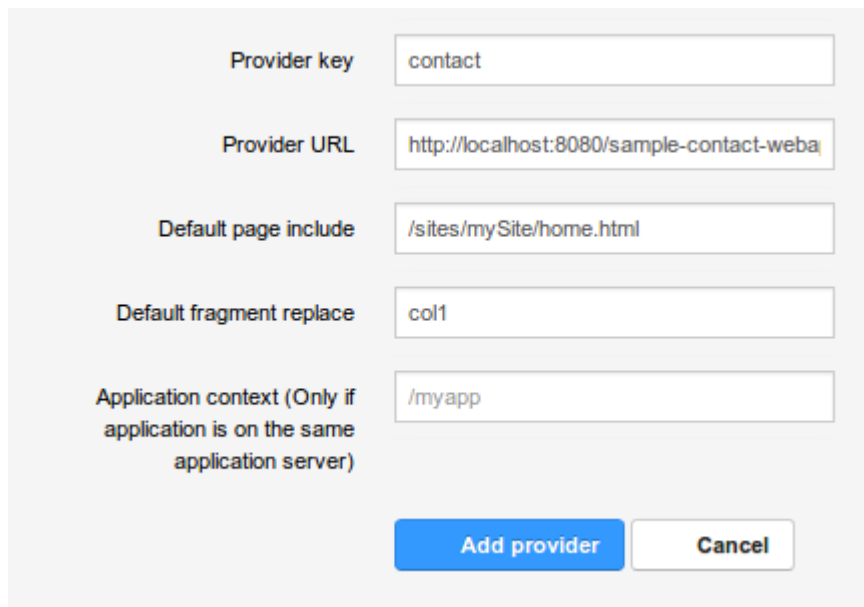
6.1 Declaring the provider

The first thing to do is to declare the application as a provider. The parameters you have to specify depends if you're aggregating an application containing Esi Tags or not.

In *Administration > System Components > Portal Factory > Esigate settings*

Click on the button **"Add provider"**

The following form is displayed



Provider key	<input type="text" value="contact"/>
Provider URL	<input type="text" value="http://localhost:8080/sample-contact-webapp"/>
Default page include	<input type="text" value="/sites/mySite/home.html"/>
Default fragment replace	<input type="text" value="col1"/>
Application context (Only if application is on the same application server)	<input type="text" value="/myapp"/>

Provider key: the provider represents the name of the provider, you can't use a key already used by another provider. The key is also used to generate the mapping for the provider. Here the generate

mapping will be /ext/contact/. All the external providers start with /ext because the Jahia extension makes operations in the Esigate flow related to application bind on /ext/{key}/*

Provider URL: it's the URL of your external app, here the application is deployed in tomcat under **/sample-contact-webapp**

It is mandatory that the URL finishes with a /

Then the value to input is /sample-contact-webapp/

Default page include: this parameter is used only when the aggregated application doesn't contain **esi:include** tag to explicitly specify the Digital Factory page that has to be used to decorate the external application.

Default fragment replace: the default fragment to be replaced by the aggregated application. This parameter works with the default page include and does not need to be specified if the external application contains **Esi tags already**. You can specify here the name of the "ESI fragment list" in your default page include.

Application context: the application context can be specified **only** when the external app runs under the same application server. This will force Esigate to use tomcat internal cross-context calls. Note that cross-context must be activated on Digital Factory (in context.xml file, <Context crossContext="true">

