

# Разработка REST-сервисов

*Версия 3.1.0*

**"ЦФТ-Платформа Развития"**

## Оглавление

<b>1</b>	<b>Термины и определения.....</b>	<b>1</b>
<b>2</b>	<b>Разработка REST-сервисов в режиме поставщика.....</b>	<b>2</b>
2.1	Возможности приложения.....	3
2.2	Действия, необходимые для установки приложения и начала работы с ним.....	4
2.3	Особенности текущей версии приложения.....	5
2.4	Публикация ТБП как REST-сервисов.....	7
2.4.1.....	Регистрация ТБП.....	7
2.4.2.....	Регистрация методов CRUD.....	8
2.4.3.....	Регистрация дополнительных методов.....	13
2.4.4.....	Требования к прикладным операциям, функциям, процедурам.....	13
2.4.5.....	Версионность.....	14
2.4.6.....	Получение метаинформации о зарегистрированном сервисе.....	15
2.5	Проверка работоспособности зарегистрированного сервиса.....	29
2.5.1.....	Примеры.....	29
<b>3</b>	<b>Работа с REST-сервисами в режиме потребителя.....</b>	<b>33</b>
3.1	Возможности приложения.....	34
3.2	Действия, необходимые для установки приложения и начала работы с ним.....	35
3.3	Особенности текущей версии приложения.....	36
3.4	Использование приложения.....	37
3.4.1.....	Регистрация сервиса.....	37
3.4.2.....	Реализация прикладной операции, вызывающей сервис.....	37
3.4.3.....	Использование приложения в асинхронном режиме.....	41
<b>4</b>	<b>Справочник "Веб-сервисы. Журнал выполнения".....</b>	<b>42</b>
<b>5</b>	<b>Приложение 1 .....</b>	<b>43</b>
<b>6</b>	<b>Приложение 2 .....</b>	<b>45</b>
<b>7</b>	<b>Приложение 3 .....</b>	<b>46</b>
<b>8</b>	<b>Приложение 4 .....</b>	<b>48</b>

# 1 Термины и определения

**Система** - любая система, реализованная на "ЦФТ - Платформа Развития": "ЦФТ-Банк", "ЦФТ-Ритейл банк", "ЦФТ-Страховая компания".

**2MCA** - Платформа 2 MCA (Mission Critical Application) – технологическая платформа с использованием 3-х уровневой архитектуры.

**ЦФТ Веб-шлюз** - системное приложение для организации взаимодействия с внешними системами по протоколу http/https.

**REST-сервисы** - Web-сервисы, ориентированные на системные ресурсы, включая способы обработки и передачи состояний ресурсов по HTTP разнообразными клиентскими приложениями, написанными на различных языках программирования.

**ТБП** - Тип Базового Понятия - любой тип данных (ТБП, структура, справочник, продукт), имеющий экземпляры.

**REST API** - интерфейс для программирования приложений – набор готовых методов внутри протокола http для общения с сервером (GET,POST,PUT и т.д.), предоставляемых приложением для использования во внешних программных продуктах.

**URI** - унифицированный идентификатор ресурса - последовательность символов, идентифицирующая абстрактный или физический ресурс.

**JSON** - текстовый формат обмена данными, основанный на JavaScript.

**XML** - текстовый формат обмена данными, состоящий из набора условных меток (тегов).

**Advanced Rest Client** - отдельное приложение или расширение Google Chrome для веб-разработчиков, позволяющее делать запросы к RestFul сервисам, получать ответы от сервера.

## 2 Разработка REST-сервисов в режиме поставщика

Приложение ["REST. Поставщик веб-служб"](#) предоставляет возможность публиковать в Системе в качестве REST-сервисов модули, обращение к которым доступно из других приложений.

Использование Приложения возможно только совместно с Приложением ["ЦФТ Веб-шлюз"](#) не ниже версии 1.6.2.

## 2.1 Возможности приложения

Приложение ["REST. Поставщик веб-служб"](#) позволяет опубликовать любой ТБП Системы как REST-сервис для получения доступа к нему из других приложений. При этом для ТБП будут доступны действия CRUD (create, read, update, delete - *создание, чтение, обновление, удаление*). Кроме того, есть возможность опубликовать свободные действия, например, групповые, простые операции этого ТБП или процедуры/функции из библиотек.

Приложение поддерживает версию REST API. При необходимости расширения API, можно опубликовать новые версии для действий CRUD и свободных действий, при этом предыдущие версии останутся без изменений.

### Использование HTTP методов

В приложении используются 4 основных HTTP метода: PUT, GET, POST, DELETE. Каждый из методов служит для выполнения определённого ему действия из CRUD (*create, read, update, delete* — «создание, чтение, обновление, удаление»).

Соответствие HTTP методов действиям CRUD:

1. HTTP метод **PUT** соответствует действию **Create**. Если для операции необходимы параметры, они передаются в теле запроса.
2. HTTP метод **GET** соответствует действию **Read**. В строке запроса могут быть переданы уточнения. Например, фильтр в формате oData. Он передаётся в параметре FILTER. Например, ?FILTER=id EQ 3064617626. Полный список возможностей описан в [Приложении 1](#) <sup>[43]</sup>.
3. HTTP метод **POST** соответствует действию **Update**. Если для операции необходимы параметры, они передаются в теле запроса.
4. HTTP метод **DELETE** соответствует действию **Delete**. Если для операции необходимы параметры, они передаются в теле запроса.

**Строка соединения** в общем случае имеет следующий вид:

http:// [имя и порт сервера] / [webproxy]/REST/{номер версии}/{URI }? {запрос}

, где

**имя и порт сервера** - имя и порт сервера, на котором развернут ЦФТ Веб-шлюз,

**webproxy** – имя веб-приложения для webProxy,

**REST** – namespace, который возвращает операция, указанная в settings\_name файла webproxy-settings.xml. Это константы, которые настраиваются один раз и впоследствии не меняются,

**номер версии** - необязательный компонент. Если номер версии не указан, определяется последняя версия для ТБП.

**URI** - URI зарегистрированного ТБП.

**запрос** - необязательный компонент, который служит для идентификации ресурса в пределах URI.

Пример:

[http://book187vt1:8095/webproxy/REST/REST\\_ACT\\_IMPL?meta](http://book187vt1:8095/webproxy/REST/REST_ACT_IMPL?meta) <sup>[3]</sup> (где book187vt1:8095 – имя и порт сервера, на котором развёрнут ЦФТ Веб-шлюз, REST\_ACT\_IMPL – URI зарегистрированного ТБП.

