

KULRICE-9918 - Providing a way for applications to specify custom KEN document types - Design Analysis

Introduction

The KC project has requested for `KRACOEUS-6593` - Data cannot be retrieved due to an unexpected error that Rice be able to specify custom KEN document types in order to specify what notifications get dispatched to users. The service `NotificationMessageDeliveryResolverService` periodically picks up any unresolved notifications and turns them into KEW notifications via `KEWActionListMessageDeliverer.deliverMessage(NotificationMessageDelivery messageDelivery)`. Eventually, this method will call into `NotificationWorkflowDocument.createNotificationDocument(InitiatorUserId)` and use the default KEN notification type. This is the only place in the production code where `KualiNotification` is used; thus, only a few internal classes and data objects will have to be changed to make this happen.

Analysis

The way that KEN handles notifications is to allow them to be sent through `SendNotificationService` (which has an implementing class of `SendNotificationServiceKewXmlImpl` via the method `public NotificationResponse sendNotification(Notification notification)`. The notification is immediately handed to the `NotificationService` where it is persisted in the database to wait to be dispatched to users. The service `NotificationMessageDeliveryResolverService` periodically picks up any unresolved notifications and turns them into KEW notifications via `KEWActionListMessageDeliverer.deliverMessage(NotificationMessageDelivery messageDelivery)`. Eventually, this method will call into `NotificationWorkflowDocument.createNotificationDocument(InitiatorUserId)` and use the default KEN notification type. This is the only place in the production code where `KualiNotification` is used; thus, only a few internal classes and data objects will have to be changed to make this happen.

Design

Since the `NotificationService` merely persists notifications and is not responsible for disseminating them, completing this request requires an API addition to `NotificationContract` and a subsequent change all dependent classes that affect the database. To follow convention, it is recommended to use `getDocTypeName()` in the interface and `DOC_TYP_NM (VARCHAR(64))` as the new optional field in `KREN_NTFCTN_T`. Once this is added to both `Notification` and `NotificationBo`, then the document type will be passed all the way to `NotificationWorkflowDocumentService.createAndAdHocRouteNotificationWorkflowDocument(...)` where the call

```
WorkflowDocument document =
NotificationWorkflowDocument.createNotificationDocument(InitiatorUserId);
```

will include a safe version of something like

```
WorkflowDocument document =
NotificationWorkflowDocument.createNotificationDocument(InitiatorUserId,
messageDelivery.getNotification().getDocTypeName());
```

If there are any errors like not being able to find the notification or having a null document type, then the code will revert to the previous method and use the standard `KualiNotification` type.

Impact

This change will have the following impacts:

- **Database change in Rice 2.3.1 with a non-required field.** This should not affect backwards compatibility.
- **API change for `NotificationContract`.** This should not affect backwards compatibility.
- **Internal code change.** Safe.