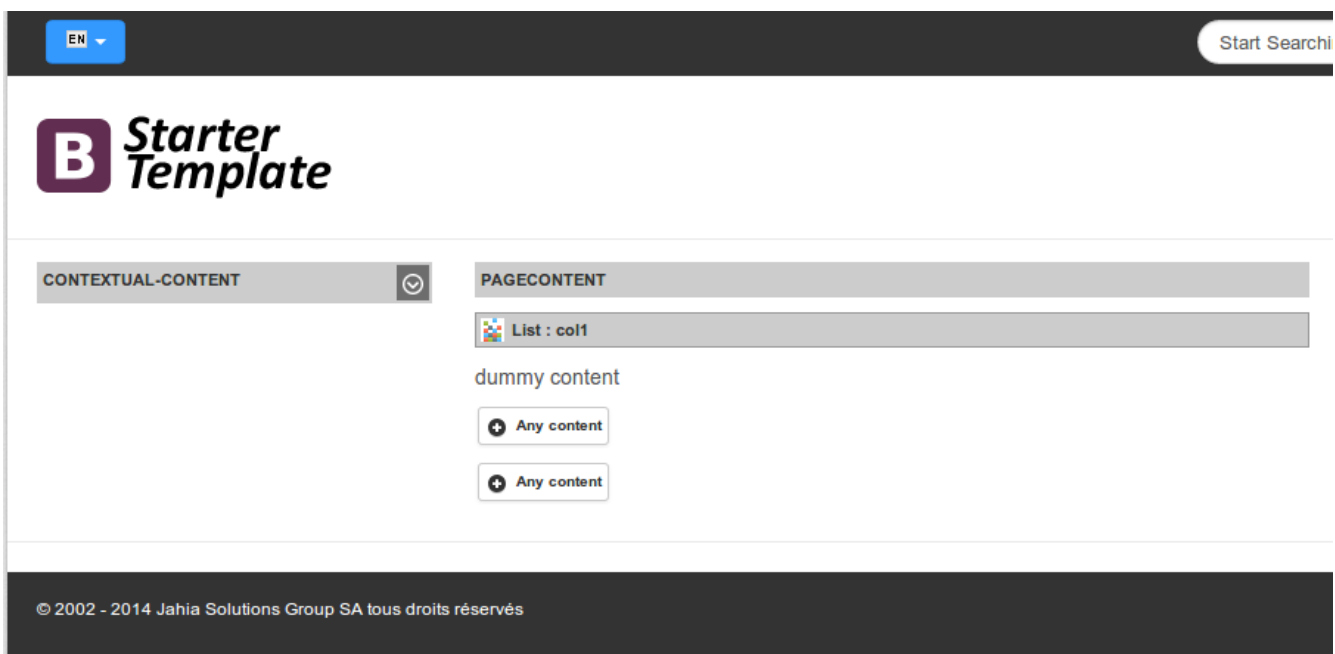
By saving this form the configuration will be reloaded and will look like that this:

```
1 ## Esigate configuration
2
3 connectTimeout=2000
4 ttl=5
5 extensions=org.esigate.servlet.ServletExtension,org.esigate.extension.Esi
6 mappings=/*
7 maxObjectSize=1000000
8 useCache=false
9 remoteUrlBase=http://localhost:8080/jahia
10
11
12 # Provider contact
13 contact.remoteUrlBase=http://localhost:8080/sample-contact-webapp/
14 contact.extensions=org.esigate.servlet.ServletExtension,org.jahia.modules.portalFactory.esigate.filter.JahiaEsiExtension,org.esigate.extension.Esi
15 contact.uriEncoding=UTF-8
16 contact.connectTimeout=1000
17 contact.useCache=false
18 contact.mappings=/ext/contact/*
19 contact.visibleUrlBase=http://localhost:8080/jahia/ext/contact/${jahia.mode}/${jahia.language}
20 contact.defaultPageInclude=/sites/mySite/home.html
21 contact.defaultFragmentReplace=coll
22 contact.context=
23
```

The provider “contact” has been added to the Esigate configuration, including all necessary information to be able to “clip” the application into your web page.

6.2 Inserting the application into a web page

In the form we have specified the home page of the site named mySite as default include page. Then, in this page we can now add an **“ESI fragment list”** component (this is the component that adds the esi:fragment tag). As for any other component it can be found in the selector’s second tab, or in the components list that appears in pop-up when clicking on one of the “Ad content” buttons in the page itself. Drag and drop the component where you want into an active area of the page.



In our example, we have inserted the “ESI fragment list” component and named it col1. The list can be filled with any standard content as usual. In the above example we have put a dummy content (rich text). If a user navigates to that page the content will be displayed normally, but when this page is used by Esigate, the external app will be placed in this Esi fragment list and this content will be replaced automatically.