## 2.2 Действия, необходимые для установки приложения и начала работы с ним

Для работы с приложением ["REST. Поставщик веб-служб"](#) предварительно нужно выполнить действия:

1. Установить приложение ["ЦФТ Веб - шлюз"](#).
2. В конфигурационном файле webproxy-settings.xml указать значение параметра settings\_name:  

```
<entry key="settings_name"> WEB_EXTENSIONS:PLP_DADS:DISTR_PLP_DADS</entry>
```

- если в лицензии на веб-шлюз установлено ограничение на выполняемые методы (опция 2MCA.WEBPROXY.METHODS = WEB\_EXTENSIONS.DISTR\_CALLS), т.е. с помощью веб-шлюза можно взаимодействовать только дистрибутивными точками вызова или  

```
<entry key="settings_name"> WEB_EXTENSIONS:PLP_DADS:PLP_DADS</entry>
```

- если ограничение на выполняемые методы не установлено
3. Выполнить настройки в справочнике "Настройки." (FP\_TUNE). Подробнее см. в [Приложении 3](#) <sup>46</sup>.
4. Для запуска веб-шлюза пользователю необходимо предоставить права на:  
ТБП:
  - WEB\_EXTENSIONS
  - REST\_CONS
  - REST\_PROVОперации:
  - WEB\_EXTENSIONS.PLP\_DADS
  - WEB\_EXTENSIONS.DISTR\_CALLS
  - REST\_CONS.WS\_MAIN
  - REST\_PROV.WS\_MAIN

Кроме того, при работе с представлениями в случае GET запросов, пользователю необходимо предоставить права на эти представления.

### Внимание!

Для работы с приложением необходимо наличие приложений:

- для "ЦФТ - Банк" - ["Интеграция на основе механизмов сопряжения Oracle"](#)
- для "ЦФТ-Ритейл банк" - "Универсальный шлюз - Базовый функционал"
- для "ЦФТ-Страховая компания" - ["Интеграция на основе механизмов сопряжения Oracle"](#)

Также необходимо убедиться, что установлен pl/json. Pl/json устанавливается автоматически при установке приложения с помощью APM "Программа установки обновления".

## 2.3 Особенности текущей версии приложения

Текущая версия приложения имеет следующие особенности:

1. Версия API определяется из URL. Из заголовков запроса версия не вычитывается. Если в URL не указана версия, берётся последняя для ТБП.
2. В качестве HTTP заголовка CONTENT\_TYPE допустимы application/json и multipart/form-data.
  - 2.1. Если заголовок имеет тип содержимого application/json, то тело запроса должно быть в формате JSON.
  - 2.2. Если заголовок имеет тип содержимого multipart/form-data – допускается передача текстовых полей и файлов.
3. В качестве заголовка ACCEPT поддерживается application/json. Если тип содержимого заголовка ACCEPT отличается от application/json, то функция не должна иметь OUT параметров, и может возвращать BLOB. В таком случае возвращаемое значение будет помещено в качестве приложенного файла.
4. Если для методов PUT, POST, DELETE необходимы параметры, они передаются в теле запроса. Это может быть JSON или текстовые поля и файлы. При этом переданные файлы преобразуются в одноименные параметры операции типа BLOB, а текстовые поля – в одноименные строковые параметры.
5. Если CONTENT\_TYPE application/json, в качестве методов для PUT, POST, DELETE могут быть указаны как операции, так и процедуры/функции.
6. Для CONTENT\_TYPE multipart/form-data использование операций недопустимо.
7. Доступна мета-информация по ТБП. Для этого нужно указать в строке [ТБП]?meta, например, MAIN\_DOCUM?meta. При этом мета-информация выдаётся в json в специфичном формате для всех версий API. Пример вывода мета-информации см. в [Получение метаинформации о зарегистрированном сервисе](#)<sup>[15]</sup>. В следующей версии реализации ядра REST-сервисов планируется выводить мета-информацию в формате OPEN API [https://ru.wikipedia.org/wiki/OpenAPI\\_\(%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F\)](https://ru.wikipedia.org/wiki/OpenAPI_(%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F))<sup>[5]</sup>
8. Если прикладная операция простая и должна быть выполнена над конкретным экземпляром, идентификатор экземпляра можно передать в URL в параметре id. Например, MAIN\_DOCUM?id=3064617626. В этом случае будет произведён поиск экземпляра в ТБП MAIN\_DOCUM. Если он не будет найден, будет отправлена ошибка 404. Иначе операция будет выполнена над найденным экземпляром.
9. В качестве параметров вызываемых модулей и возвращаемых типов могут быть типы ORACLE, скалярные типы ЦФТ (число, дата, логика, строка), ссылки на ТБП, а также конструкции вида table of, record(). Не поддерживается работа с массивами Системы. При этом проверка корректности параметров операций/процедур осуществляется на этапе обработки запросов. На этапе добавления/изменения методов проверяется только наличие указанного модуля.
10. Кроме того, поддержана возможность передачи файлов. При этом функция/процедура может иметь строковые параметры и BLOB параметры
11. Если какой-то параметр не заполняется в запросе, вместо него в прикладную операцию будет передано пустое значение.
12. Есть возможность передачи тела запроса и формирования тела ответа напрямую в функции. Для этого функция должна иметь один входной параметр типа json и отдавать в

ответ json. Разбор тела запроса и формирование тела ответа формируются в прикладной функции. В этом случае запрещается использование операций.

13. Для метода GET заголовок ACCEPT может иметь только тип содержимого application/json.
14. При использовании метода GET при обращении к ТБП есть возможность наложения фильтра и других параметров в формате oData. Описание поддерживаемых возможностей см. в [Приложении 1](#)<sup>43</sup>.



## 2.4 Публикация ТБП как REST-сервисов

Для публикации ТБП необходимо выполнить следующие действия:

1. [Зарегистрировать ТБП.](#)<sup>[7]</sup>
2. [Зарегистрировать операции, соответствующие методам CRUD HTTP запроса.](#)<sup>[8]</sup>
3. [При необходимости зарегистрировать дополнительные операции.](#)<sup>[13]</sup>

Прикладные операции, указанные как методы CRUD или свободные действия, должны соответствовать описанным [требованиям](#)<sup>[13]</sup>.

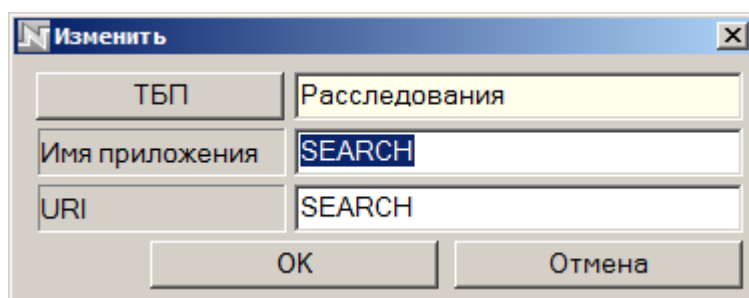
При регистрации операций учитывается [версионность](#)<sup>[14]</sup>.

