Documentation Extensions

Bienvenue dans la documentation des extensions de FormPublisher.

1. Extensions et échanges programmés

FormPublisher intègre un langage d'expressions appelé langage **JIL**.

Ce langage permet de décrire des séquences qui vont influer sur le rendu d'un document (**View**), et sur les données (**Model**) en permettant par exemple de les valider.

Les composants du langage représentent un ensemble fini d'instructions. Il est néanmoins possible d'ajouter de nouveaux comportements par le biais des extensions.

La réalisation d'une extension se passe en deux étapes :

- la création de l'extension :
  la classe doit être mise dans le package *lu.jway.extension*
  elle doit étendre *lu.jway.webapp.jil.FormPublisherExtension*

- L'appel de l'extension en langage Jil:
  <?variable param[0]= ' ...'?>
  <?variable param[1]='....'?>
  <?variable param[2]='....'?>
  <?variable result=FormPublisherExtension(this,'UneExtension', $(param))?>
  Nous voyons ici comment une extension est appelée par le biais de l'instruction **JIL** FormPublisherExtension et la structure **JIL** retournée.

FormPublisher adhère donc à la logique MVC mais avec néanmoins une exception que nous appellons « échanges programmés ».

Les échanges programmés peuvent prendre différentes formes dans FormPublisher, ils ont pour but d'avoir une incidence sur les données soit interne, soit externe.

Dans FormPublisher, nous avons donc une notion de modèle étendu qui ne se limite pas au contexte de la session du formulaire.

Pour influer sur ce modèle étendu, plusieurs possibilités :

1.1. Extensions par le biais de l'instruction FormPublisherExtension

- JdbcCachedQuery
  Cette extension permet de faire des requêtes sur une base de données par le biais d'un drivers JDBC. La réponse retournée est transformée en une structure de données JIL utilisable par la partie serveur du formulaire.
  Les données doivent être intégrées dans le contexte du formulaire pour figurer dans le DataStore complete.
  <Variable Name="args[0]" Expression=""request"" Submit="false" DataType="string" />
  *<?variable args[1]="SELECT * FROM rues_luxembourg WHERE cp="'+ $ (lieuCodepostal)+"' AND reseauVille=1"?>*
  <Variable Name="res" Expression="FormPublisherExtension(this,"JdbcCachedConnector",$(args))" Submit="false" DataType="string" />

- DataStoreSpy

  Cette extension permet de faire une requête de type HTTP GET à une serveur de services qui peut retourner différents formats de réponse. Le format par défaut étant le format DataStore. Dans le cas d'autres formats, un filtre doit être implémenté et spécifié au niveau de l'url du service par le paramètre : parser.

  Les données peuvent être ou non intégrées dans le contexte du formulaire en fonction des paramètres fournis.

  <?variable param[0]='http://<serveur>/<service>&parser=json'?>

  <?variable param[1]='userData,metaInfoData'?>

  <?variable param[2]='false'?>

  <?variable timbre_called=FormPublisherExtension(this,'DatastoreSpy', $(param))?>

  Le premier paramètre correspond à la requête URL qui sera faite. Cette URL peut contenir des marqueurs *{x}* pour signifier des valeurs de remplacement à prendre en compte ou x est l'indice dans la liste de paramètres.

  Le second paramètre correspond aux espaces de données qui seront actualisés ou recréés en fonction du paramètre suivant.

  En dernier vient l'appel de l'extension.

  Dans le cas présenté le filtre qui va réaliser le parsing aura comme identification *lu.jway.webapp.ds.JsonFilter*