Оглавление

[Список задач для выполнения рефакторинга в CMJ 2](#_Toc366074932)

[Приложение 1 4](#_Toc366074933)

# Список задач для выполнения рефакторинга в CMJ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №№ | Наименование работ | Исполнители | Трудо-емкость (чел/часы) | Статус/Задача | Комментарии |
| 0. | Постановка и уточнение задачи | Тугушев | 4 | Уже выполнено |  |
| 1. | Иметь возможность конфигурацией подменять NotesFactory на нашу фабрику сессий, например, через xml-дескрипторы Spring Beans | Ноздрачёв, Иванов | 4 | Принято исполнителем Ноздрачев. | Выполняется совместно обоими исполнителями |
| 2. | Перенести из web-модуля “rest” все классы в какой-нибудь нижестоящий jar-модуль | Болотов или Тугушев | 2 | Уже сделано в 4.2 |  |
| 3. | Где возможно, заменить Evaluate на аналоги через Java API | Ноздрачёв, Новикова | 16 | Принято исполнителем  Новикова + Ноздрачев | Выполняется совместно обоими исполнителями |
| 4. | Оставшиеся операции, использующие @-формулы вынести в отдельные программные компоненты, подключаемые через конфигурацию | Ноздрачёв, Болотов, Новикова, Иванов | 16 | Принято исполнителем, Ноздрачёв, Болотов, Новикова, Иванов | Строго после п.3!!!  Задача выполняется всеми исполнителями. Сначала Ноздрачёв или Тугушев готовит образец |
| 5. | Иметь возможность подменять имплементацию сохранения форматированного текста на прямое сохранение HTML | Болотов | 16 | Принято исполнителем, Болотов |  |
| 6. | Доработка API ПКД в части создания/удаления уведомлений и входящих задач | Левковская | 2 | Принято исполнителем Левковска |  |
| 7. | Возможность отключения AFSaveDocUtils | Ноздрачёв | 4 | Принято исполнителем Ноздрачев. |  |
| 8. | Новый, автономный код для 4.2 (интеграция с exchange) |  |  | А. Савельев /О. Таранченко - довести до разработчикаов правила рефакторинга и обнаружения подозрительных мест. (приложение 1) |  |
| 9!!! | Подготовка договоров - BPM движок |  |  |  | Ожидается консультация с Ивановым, отв. Тугушев. Не позднее 06.09 |

* Организация ветки для 4.2.1 не ранее 15.09.2013!
* Организационно (для исполнителей!!!) : вставить комментарии в код при рефакторкнге для облегчения процесса поиска изменений при переносе в 4.3
* Проект см 4/5 в конце сентября получает 4.2.1.x, как Snapshot версию с выполненным рефакторингом.

## Приложение 1

**Цели рефакторинга.**

Рефакторинг должен обеспечить наличие уровня опосредованности функционала, сильно завязанного на возможности Domino. Иными словами, позволять подменять имплементацию функций.

1. Иметь возможность конфигурацией подменять NotesFactory на нашу фабрику сессий, например, через xml-дескрипторы Spring Beans

Компоненты, использующие родную имплементацию Notes Java API задействуют бин, который получает Session через NotesFactory. Компоненты, использующие другую имплементацию Notes Java API задействуют бин, который создаёт кастомную имплементацию Session (EmulatorSession) .

Оба бина имеют единый интерфейс.



2. Перенести из web-модуля “rest” все классы в какой-нибудь нижестоящий jar-модуль

Чтобы переиспользовать на уровне бинарников код CMJ-Server. Web-модуль нельзя подключить как библиотеку к другому проекту.

3. Где возможно, заменить Evaluate на аналоги через Java API

Тем самым избавимся от сильной зависимости Domino-интерпретатора @-формул.

4. Оставшиеся операции, использующие @-формулы вынести в отдельные программные компоненты, подключаемые через конфигурацию

Тем самым избавимся от сильной зависимости Domino-интерпретатора @-формул. Для реализации предлагается вынести функции, выполненяемые @-формулами в отдельный Java-метод. Этот метод должен быть интерфейсным, давая возможность подменять имплементацию. Например, есть функция поиска шаблона РКК реализованная через поиск, критерии которого задаются @-формулой:

final DocumentCollection coll = db.search("Form = \"RKKTemplate\" & Met <> \"1\" & @IsUnavailable($Conflict) & TemplName = \"" + templateName + "\"");

Заводим интерфейс, в котором будут перечислены методы для различных выборок, требуемых модулю. В нем, среди прочих метод для поиска шаблонов РКК

interface DpDao {

....

DocumentCollection findRkkTemplate(Database db, String templateName );

}

Делаем имплементацию

class CmjDpDao implements DpDao{

...

DocumentCollection findRkkTemplate(Database db, String templateName )

{

return db.search("Form = \"RKKTemplate\" & Met <> \"1\" & @IsUnavailable($Conflict) & TemplName = \"" + templateName + "\"");

}

}

Регистрируем бин для DpDao.

<bean id="dpDao" class = "...CmjDpDao">

Заменяем прямой вызов db.search на получение бина и вызов findRkkTemplate

DpDao dao = BeanUtils.getBean("dpDao");

dao.findRkkTemplate(templateName );

Подмена имплементации осуществляется заменой конфигурации бина без вмешательства в кодовую базу CMJ-Server

<bean id="dpDao" class = "...CustomDpDao">

5. Иметь возможность подменять имплементацию сохранения форматированного текста на прямое сохранение HTML

С CMJ-WebGui поступает HTML. Нужно ввести уровень опосредования для варирования способа конвертирования и хранения этого HTML.

Техника аналогична п.4

6. Доработка API ПКД в части создания/удаления уведомлений и входящих задач

Для возможности повторения существующего функционала CMJ исключительно на Java-коде.

7. Возможность отключения AFSaveDocUtils

Его функции надо реализовать другими средствами.

**Требования к дизайну разрабатываемого кода**

Чтобы не выполнять указанный рефакторинг повторно. Необходимо следующее.

1. Операции, использующие @-формулы должны выделяться в заменяемые программные компоненты. Компонента не является заменяемой, если для её замены требуется модифицировать компоненты, с ней взаимодействующие.

2. Как максимум. Создание любого экземпляра любого объекта выполнять через фабрику, чью имплементацию можно подменять (паттерн AbstractFactory). Spring Beans является AbstractFactory. Static factory method must die. Исключения - по согласованию с архитектором.

Как минимум. Через абстрактную фабрику получать

- NotesFactory

- объекты, использующие @-формулы

- объекты, осуществляющие непосредственное взаимодействие с системой хранения

- объекты, выполняющие конвертацию структур данных.

3. В web-, ear-модулях проекта только файлы конфигураций. Исключения - по согласованию с архитектором.