

Код:	[TM119]	Версия:	0.01
Дата создания:	04.09.2014	Дата изменения:	05.09.2014
Автор:	Абдулхаиров Равиль	Редактор:	Радзишевский Алексей
Статус:	В работе		
Назначение ТК:	Реализация WEB сервиса на базе платформы		
Входящие записи:	Спецификация требования на разработку ПО		
Исходящие записи:	Функционирующий программный код, реализующий WEB сервис, описанный в спецификации		

ШАГ 1. [TM119.01] ОПРЕДЕЛЕНИЕ ЦЕЛЕВЫХ МОДУЛЕЙ

- ☒ Определите модули проекта в которых будет реализован функционал WEB-сервиса.

ШАГ 2. [TM119.02] СОЗДАНИЕ ДОМЕННЫХ ОБЪЕКТОВ

- ☐ Если в функциональных требованиях по созданию сервиса описан раздел определяющий структуру доменных объектов, используемых сервисом, и эти объекты не созданы в других модулях, разработанных ранее, необходимо выполнить создание, недостающих доменных объектов.

ШАГ 3. [TM119.03] ПОДГОТОВКА К РЕАЛИЗАЦИИ WEB-СЕРВИСА

- ☐ Ознакомьтесь со спецификацией технологии JAX-WS. Документация доступна по ссылке: <http://docs.oracle.com/javaee/7/tutorial/doc/webservices-intro.htm#GIJTI>

Java API for XML Web Services (JAX-WS) - это прикладной программный интерфейс языка Java для создания веб-служб, являющийся частью платформы Java EE. JAX-WS является заменой технологии JAX-RPC, предоставляя более документо-ориентированную модель сообщений и упрощая разработку веб-служб за счёт использования аннотаций, впервые появившихся в Java SE 5. Технология JAX-WS является стандартом и описана в JSR 224.

Никаких особенностей специфичных для платформы ECMP5 при реализации -служб нет.

ШАГ 4. [TM119.04] ИМЕНОВАНИЕ СЕРВИСА

- ☐ Если сервис еще не создан, создайте его.
- ☒ Создайте в сервисе метод с именем, согласно функциональным требованиям.

ШАГ 5. [TM119.05] ЗАДАНИЕ ПАРАМЕТРОВ

- ☒ Опишите входящие параметры сервиса согласно таблице входящих параметров из функциональных требований
- ☒ Опишите выходящие параметры сервиса и коды возврата согласно таблице выходящих параметров из функциональных требований. Для кодов возврата используйте общую таблицу сквозных кодов возврата для всех веб-сервисов приложения.

ШАГ 6. [TM119.06] ОПРЕДЕЛЕНИЕ СТРУКТУРЫ ОБМЕНА ФАЙЛОВ ВЛОЖЕНИЙ

- Файлы вложений должны передаваться как отдельный объект. Не передавайте бинарные файлы внутри XML пакета.

ШАГ 7. [TM119.07] ОПРЕДЕЛЕНИЕ ПРАВИЛА АВТОРИЗАЦИИ

- В случае если доступ к веб-сервисам должен быть авторизованным, используйте авторизацию, предусмотренную технологией JAX-WS с использованием handler framework. Ознакомится с использованием этой технологии можно перейдя по ссылке:
<http://www.javaworld.com/article/2077679/soaet-a-handle-on-the-jax-ws-api-s-handler-framework/soa/get-a-handle-on-the-jax-ws-api-s-handler-framework.html>

ШАГ 8. [TM119.08] ОПРЕДЕЛЕНИЕ БИЗНЕС-ЛОГИКИ

- Бизнес-логика не должна реализовываться внутри сервиса. Функция, несущая полезную нагрузку должна быть реализована в соответствии с функциональными требованиями в отдельном сервисном классе. Функциональные требования описывают имя функции, входящие и исходящие параметры. В теле класса сервиса могут быть описаны только вспомогательные методы, предназначенные для валидации параметров.