### 2.4.1 Регистрация ТБП

Для регистрации ТБП нужно в типе "ВЕБ-службы. Поставщик REST" выполнить операцию "Добавить" и указать публикуемый ТБП.

Режим работы: "Веб-расширения" -> "ВЕБ-службы. Поставщик REST"

Операции "Добавить" и "Изменить"



Параметры экранной формы:

- ✓ **ТБП** - публикуемый ТБП, параметр обязательный для заполнения.
- ✓ **Имя приложения** - информационное поле, используется для описания приложения.
- ✓ **URI** - унифицированный идентификатор ресурса, используется в строке запроса.

**Имя приложения** и **URI** автоматически заполняются коротким именем ТБП. При необходимости их можно изменить.

## 2.4.2 Регистрация методов CRUD

После регистрации ТБП необходимо описать операции, соответствующие методам CRUD HTTP запроса.

Для этого необходимо перейти по колонке "Версии API" представления в массив "REST-сервисы. CRUD Реализации" и создать запись. В записи обязательно нужно указать номер версии, выбрать представление и операции для всех методов.

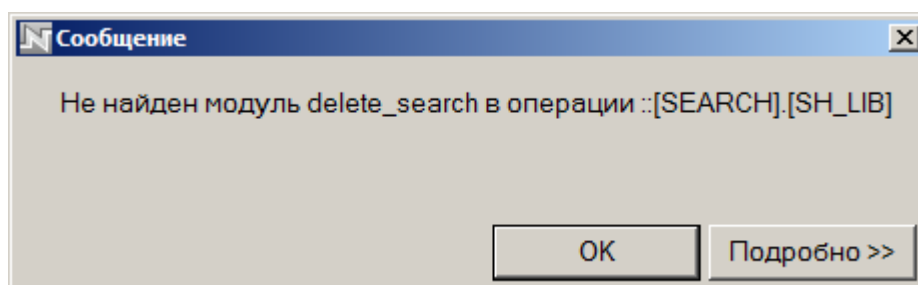
Режим работы: *"REST-сервисы. CRUD Реализации"*

Операции *"Добавить"* и *"Изменить"*

Параметры экранной формы:

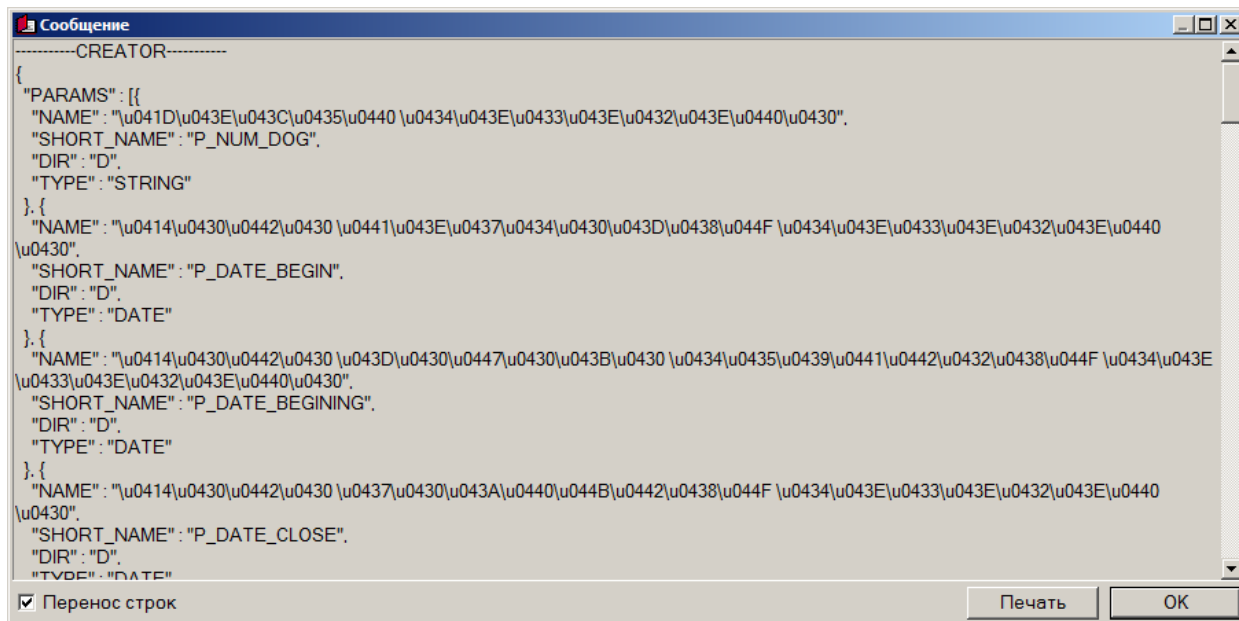
- ✓ **Версия** - номер версии, параметр обязательный для заполнения.
- ✓ **Конструктор** - операция, функция или процедура для создания экземпляра ТБП.
- ✓ **Деструктор** - операция, функция или процедура для удаления экземпляра ТБП.
- ✓ **Редактор** - операция, функция или процедура для редактирования экземпляра ТБП.

В качестве операции может быть указана функция или процедура библиотеки. Указанная функция/процедура должна быть объявлена в выбранной библиотеке как public, в противном случае возникает ошибка:



- ✓ **Представление для GET** - представление, колонки которого будут указываться в методе GET для получения информации об экземпляре ТБП.
- ✓ **Прямой доступ** - признак означает, что в теле запроса и ответа передаются данные в формате JSON. Признак устанавливается для функций, имеющих один входной параметр типа JSON, и возвращающих JSON в результате своей работы.

После нажатия «ОК» выводится буфер сессии, в котором содержится описание созданного или изменённого API.



## Пример буфера сессии

```

-----CREATOR-----
{
  "PARAMS": [{
    "NAME": "\u041D\u043E\u043C\u0435\u0440 \u0434\u043E\u0435\u0433\u043E\u0432\u0440\u043E\u0432\u0440\u0430",
    "SHORT_NAME": "P_NUM_DOG",
    "DIR": "D",
    "TYPE": "STRING"
  }, {
    "NAME": "\u0414\u0430\u0442\u0430 \u0441\u043E\u0434\u0435\u0437\u0434\u0430\u043D\u0438\u0442\u044C \u0434\u043E\u0433\u0430\u0434\u0430",
    "SHORT_NAME": "P_DATE_BEGIN",
    "DIR": "D",
    "TYPE": "DATE"
  }, {
    "NAME": "\u0414\u0430\u0442\u0430 \u0441\u043E\u0434\u0435\u0437\u0434\u0430\u043D\u0438\u0442\u044C \u0434\u043E\u0433\u0430\u0434\u0430 \u0434\u043E\u0433\u0430\u0434\u0430",
    "SHORT_NAME": "P_DATE_BEGINING",
    "DIR": "D",
    "TYPE": "DATE"
  }, {
    "NAME": "\u0414\u0430\u0442\u0430 \u0441\u043E\u0434\u0435\u0437\u0434\u0430\u043D\u0438\u0442\u044C \u0434\u043E\u0433\u0430\u0434\u0430 \u0434\u043E\u0433\u0430\u0434\u0430",
    "SHORT NAME": "P DATE CLOSE",

