**Общие положения.**

Точки расширения предназначены для обеспечение модернизации логики работы ядра системы и встраивания дополнительного функционала, который должен отрабатывать в те или иные моменты работы системы.

### Создание точки расширения

Для создания точки расширения необходимо создать интерфейс точки расширения, расширяющий интерфейс ru.intertrust.cm.core.dao.api.extension.ExtensionPointHandler. В данном интерфейсе объявить метод точки расширения. Например, интерфейс точки расширения, вызывающейся после сохранения доменного объекта выглядит следующим образом:

package ru.intertrust.cm.core.dao.api.extension;

import ru.intertrust.cm.core.business.api.dto.DomainObject;

public interface AfterSaveExtensionHandler extends ExtensionPointHandler {

void onAfterSave(DomainObject domainObject);

}

Далее, в том месте где необходимо вызвать все обработчики данной точки расширения необходимо вставить код получающий обработчики точки расширения с помощью spring сервиса ru.intertrust.cm.core.dao.api.ExtensionService и вызвать их исполнение. Например, код вызывающий обработчик точки расширения AfterSaveExtensionHandler выглядит следующим образом:

@Autowired

private ExtensionService extensionService;

…

// Вызов точки расширения после сохранения

AfterSaveExtensionHandler afterSaveExtension = (AfterSaveExtensionHandler) extensionService

.getExtentionPoint(AfterSaveExtensionHandler.class,

domainObject.getTypeName());

afterSaveExtension.onAfterSave(domainObject);

В качестве параметров метода возвращающего обработчики точек расширения передается интерфейс точки расширения и фильтр. Фильтр предназначен для оптимизации данного механизма по скорости. Оптимизация заключается в том что вызываться будут не все точки расширения, которые имплементят интерфейс, переданный в первом параметре, а только те у которых совпадает значение фильтра с тем, который указан при объявлении обработчика точки расширения.

## Создание обработчика точки расширения

Для создания обработчика точки расширения необходимо создать класс, имплементяший интерфейс нужной точки расширения и аннотировать его аннотацией ru.intertrust.cm.core.dao.api.extension. ExtensionPoint. В аннотации можно указать фильтр, когда вызывать данный обработчик. Если фильтр не указан данный обработчик будет вызываться всегда, когда вызываются обработчики данного типа. Например, реализация обработчика точки расширения, вызывающейся после сохранения доменного объекта может выглядеть следующим образом:

**package** ru.intertrust.cm.core.dao.impl.personmanager;

**import** ru.intertrust.cm.core.business.api.dto.DomainObject;

**import** ru.intertrust.cm.core.dao.api.extension.AfterSaveExtensionHandler;

**import** ru.intertrust.cm.core.dao.api.extension.ExtensionPoint;

@ExtensionPoint(filter="Person")

**public** **class** OnSavePersonExtensionPointHandler **implements** AfterSaveExtensionHandler{

@Override

**public** **void** onAfterSave(DomainObject domainObject) {

System.*out*.println("On after save for person " + domainObject.getTypeName() + ":" + domainObject.getId());

}

}

Обработчиков точек расширения может быть неограниченное множество, все они вызовутся. Порядок вызова точек расширения не декларируются. Точки расширения должны лежать в пакетах, являющимися дочерними к тому пакету, который указан в конфигурации spring бина ru.intertrust.cm.core.dao.impl.ExtensionServiceImpl в атрибуте basePackage. Вот так выглядит конфигурация на текущий момент в файле bean.xml проекта dao-impl.

<bean id="extensionService" class="ru.intertrust.cm.core.dao.impl.ExtensionServiceImpl">

<property name="basePackage" value="ru/intertrust/cm" />

</bean>