This SDL Web Deployer Notification module allows you to send notifications to different channels after a set of content conditions are met. The conditions can be applied to any content items that are published to your delivery environment. For example this extension allows you to send a notification after a content item associated to a certain taxonomy has been published.

The plugin is build targeting Web 8.1.1 or above, once the plugin is built it most likely will work with any 8.x version of SDL Web.

Features

- Conditional notifications supporting schema, publication and taxonomy conditions
- Email and Hipchat notification support out of the box
- Templating support for the notification message

Conditions

This deployer module is driven by conditions that define when notifications are being sent out. When all conditions are met the notification will be triggered.

Conditions are specified in blocks under the ‘Conditions’ element. All conditions under there are by default triggered as Or operations. This means if any of the root conditions is true, the notification will get triggered. There are two logical operations ‘And’/‘Or’ which can be

used as blocks containing multiple sub conditions, which can be fully nested in each other.

For the moment the following conditions are present.

```
<ItemTypeCondition ItemType="64"/>
<PublicationCondition PublicationId="1"/>
<TaxonomyCondition TaxonomyId="10"/>
<SchemaCondition SchemaId="555"/>
```

It is possible inside any root condition to override the notification settings. If no notification settings in the root block are specified it will use all notifiers configured.

In case you want to use a limited set of notifiers for a root condition you can override this by specifying a ‘Notifiers’ block as following, this will ensure only email gets sent:

```
<Notifiers>
    <Notifier Id="MailNotifier">
        <!-- Template can be overridden for specific cond
ition sets -->
        <Property Name="template" Value="Taxonomy Item wi
th title: '${title}' and itemId ${id} was published to pu
blication: ${publicationId} on: ${lastPublicationDate}"/>
        <Property Name="subjectTemplate" Value="Taxonomy
Item ${title} with id ${id} published"/>
```

```
    </Notifier>
</Notifiers>
```

In a notifier block it is also possible to override the default properties defined for a notifier. It will always default to the default notifier configuration if nothing specified.

Notifiers

The default notifiers for this module are an email notifier and a HipChat notifier. This means once conditions are met these two channels can be used to send notifications. It is also possible to introduce your own notifiers, see the below section on how to proceed on that.

Hipchat

One of the default notifiers is one that can send messages to the private chat for teams service called HipChat. In order to configure this you need to specify this below fragment

```
<Notifier Id="HipChatNotifier">
    <Property Name="token" Value="roomToken"/>
    <Property Name="room" Value="notification-room"/>
    <Property Name="template" Value="Item with title: '${
title}' and itemId ${id} was published in publication: ${
publicationId} on: ${lastPublicationDate}"/>
```

```
</Notifier>
```

You need to obtain a token valid for a team chatroom and specify the name of the chatroom in the configuration. Besides this you can specify a message template that defines what message is generated when a condition is met.

Email

The second default notifier is the email extension which allows sending email notifications when the content conditions are met. In order to use the email notifier make sure all the connection details are specified like below:

```
<Notifier Id="MailNotifier">
    <Property Name="mail.smtp.auth" Value="false"/>
    <Property Name="mail.smtp.host" Value="smpt-host"/>
    <Property Name="mail.from" Value="noreply@mydomain.co
m"/>
    <Property Name="mail.to" Value="notifier@mydomain.com
"/>
    <Property Name="subjectTemplate" Value="Item ${title}
 with id ${id} published"/>
    <Property Name="template" Value="Item with title: '${
title}' and itemId ${id} was published in publication: ${
publicationId} on: ${lastPublicationDate}"/>
```

```
    <!--<Property Name="mail.smtp.starttls.enable" Value=
""/>-->
    <!--<Property Name="mail.smtp.port" Value=""/>-->
    <!--<Property Name="mail.smtp.user" Value=""/>-->
    <!--<Property Name="mail.smtp.password" Value=""/>-->
</Notifier>
```

Also in this sample both a template for the content as the subject can be customised for a custom message.

Setup and Configuration of Extension

Step1: Download the latest distribution from the github releases pages:

Step2: Install the below fragment in your deployer configuration:

```
        <Processor Action="Deploy" Class="com.tridion.dep
loyer.Processor" Phase="post-transaction">
            <Module Class="com.sdl.deployer.notification.
NotificationModule" Type="NotificationHook">
                <Conditions>
                    <!-- Multiple condition blocks can be
 specified, each can trigger a notification -->
                    <AndCondition>
                        <ItemTypeCondition ItemType="64"/
```

```
>
                        <PublicationCondition Publication
Id="1"/>
                        <TaxonomyCondition TaxonomyId="10
"/>

                        <!-- You can select which notifie
rs to use -->
                        <Notifiers>
                            <Notifier Id="MailNotifier">
                                <!-- Template can be over
ridden for specific condition sets -->
                                <Property Name="template"
 Value="Taxonomy Item with title: '${title}' and itemId $
{id} was published to publication: ${publicationId} on: $
{lastPublicationDate}"/>
                                <Property Name="subjectTe
mplate" Value="Taxonomy Item ${title} with id ${id} publi
shed"/>
                            </Notifier>
                            <Notifier Id="HipChatNotifier
"/>
                        </Notifiers>
                    </AndCondition>
                    <AndCondition>
                        <!-- all these conditions will us
e the default notifiers -->
                        <ItemTypeCondition ItemType="16"/
```

```
>
                        <SchemaCondition SchemaId="555"/>
                    </AndCondition>
                </Conditions>
                <Notifiers>
                    <Notifier Id="MailNotifier">
                        <Property Name="mail.smtp.auth" V
alue="false"/>
                        <Property Name="mail.smtp.host" V
alue="smpt-host"/>
                        <Property Name="mail.from" Value=
"noreply@mydomain.com"/>
                        <Property Name="mail.to" Value="n
otifier@mydomain.com"/>
                        <Property Name="subjectTemplate"
Value="Item ${title} with id ${id} published"/>
                        <Property Name="template" Value="
Item with title: '${title}' and itemId ${id} was publishe
d in publication: ${publicationId} on: ${lastPublicationD
ate}"/>

                        <!--<Property Name="mail.smtp.sta
rttls.enable" Value=""/>-->
                        <!--<Property Name="mail.smtp.por
t" Value=""/>-->
                        <!--<Property Name="mail.smtp.use
r" Value=""/>-->
                        <!--<Property Name="mail.smtp.pas
```

```
sword" Value=""/>-->
                    </Notifier>
                    <Notifier Id="HipChatNotifier">
                        <Property Name="token" Value="roo
mToken"/>
                        <Property Name="room" Value="noti
fication-room"/>
                        <Property Name="template" Value="
Item with title: '${title}' and itemId ${id} was publishe
d in publication: ${publicationId} on: ${lastPublicationD
ate}"/>
                    </Notifier>
                </Notifiers>
            </Module>
        </Processor>
```

Step3: Unpack the extension and install all the jar files in your deployer services directory (In case of windows run the uninstall/installService command to update the classpath)

Building from source

Install Maven dependencies

In order to build this plugin you will need to have the dependencies for the Deployer installed. You can do this by providing the Jar files from your local SDL Web distribution layout and installing them as Maven

dependencies in your local maven repository.

```
mvn install:install-file -Dfile=cd_deployer-8.1.1-1016.ja
r -DgroupId=com.sdl.delivery -DartifactId=cd_deployer -Dv
ersion=8.1.1 -Dpackaging=jar
mvn install:install-file -Dfile=deployer-api-8.1.1-1016.j
ar -DgroupId=com.sdl.delivery -DartifactId=deployer-api -
Dversion=8.1.1 -Dpackaging=jar

mvn install:install-file -Dfile=cd_model-8.1.1-1008.jar -
DgroupId=com.tridion -DartifactId=cd_model -Dversion=8.1.
1 -Dpackaging=jar

mvn install:install-file -Dfile=cd_common_config_legacy-8
.1.1-1005.jar -DgroupId=com.tridion -DartifactId=cd_commo
n_config_legacy -Dversion=8.1.1 -Dpackaging=jar

mvn install:install-file -Dfile=cd_core-8.1.1-1010.jar -D
groupId=com.tridion -DartifactId=cd_core -Dversion=8.1.1
-Dpackaging=jar

mvn install:install-file -Dfile=cd_common_util-8.1.1-1006
.jar -DgroupId=com.tridion -DartifactId=cd_common_util -D
version=8.1.1 -Dpackaging=jar
```

Please make sure to replace the filename with the actual filename your are installing, in above case jar files with exact versions are used,

pending on your SDL Web 8.x version the filenames might be called slightly differently.

Create new Notifiers

In order to add new Notifiers two steps need to be taken:

1. Implement the Notification interface In order to create a Notification plugin, all that is needed to do is implement the ‘com.sdl.deployer.notification.Notifier’ interface. This interface has one main method to be implemented called the ‘sendNotification(Optional<NotifierConfig> notifierConditionConfig, Item item);’ method.

Please see the API documentation about more information on the provided parameters.

2. Create a service registration file If the notifier has been created what is needed next is to add an extra to ‘src/main/resources/META-INF/services/com.sdl.deployer.notification.conditions.ConditionLoader’ file with the new class containing the Notifier.