

Transactional Documents 2

Creating a transactional document works a lot like creating a maintenance document, but with the added need to put together the web layer. This isn't as hard as it seems, though, and for the trouble, the developer gains a lot of flexibility. Here, we'll examine how to create a transactional document.

What is a transactional document? A transactional document represents not data that is being maintained, but an "action" that will occur in the system. Take the Internal Billing document. This is a financial document which charges one account for a service or goods and credits another account. Here, we're not really changing the core data of the chart, the accounts, or any of that. We're creating accounting lines with the sole purpose of an action: having them change the general ledger (adding money to one account and removing it from another). This is the essential difference between transactional documents and maintenance documents: while the maintenance documents do standard maintenance on data, here we want to perform an action.

That doesn't mean we're free of business objects, of course. A transactional document is itself a very special kind of business object; oftentimes, it includes heterogeneous collections of other kinds of business objects within it (though there is no requirement for it to do so). Otherwise, the recipe to create a transactional document looks a bit like this:

1. create the transactional document business object and any related business objects
2. create rules for the transactional document
3. write a document authorizer for the document if needed
4. create KEW and KFS document types
5. configure the data dictionary transactional document file
6. put all of the non-rule logic associated with the transactional document completing its action into a service-layer class
7. create a user interface for the document - the Action, the Form, and a JSP page

Let's look at each of these steps in turn:

- Creating the Transactional Document business object
 - Creating Transactional Document business object
 - Doing the OJB Mapping
 - The transactional document as a Document
 - Adding collections to Transactional Documents
- Gluing everything together with the Data Dictionary for Transactional Documents
 - The basic attributes
 - documentClass
 - businessRulesClass and preRulesCheckClass
 - Tying in the document authorizer
 - documentAuthorizerClass
 - authorizations
 - documentTypeName and documentTypeCode
 - Helpful UI Attributes
 - label
 - attributes
 - summary and description
 - help
 - headerNavigation
 - Special Transactional Document attributes
 - allowsErrorCorrection
 - allowsCopy
- Transactional Document User Interface
 - Creating the Form
 - Extending KualiTransactionalDocumentFormBase
 - The children of KualiTransactionalDocumentFormBase
 - Creating new slots for adding to collections
 - Creating the Action
 - Extending org.kuali.core.web.struts.action.KualiTransactionalDocumentActionBase
 - Pre-defined action methods
 - Action methods
 - Generating new rule events
 - Setting up the Struts mapping
 - Using JSP Pages
 - Creating a JSP page
 - Tags to know
 - Creating a taglet, or, The Return of TD

Unable to render {include} The included page could not be found.