```

```

"DIR" : "D",
"TYPE" : "DATE"
},{
"NAME" : "\u0414\u0430\u0442\u0430 \u0435\u043a\u0430\u0435\u043d\u0447\u0430\u043d\u0438\u044f
\u0434\u0435\u0439\u0441\u0441\u0442\u0432\u0438\u044f \u0434\u0435\u0435\u0432\u0435\u0440
\u0430\u0430",
"SHORT_NAME" : "P_DATE_ENDING",
"DIR" : "D",
"TYPE" : "DATE"
},{
"NAME" : "\u0421\u0435\u0437\u0434\u0430\u0430\u043d\u0435 \u0435\u0435\u0435\u0435\u0435\u0435
\u0432\u0432\u0435\u0435\u0435\u0435\u0435\u0435",
"SHORT_NAME" : "P_CREATE_USER",
"DIR" : "D",
"TYPE" : "REFERENCE"
},{
"NAME" : "\u0415\u0440\u0438\u0435\u0435\u0447\u0435\u0447\u0430\u043d\u0438\u044f",
"SHORT_NAME" : "P_NOTES",
"DIR" : "D",
"TYPE" : "MEMO"
},{
"NAME" : "\u0415\u0435\u0435\u0435\u0449\u0435\u043d\u0435\u0442\u0435\u0435 \u0441\u0441\u0442\u0430
\u0442\u0443\u0441\u0441 \u0435\u0435\u0435\u0435\u0443\u0430\u0442\u0430",
"SHORT_NAME" : "P_COM_STATUS",
"DIR" : "D",
"TYPE" : "REFERENCE"
},{
"NAME" : "\u0424\u0438\u0435\u0438\u0438\u0430\u0430",
"SHORT_NAME" : "P_FILIAL",
"DIR" : "D",
"TYPE" : "REFERENCE"
},{
"NAME" : "\u0415\u0435\u0435\u0435\u0440\u0430\u0437\u0434\u0435\u0435\u0435\u0435\u0435\u0435",
"SHORT_NAME" : "P_DEPART",
"DIR" : "D",
"TYPE" : "REFERENCE"
},{
"NAME" : "\u0421\u0435\u0434\u0435\u0440\u0435\u0440\u0435\u0435\u0435 \u0435\u0435\u0435\u0435\u0435
\u0440\u0430\u0435\u0438\u0438",
"SHORT_NAME" : "P_SH_CONTENT",
"DIR" : "D",
"TYPE" : "MEMO"
},{
"NAME" : "\u0421\u0443\u0435\u0435\u0435\u0430",
"SHORT_NAME" : "P_SH_SUM",
"DIR" : "D",
"TYPE" : "NUMBER"
},{
"NAME" : "\u0412\u0430\u0435\u0435\u0445\u0445\u0442\u0430",
"SHORT_NAME" : "P_SH_CUR",
"DIR" : "D",
"TYPE" : "REFERENCE"
},{
"NAME" : "\u0413\u0440\u0443\u0435\u0435\u0435\u0435\u0435 \u0440\u0440\u0430\u0441\u0441\u0435\u0435\u0435
\u0435\u0435\u0435\u0435\u0435\u0435\u0438\u044f",
"SHORT_NAME" : "P_SH_GROUP",
"DIR" : "D",
"TYPE" : "REFERENCE"
},{

```





Далее все обращения к REST-сервису будут отражаться в справочнике ["Веб-сервисы. Журнал выполнения"](#)<sup>[42]</sup>.

### 2.4.3 Регистрация дополнительных методов

Кроме методов CRUD, которых может не хватить для реализации полноценной бизнес-логики, можно зарегистрировать свободные операции.

Для зарегистрированного ТБП необходимо перейти по колонке "Свободные действия" представления в массив "REST-сервисы. Произвольные операции. Реализации" и создать запись аналогично тому, как это было сделано для методов CRUD. В записи обязательно нужно указать версию.

**Режим работы: "REST-сервисы. Произвольные операции. Реализации"**

**Операции "Добавить" и "Изменить"**

**Параметры экранной формы:**

- ✓ **Версия** - номер версии, параметр обязательный для заполнения.
  - ✓ **URI** - унифицированный идентификатор свободной операции, параметр обязательный для заполнения.
- Для одного URI свободной операции возможна регистрация нескольких версий.
- ✓ **Операция** - простая или групповая операция ТБП, а также функция/процедура библиотеки. Функции/процедуры должны быть описаны как глобальные.
  - ✓ **Прямой доступ** - признак означает, что в теле запроса передаются данные в формате JSON. Признак устанавливается для функций, имеющих один входной параметр типа JSON, и возвращающих JSON в результате своей работы.

### 2.4.4 Требования к прикладным операциям, функциям, процедурам

#### Для операций с прямым доступом

Если зарегистрированный метод помечен признаком "Прямой доступ", то операция, функция или процедура должна иметь только один входной параметр типа JSON и возвращать JSON в качестве результата.

При этом HTTP заголовки CONTENT\_TYPE (влияет на входные параметры) и ACCEPT (влияет на выходные параметры) должны иметь тип содержимого application/json

### **Для всех остальных операций**

#### **HTTP заголовок CONTENT\_TYPE имеет тип содержимого application/json**

Если прикладная операция, функция или процедура указана как один из методов CRUD или свободное действие, она может иметь входные параметры: скалярные типы ORACLE, скалярные типы ЦФТ, ссылки, а также конструкции вида table of, record().

#### **HTTP заголовок АСCEPT имеет тип содержимого application/json**

Если прикладная операция, функция или процедура указана как один из методов CRUD или свободное действие, она может иметь выходные параметры: скалярные типы ORACLE, скалярные типы ЦФТ, ссылки, а также конструкции вида table of, record().

#### **HTTP заголовок CONTENT\_TYPE имеет тип содержимого multipart/form-data**

В качестве методов CRUD допускается использование только процедур и функций. При этом функция/процедура в качестве входных параметров может иметь строковые параметры и BLOB параметры.

#### **HTTP заголовок АСCEPT отличается от application/json**

Процедура / функция не должна иметь OUT параметров и может возвращать BLOB в качестве приложенного файла.

При этом заголовки CONTENT-TYPE и АСCEPT могут сочетаться между собой различным образом.

## **2.4.5 Версионность**

Для одного ТБП может быть зарегистрировано несколько действующих версий REST API. В случае изменения API можно зарегистрировать новые версии для действий CRUD и свободных действий, при этом старые версии оставить без изменений.