You can access to the webapp directly by calling its URL <http://localhost:8080/sample-contact-webapp/> to see how it behaves as a stand-alone application.

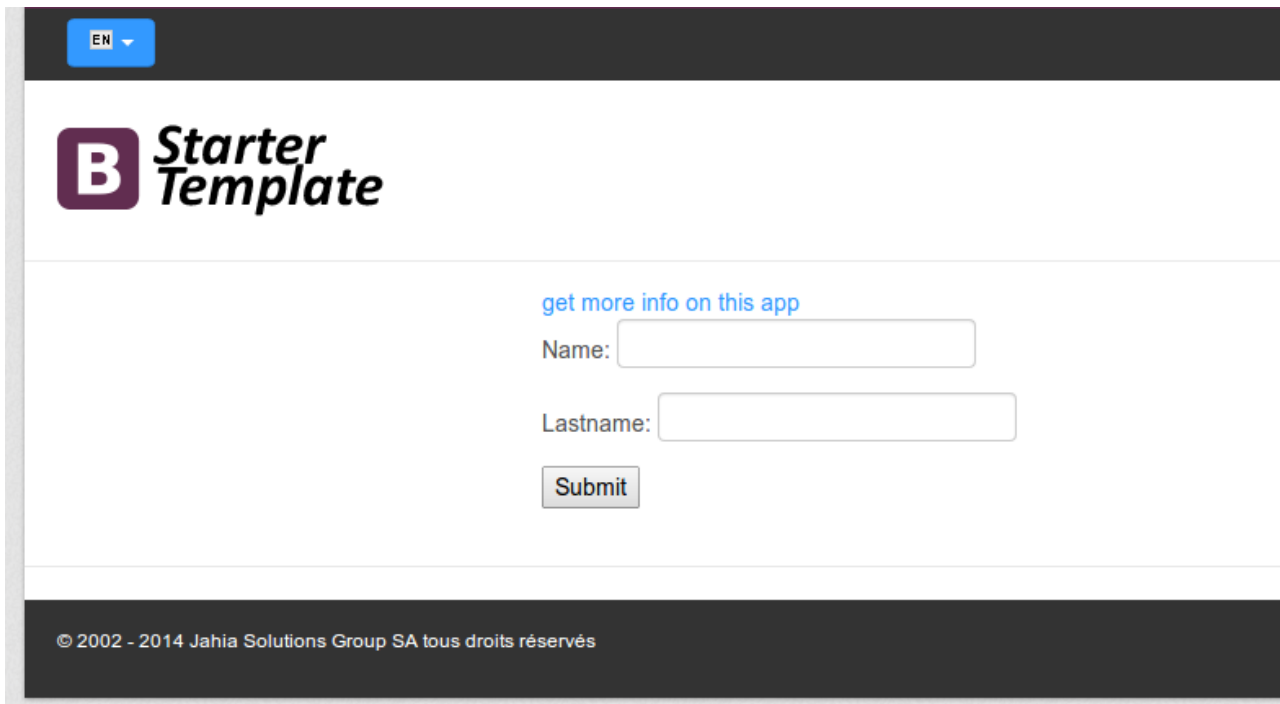
[get more info on this app](#)

Name: Lastname:

You can also access to the application through Esigate using the URL

<http://localhost:8080/ext/contact/render/live/en>. The Esigate url is formed using the following pattern

http://<DOMAIN>/<CONTEXT>/ext/<PROVIDER_KEY>/render/live/en



This time the external application is fully aggregated into the Digital Factory page.

6.3 Direct links to aggregated apps

Now that your application is bound to the URL http://localhost:8080/jahia/ext/contact/* you'll need to create links to access it through Esigate. In this case it is important to specify the mode and the language in which Digital Factory must render it. The module provides an Esigate extension to replace and rewrite dynamically URLs in the external app.

To make a correct link to the external app, you can use this URL in your pages:

<http://localhost:8080/jahia/ext/contact/{mode}/{lang}>

Digital Factory will replace automatically the {mode} and {language} placeholders by the current values in the current page.



Jahia Solutions Group SA

9 route des Jeunes,
CH-1227 Les acacias
Geneva, Switzerland

<http://www.jahia.com>

