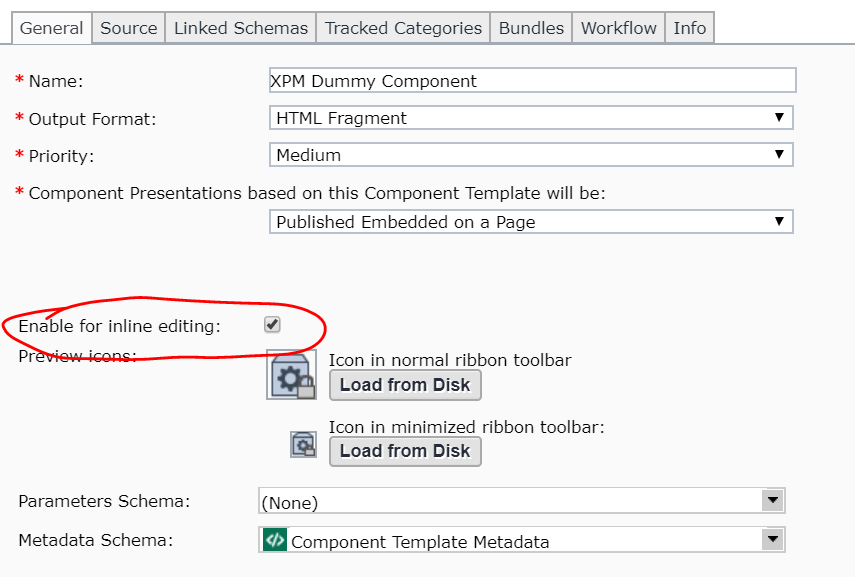
# Experience Manager Second Level Editing

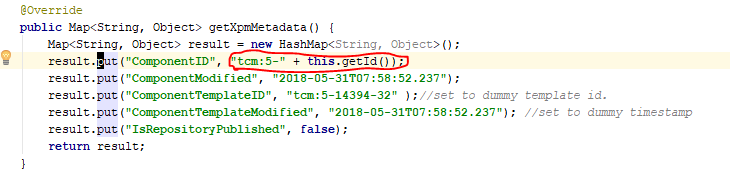
## Create Dummy Component Template in Tridion

The experience manager operates based a commented metadata section in the HTML file, which requires a component template ID to work.

Normally first level components in layout have existing component templates, however linked components don’t, so we have to create a dummy component template to link tEverything can be left as default as they don’t matter EXCEPT the inline editing field must be enabled for xpm to work.

## Implement getXpmMetadata function in the entity java file.

In this example, we wanted to edit text objects no matter what level. Since all text objects have an attribute of type Text, we implemented the function in the Text entity in Text.java.

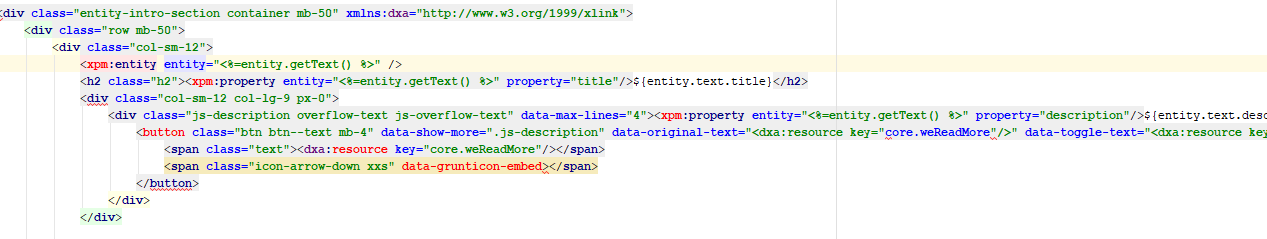


This creates the commented metadata section in HTML necessary for XPM display. Note the componentTemplateID is the tcm of the dummy component template we just created. The modified field does not matter, and the componentID can be obtained through the getID() function.

Note the publication is hard coded. If the component is not in 400 EN, you will need to change the publication number.

If we want to display other second level components such as links, we would have to implement the same function in the corresponding entity class.

## Configure XPM in the jsp file.

The XPM only automatically picks up first level entities and generate metatdata text for them, therefore if we want metadata generated for linked components, we have to specify them as XPM entities in the entity jsp file. In this example we used TextLeftLinksRight.jsp

First we specify the entity

<xpm: entity entity= “%=entity.getText() %>” />

This tells the XPM that there exist a second level entity, which is “entity.getText()”. This getText() function returns the Text object, in which we implemented the getXpmMetadata() function. Always make sure the entity you assign has that function implemented.

Then we specify which fields of that entity we want to edit.

<xpm: property entity= … property = “title”/>

This tells the XPM that we want to modify the title field. Similarly we can add a tag to enable editing for description field.

## Enable XPM feature

Lastly, we have to enable XPM feature, ie the pen icon on the page to enter SDL.



Add this to webExtraPage.jspx. Usually it’s above the header in the body section.