Например, вместо вызова дистрибутивных конструктора, деструктора и редактора можно создать библиотеку, в которой разместить функции/процедуры, и создать 2-ю версию сервиса.



## 2.4.6 Получение метайнформации о зарегистрированном сервисе

Для того чтобы получить метайнформацию о зарегистрированном сервисе, необходимо сделать запрос GET, например, <http://book187vt1:8095/webproxy/REST/SEARCH?meta><sup>[15]</sup>. Ответ будет содержать информацию в формате JSON с описанием сервиса.

Пример описания. В приведенном примере можно увидеть 2 версии сервиса для ТБП и одно свободное действие.

```
{
  "METAClass_REF": {
    "URI": "SEARCH",
    "ID": "SEARCH",
    "NAME": "Расследования"
  },
  "API_VERSIONS": [
    {
      "VERSION_NUM": "2",
      "ACTIONS": {
        "PUT": {
          "PARAMS": [
            {
              "NAME": "",
              "SHORT_NAME": "ReturnValue",
              "DIR": "O",
              "TYPE": "REFERENCE"
            },
            {
              "NAME": "",
              "SHORT_NAME": "P_VID",

```

```

        "DIR": "I",
        "TYPE": "REFERENCE"
    },
    ],
    "CFT_PATH": "[SEARCH].[SH_LIB].create_search"
},
"POST": {
    "PARAMS": [
        {
            "NAME": "Номер договора",
            "SHORT_NAME": "P_NUM_DOG",
            "DIR": "I",
            "TYPE": "STRING"
        }, {
            "NAME": "Дата создания договора",
            "SHORT_NAME": "P_DATE_BEGIN",
            "DIR": "I",
            "TYPE": "DATE"
        }, {
            "NAME": "Дата начала действия договора",
            "SHORT_NAME": "P_DATE_BEGINING",
            "DIR": "I",
            "TYPE": "DATE"
        }, {
            "NAME": "Дата закрытия договора",
            "SHORT_NAME": "P_DATE_CLOSE",
            "DIR": "I",
            "TYPE": "DATE"
        }, {
            "NAME": "Дата окончания действия
договора",
            "SHORT_NAME": "P_DATE_ENDING",
            "DIR": "I",
            "TYPE": "DATE"
        }, {
            "NAME": "Создано пользователем",
            "SHORT_NAME": "P_CREATE_USER",
            "DIR": "I",
            "TYPE": "REFERENCE"
        }, {
            "NAME": "Примечания",
            "SHORT_NAME": "P_NOTES",
            "DIR": "I",
            "TYPE": "MEMO"
        }, {
            "NAME": "Обобщенный статус продукта",
            "SHORT_NAME": "P_COM_STATUS",
            "DIR": "I",
            "TYPE": "REFERENCE"
        }, {
            "NAME": "Филиал",
            "SHORT_NAME": "P_FILIAL",
            "DIR": "I",
            "TYPE": "REFERENCE"
        }, {
            "NAME": "Подразделение",
            "SHORT_NAME": "P_DEPART",
            "DIR": "I",
            "TYPE": "REFERENCE"
        }
    ]
}

```

```

        "NAME": "Содержание операции",
        "SHORT_NAME": "P_SH_CONTENT",
        "DIR": "I",
        "TYPE": "MEMO"
    }, {
        "NAME": "Сумма",
        "SHORT_NAME": "P_SH_SUM",
        "DIR": "I",
        "TYPE": "NUMBER"
    }, {
        "NAME": "Валюта",
        "SHORT_NAME": "P_SH_CUR",
        "DIR": "I",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Группа расследования",
        "SHORT_NAME": "P_SH_GROUP",
        "DIR": "I",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Вид расследования",
        "SHORT_NAME": "P_SH_VID",
        "DIR": "I",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Маршрут",
        "SHORT_NAME": "P_SH_ROUTE",
        "DIR": "I",
        "TYPE": "REFERENCE"
    }
    ],
    "CFT_PATH": "[SEARCH].[EDIT_AUTO]"
},
"DEL": {
    "CFT_PATH": "[SEARCH].[DELETE_AUTO]"
},
"GET": {
    "COLUMNS": [
        {
            "NAME": "CLASS_ID",
            "POSITION": 0,
            "WIDTH": 10,
            "UNVISIBLE": "1",
            "ALIAS": "CLASS_ID",
            "REFERENCE_ID": "",
            "TARGET_CLASS_ID": ""
        }, {
            "NAME": "COLLECTION_ID",
            "POSITION": 0,
            "WIDTH": 10,
            "UNVISIBLE": "1",
            "ALIAS": "COLLECTION_ID",
            "REFERENCE_ID": "",
            "TARGET_CLASS_ID": ""
        }, {
            "NAME": "STATE_ID",
            "POSITION": 0,
            "WIDTH": 10,
            "UNVISIBLE": "1",
            "ALIAS": "STATE_ID",

```

```

"REFERENCE_ID": "",
"TARGET_CLASS_ID": ""
}, {
  "NAME": "Расследования # ID",
  "POSITION": 1,
  "WIDTH": 18,
  "UNVISIBLE": "2",
  "ALIAS": "C_ID",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Номер расследования",
  "POSITION": 2,
  "WIDTH": 12,
  "UNVISIBLE": "0",
  "ALIAS": "C_NUM_DOG",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Вид расследования",
  "POSITION": 3,
  "WIDTH": 25,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_VID#SHV_NAME",
  "REFERENCE_ID": "a1.C_SH_VID",
  "TARGET_CLASS_ID": "SH_VID"
}, {
  "NAME": "Дата создания расследования",
  "POSITION": 4,
  "WIDTH": 11,
  "UNVISIBLE": "0",
  "ALIAS": "C_DATE_BEGIN",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Плановая дата окончания",
  "POSITION": 5,
  "WIDTH": 16,
  "UNVISIBLE": "0",
  "ALIAS": "C_DATE_ENDING",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Содержание операции",
  "POSITION": 6,
  "WIDTH": 28,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_CONTENT",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Обобщенный статус",
  "POSITION": 7,
  "WIDTH": 23,
  "UNVISIBLE": "0",
  "ALIAS": "C_COM_STATUS#NAME",
  "REFERENCE_ID": "a2.C_COM_STATUS",
  "TARGET_CLASS_ID": "COM_STATUS_PRD"
}, {
  "NAME": "Исходящие сообщения",

