

Création/Manipulation de fichier XML

Les méthodes suivantes permettent de créer et manipuler les fichiers XML . Cela peut par exemple servir pour exporter un fichier XML à partir de données d'Open-Prod depuis une action serveur par exemple.

Méthode / Exemple	Données d'entrée	Données de sortie
mf_get_lxml_library() Exemple : lxml = env['mf.tools'].mf_get_lxml_library() etree=lxml.etree	--	Retourne la librairie lxml
lxml_element_get_text(xml)	Objet Element de la la librairie etree	Texte de l'objet Element donnée en paramètre
lxml_element_get_tag(xml)	Objet Element de la la librairie etree	Nom de l'objet Element donnée en paramétré
lxml_element_set_text(xml,text)	Objet Element de la la librairie etree ,Texte à mettre dans la balise	Objet Element modifié
lxml_element_set_ta(xml,tag)	Objet Element de la la librairie etree ,Nom à affecter à la balise	Objet Element modifié

Exemple : dans une action serveur

```
tools = env['mf.tools']
etree = tools.mf_get_lxml_library().etree

data_xml_de_base=""<?xml version="1.0" encoding="utf-8"?>
<ProcessMaterialInformation>
  □<ApplicationArea>
    <Sender>
      <LogicalID>OPP</LogicalID>
    </Sender>
    <Receiver>
      <ID>QUBES</ID>
    </Receiver>
    <CreationDateTime>2010-07-22T11:20:33</CreationDateTime>
```

```

</ApplicationArea>
  <DataArea>
    <<Process>
    <</Process>
    <<MaterialInformation>
      <<<ID>ItemsDefinition</ID>
      <<<Description>Material classes and material definitions for all the items</Description>
      <<<PublishedDate>2010-07-22T11:20:33</PublishedDate>
    <</MaterialInformation>
  </DataArea>
</ProcessMaterialInformation>
""

```

```

xml=etree.fromstring(data_xml_de_base)
#incrément auto à chaque export : sequence code ITEM_QUBES
res=env['ir.sequence'].next_by_code('ITEM_QUBES')
BODID=etree.SubElement(xml,'BODID') # crée balise BODID dans la balise root du xml
tools.lxml_element_set_text(BODID,res) #affecte valeur texte à la balise créé

#Ajoute une balise article pour chaque produit en vie avec une sous balise id et description
MaterialInformations= xml.xpath("//MaterialInformation")
article_rcs=env['product.product'].search([('state','=', 'lifeserie')])
for MaterialInformation in MaterialInformations:
    for article_rc in article_rcs:
        # créer une sous balise MaterialDefinition dans MaterialInformation
        MaterialDefinition=etree.SubElement(MaterialInformation,'MaterialDefinition')# crée
balise MaterialDefinition
        # Créer balise ID avec la valeur -> code
        codep=etree.SubElement(MaterialDefinition,'ID')
        tools.lxml_element_set_text(codep,article_rc.code)
        # Créer balise Description avec la valeur -> name
        codep=etree.SubElement(MaterialDefinition,'Description')
        tools.lxml_element_set_text(codep,article_rc.name)

#Récupère le contenu en texte
file_content=etree.tostring(xml)
#sauvegarde le text dans fichier sur le serveur
env['mf.tools'].save_file("/home/openprod/Documents/toto.xml",file_content)
#affiche le contenu
#raise Warning(etree.tostring(xml))

```

Revision #3

Created 23 September 2024 07:09:46 by Emeric ORSEL

Updated 23 September 2024 07:54:07 by Emeric ORSEL