```

```
        "POSITION": 8,  
        "WIDTH": 18,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_OUTMSG",  
        "REFERENCE_ID": "",  
        "TARGET_CLASS_ID": "SH_OUTMSG"  
    }, {  
        "NAME": "Объекты",  
        "POSITION": 9,  
        "WIDTH": 10,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_OBJ",  
        "REFERENCE_ID": "",  
        "TARGET_CLASS_ID": ""  
    }, {  
        "NAME": "Номера договоров",  
        "POSITION": 10,  
        "WIDTH": 18,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_NUM_DOG_1",  
        "REFERENCE_ID": "",  
        "TARGET_CLASS_ID": ""  
    }, {  
        "NAME": "Субъекты",  
        "POSITION": 11,  
        "WIDTH": 11,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_SUBJ",  
        "REFERENCE_ID": "",  
        "TARGET_CLASS_ID": ""  
    }, {  
        "NAME": "Группа расследования",  
        "POSITION": 12,  
        "WIDTH": 30,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_SEARCH_GROUP",  
        "REFERENCE_ID": "a1.C_SH_GROUP",  
        "TARGET_CLASS_ID": "SH_GROUP"  
    }, {  
        "NAME": "Ответственные",  
        "POSITION": 13,  
        "WIDTH": 18,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_OFFICER",  
        "REFERENCE_ID": "",  
        "TARGET_CLASS_ID": ""  
    }, {  
        "NAME": "Отметки",  
        "POSITION": 14,  
        "WIDTH": 18,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_MARKS",  
        "REFERENCE_ID": "",  
        "TARGET_CLASS_ID": ""  
    }, {  
        "NAME": "Дата закрытия расследования",  
        "POSITION": 15,  
        "WIDTH": 14,  
        "UNVISIBLE": "0",  
        "ALIAS": "C_DATE_CLOSE",
```

```

"REFERENCE_ID": "",
"TARGET_CLASS_ID": ""
}, {
  "NAME": "Примечания",
  "POSITION": 16,
  "WIDTH": 24,
  "UNVISIBLE": "0",
  "ALIAS": "C_NOTES",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Создано пользователем",
  "POSITION": 17,
  "WIDTH": 23,
  "UNVISIBLE": "0",
  "ALIAS": "C_CREATE_USER#NAME",
  "REFERENCE_ID": "a2.C_CREATE_USER",
  "TARGET_CLASS_ID": "USER"
}, {
  "NAME": "Сумма",
  "POSITION": 18,
  "WIDTH": 14,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_SUM",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Валюта",
  "POSITION": 19,
  "WIDTH": 30,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_CUR#CUR_SHORT",
  "REFERENCE_ID": "a1.C_SH_CUR",
  "TARGET_CLASS_ID": "FT_MONEY"
}, {
  "NAME": "Ответственный",
  "POSITION": 20,
  "WIDTH": 30,
  "UNVISIBLE": "0",
  "ALIAS": "C_SHOF_USER#NAME",
  "REFERENCE_ID": "b1.C_SHOF_USER",
  "TARGET_CLASS_ID": "USER"
}, {
  "NAME": "Решение",
  "POSITION": 21,
  "WIDTH": 7,
  "UNVISIBLE": "0",
  "ALIAS": "C_DECISION",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Подразделение",
  "POSITION": 22,
  "WIDTH": 19,
  "UNVISIBLE": "0",
  "ALIAS": "C_DEPART_CODE",
  "REFERENCE_ID": "a2.C_DEPART",
  "TARGET_CLASS_ID": "DEPART"
}, {
  "NAME": "Подразделение ID",

```

```

        "POSITION": 23,
        "WIDTH": 13,
        "UNVISIBLE": "2",
        "ALIAS": "C_DEPART",
        "REFERENCE_ID": "",
        "TARGET_CLASS_ID": "DEPART"
    }, {
        "NAME": "Сообщения РСФМ",
        "POSITION": 24,
        "WIDTH": 18,
        "UNVISIBLE": "0",
        "ALIAS": "C_RSFM_MESS",
        "REFERENCE_ID": "",
        "TARGET_CLASS_ID": ""
    }
],
"VIEW_DETAILS": {
    "NAME": "Полный список",
    "SHORT_NAME": "VW_CRIT_ALL_PLPLUS"
}
}
}
}, {
    "VERSION_NUM": "1",
    "ACTIONS": {
        "PUT": {
            "PARAMS": [
                {
                    "NAME": "Номер договора",
                    "SHORT_NAME": "P_NUM_DOG",
                    "DIR": "D",
                    "TYPE": "STRING"
                }, {
                    "NAME": "Дата создания договора",
                    "SHORT_NAME": "P_DATE_BEGIN",
                    "DIR": "D",
                    "TYPE": "DATE"
                }, {
                    "NAME": "Дата начала действия договора",
                    "SHORT_NAME": "P_DATE_BEGINING",
                    "DIR": "D",
                    "TYPE": "DATE"
                }, {
                    "NAME": "Дата закрытия договора",
                    "SHORT_NAME": "P_DATE_CLOSE",
                    "DIR": "D",
                    "TYPE": "DATE"
                }, {
                    "NAME": "Дата окончания действия
договора",
                    "SHORT_NAME": "P_DATE_ENDING",
                    "DIR": "D",
                    "TYPE": "DATE"
                }, {
                    "NAME": "Создано пользователем",
                    "SHORT_NAME": "P_CREATE_USER",
                    "DIR": "D",
                    "TYPE": "REFERENCE"
                }, {
                    "NAME": "Примечания",

```

```

        "SHORT_NAME": "P_NOTES",
        "DIR": "D",
        "TYPE": "MEMO"
    }, {
        "NAME": "Обобщенный статус продукта",
        "SHORT_NAME": "P_COM_STATUS",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Филиал",
        "SHORT_NAME": "P_FILIAL",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Подразделение",
        "SHORT_NAME": "P_DEPART",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Содержание операции",
        "SHORT_NAME": "P_SH_CONTENT",
        "DIR": "D",
        "TYPE": "MEMO"
    }, {
        "NAME": "Сумма",
        "SHORT_NAME": "P_SH_SUM",
        "DIR": "D",
        "TYPE": "NUMBER"
    }, {
        "NAME": "Валюта",
        "SHORT_NAME": "P_SH_CUR",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Группа расследования",
        "SHORT_NAME": "P_SH_GROUP",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Вид расследования",
        "SHORT_NAME": "P_SH_VID",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }, {
        "NAME": "Маршрут",
        "SHORT_NAME": "P_SH_ROUTE",
        "DIR": "D",
        "TYPE": "REFERENCE"
    }
    ],
    "CFT_PATH": "[SEARCH].[NEW_AUTO]"
},
"POST": {
    "PARAMS": [
        {
            "NAME": "Номер договора",
            "SHORT_NAME": "P_NUM_DOG",
            "DIR": "I",
            "TYPE": "STRING"
        },
        {

```



договора",

```

"NAME": "Дата создания договора",
"SHORT_NAME": "P_DATE_BEGIN",
"DIR": "I",
"TYPE": "DATE"
}, {
"NAME": "Дата начала действия договора",
"SHORT_NAME": "P_DATE_BEGINING",
"DIR": "I",
"TYPE": "DATE"
}, {
"NAME": "Дата закрытия договора",
"SHORT_NAME": "P_DATE_CLOSE",
"DIR": "I",
"TYPE": "DATE"
}, {
"NAME": "Дата окончания действия
"SHORT_NAME": "P_DATE_ENDING",
"DIR": "I",
"TYPE": "DATE"
}, {
"NAME": "Создано пользователем",
"SHORT_NAME": "P_CREATE_USER",
"DIR": "I",
"TYPE": "REFERENCE"
}, {
"NAME": "Примечания",
"SHORT_NAME": "P_NOTES",
"DIR": "I",
"TYPE": "MEMO"
}, {
"NAME": "Обобщенный статус продукта",
"SHORT_NAME": "P_COM_STATUS",
"DIR": "I",
"TYPE": "REFERENCE"
}, {
"NAME": "Филиал",
"SHORT_NAME": "P_FILIAL",
"DIR": "I",
"TYPE": "REFERENCE"
}, {
"NAME": "Подразделение",
"SHORT_NAME": "P_DEPART",
"DIR": "I",
"TYPE": "REFERENCE"
}, {
"NAME": "Содержание операции",
"SHORT_NAME": "P_SH_CONTENT",
"DIR": "I",
"TYPE": "MEMO"
}, {
"NAME": "Сумма",
"SHORT_NAME": "P_SH_SUM",
"DIR": "I",
"TYPE": "NUMBER"
}, {
"NAME": "Валюта",
"SHORT_NAME": "P_SH_CUR",
"DIR": "I",
"TYPE": "REFERENCE"

```

```

    }, {
      "NAME": "Группа расследования",
      "SHORT_NAME": "P_SH_GROUP",
      "DIR": "I",
      "TYPE": "REFERENCE"
    }, {
      "NAME": "Вид расследования",
      "SHORT_NAME": "P_SH_VID",
      "DIR": "I",
      "TYPE": "REFERENCE"
    }, {
      "NAME": "Маршрут",
      "SHORT_NAME": "P_SH_ROUTE",
      "DIR": "I",
      "TYPE": "REFERENCE"
    }
  ],
  "CFT_PATH": "[SEARCH].[EDIT_AUTO]"
},
"DEL": {
  "CFT_PATH": "[SEARCH].[DELETE_AUTO]"
},
"GET": {
  "COLUMNS": [
    {
      "NAME": "CLASS_ID",
      "POSITION": 0,
      "WIDTH": 10,
      "UNVISIBLE": "1",
      "ALIAS": "CLASS_ID",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "COLLECTION_ID",
      "POSITION": 0,
      "WIDTH": 10,
      "UNVISIBLE": "1",
      "ALIAS": "COLLECTION_ID",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "STATE_ID",
      "POSITION": 0,
      "WIDTH": 10,
      "UNVISIBLE": "1",
      "ALIAS": "STATE_ID",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Расследования # ID",
      "POSITION": 1,
      "WIDTH": 18,
      "UNVISIBLE": "2",
      "ALIAS": "C_ID",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Номер расследования",
      "POSITION": 2,
      "WIDTH": 12,

```

```

"UNVISIBLE": "0",
"ALIAS": "C_NUM_DOG",
"REFERENCE_ID": "",
"TARGET_CLASS_ID": ""
}, {
  "NAME": "Вид расследования",
  "POSITION": 3,
  "WIDTH": 25,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_VID#SHV_NAME",
  "REFERENCE_ID": "a1.C_SH_VID",
  "TARGET_CLASS_ID": "SH_VID"
}, {
  "NAME": "Дата создания расследования",
  "POSITION": 4,
  "WIDTH": 11,
  "UNVISIBLE": "0",
  "ALIAS": "C_DATE_BEGIN",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Плановая дата окончания",
  "POSITION": 5,
  "WIDTH": 16,
  "UNVISIBLE": "0",
  "ALIAS": "C_DATE_ENDING",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Содержание операции",
  "POSITION": 6,
  "WIDTH": 28,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_CONTENT",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Обобщенный статус",
  "POSITION": 7,
  "WIDTH": 23,
  "UNVISIBLE": "0",
  "ALIAS": "C_COM_STATUS#NAME",
  "REFERENCE_ID": "a2.C_COM_STATUS",
  "TARGET_CLASS_ID": "COM_STATUS_PRD"
}, {
  "NAME": "Исходящие сообщения",
  "POSITION": 8,
  "WIDTH": 18,
  "UNVISIBLE": "0",
  "ALIAS": "C_OUTMSG",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": "SH_OUTMSG"
}, {
  "NAME": "Объекты",
  "POSITION": 9,
  "WIDTH": 10,
  "UNVISIBLE": "0",
  "ALIAS": "C_OBJ",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""

```

```
    }, {
      "NAME": "Номера договоров",
      "POSITION": 10,
      "WIDTH": 18,
      "UNVISIBLE": "0",
      "ALIAS": "C_NUM_DOG_1",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Субъекты",
      "POSITION": 11,
      "WIDTH": 11,
      "UNVISIBLE": "0",
      "ALIAS": "C_SUBJ",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Группа расследования",
      "POSITION": 12,
      "WIDTH": 30,
      "UNVISIBLE": "0",
      "ALIAS": "C_SEARCH_GROUP",
      "REFERENCE_ID": "a1.C_SH_GROUP",
      "TARGET_CLASS_ID": "SH_GROUP"
    }, {
      "NAME": "Ответственные",
      "POSITION": 13,
      "WIDTH": 18,
      "UNVISIBLE": "0",
      "ALIAS": "C_OFFICER",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Отметки",
      "POSITION": 14,
      "WIDTH": 18,
      "UNVISIBLE": "0",
      "ALIAS": "C_MARKS",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Дата закрытия расследования",
      "POSITION": 15,
      "WIDTH": 14,
      "UNVISIBLE": "0",
      "ALIAS": "C_DATE_CLOSE",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Примечания",
      "POSITION": 16,
      "WIDTH": 24,
      "UNVISIBLE": "0",
      "ALIAS": "C_NOTES",
      "REFERENCE_ID": "",
      "TARGET_CLASS_ID": ""
    }, {
      "NAME": "Создано пользователем",
      "POSITION": 17,
      "WIDTH": 23,
```

```

"UNVISIBLE": "0",
"ALIAS": "C_CREATE_USER#NAME",
"REFERENCE_ID": "a2.C_CREATE_USER",
"TARGET_CLASS_ID": "USER"
}, {
  "NAME": "Сумма",
  "POSITION": 18,
  "WIDTH": 14,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_SUM",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Валюта",
  "POSITION": 19,
  "WIDTH": 30,
  "UNVISIBLE": "0",
  "ALIAS": "C_SH_CUR#CUR_SHORT",
  "REFERENCE_ID": "a1.C_SH_CUR",
  "TARGET_CLASS_ID": "FT_MONEY"
}, {
  "NAME": "Ответственный",
  "POSITION": 20,
  "WIDTH": 30,
  "UNVISIBLE": "0",
  "ALIAS": "C_SHOF_USER#NAME",
  "REFERENCE_ID": "b1.C_SHOF_USER",
  "TARGET_CLASS_ID": "USER"
}, {
  "NAME": "Решение",
  "POSITION": 21,
  "WIDTH": 7,
  "UNVISIBLE": "0",
  "ALIAS": "C_DECISION",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""
}, {
  "NAME": "Подразделение",
  "POSITION": 22,
  "WIDTH": 19,
  "UNVISIBLE": "0",
  "ALIAS": "C_DEPART_CODE",
  "REFERENCE_ID": "a2.C_DEPART",
  "TARGET_CLASS_ID": "DEPART"
}, {
  "NAME": "Подразделение ID",
  "POSITION": 23,
  "WIDTH": 13,
  "UNVISIBLE": "2",
  "ALIAS": "C_DEPART",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": "DEPART"
}, {
  "NAME": "Сообщения РСФМ",
  "POSITION": 24,
  "WIDTH": 18,
  "UNVISIBLE": "0",
  "ALIAS": "C_RSFM_MESS",
  "REFERENCE_ID": "",
  "TARGET_CLASS_ID": ""

```

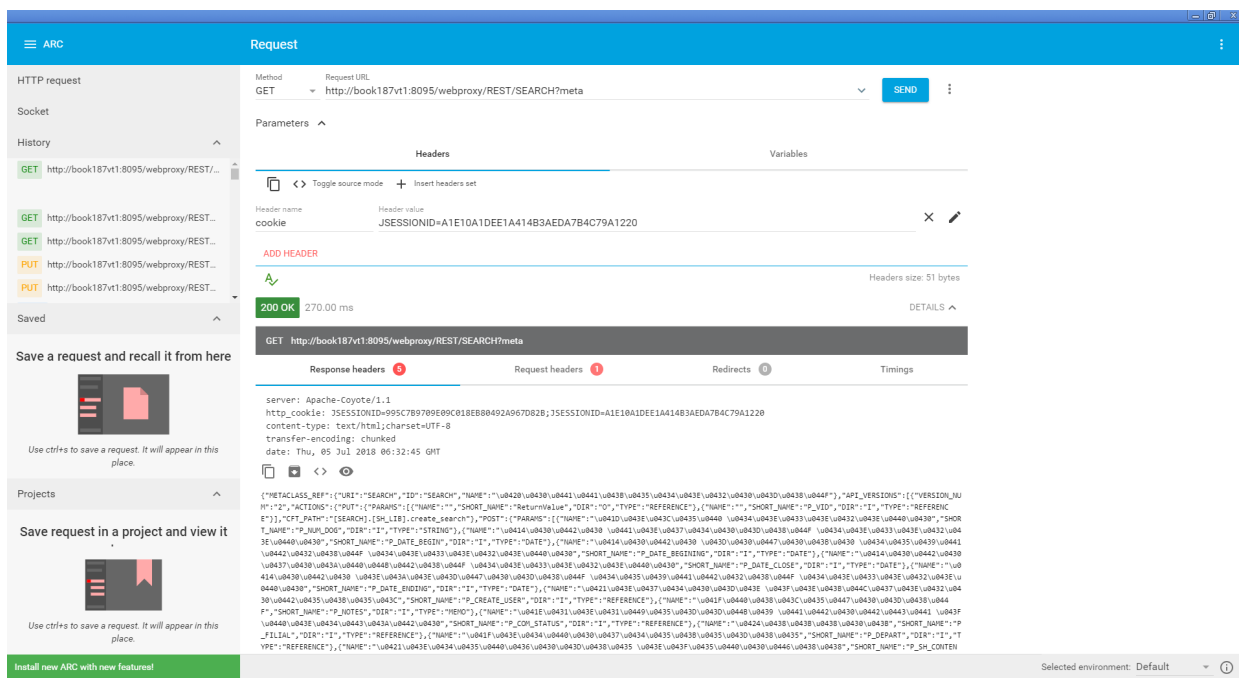
```
    },
    ],
    "VIEW_DETAILS": {
      "NAME": "Полный список",
      "SHORT_NAME": "VW_CRIT_ALL_PLPLUS"
    }
  },
  },
  },
  },
  ],
  "FREE_ACTIONS": {
    "PARAMS": {
      "NAME": "Статус",
      "SHORT_NAME": "P_NEW_STATUS",
      "DIR": "I",
      "TYPE": "REFERENCE"
    },
    "CFT_PATH": "[SEARCH].[CHNG_ST]",
    "VERSION_NUM": "1",
    "URI": "CHNG_ST"
  }
}
```

## 2.5 Проверка работоспособности зарегистрированного сервиса

Проверять работу REST сервиса удобно с помощью **Advanced REST client**. **Advanced REST client** может быть как отдельным приложением, так и расширением Google Chrome. Последнее удобно тем, что нет необходимости устанавливать полноценное приложение.

В Advanced REST client нужно ввести запрос метаданных, например, GET [http://book187vt1:8095/webproxy/REST/REST\\_XMPL\\_DICT?meta](http://book187vt1:8095/webproxy/REST/REST_XMPL_DICT?meta)<sup>[29]</sup>.

Получим примерно следующий результат:



### 2.5.1 Примеры

В качестве примера используется ТБП SEARCH, все действия будут выполняться над ним.

#### Пример 1. Добавление записи в ТБП с помощью 1-й версии API

Для добавления записи нужно выбрать метод **PUT**, в заголовках **content-type** и **accept** указать значение **application/json**,

в Request URL указать URL, например: <http://book187vt1:8095/webproxy/REST/v1/SEARCH><sup>[29]</sup>

В теле запроса json указать значения параметров:

```
{
  "P_NUM_DOG": "1777",
  "P_CREATE_USER": "3257227842",
  "P_FILIAL": "1985880",
  "P_DEPART": "4008483",
  "P_SH_SUM": "1200000"
}
```

```
, "P_SH_CUR": 43353
```

```
, "P_SH_VID": 4677792047}
```

Здесь 3257227842, 1985880, 4008483, 43353, 4677792047 – Id пользователя, филиала, подразделения, валюты и вида расследования.

Результат выполнения запроса - json следующего вида:

```
{
  "NEW_AUTO$result": "4686238012"
}
```

**Пример 2. Чтение данных из записи, добавленной в примере 1. В запросе перечислены колонки представления, по которым нужно получить информацию.**

Нужно выбрать метод **GET**, в Request URL указать URL, например:

```
http://book187vt1:8095/webproxy/REST/v1/SEARCH?FILTER=id eq
4686238012&SELECT=C_DATE_BEGIN,C_NUM_DOG,C_SH_SUM,C_SH_CUR_CUR_SHORT
```

В этом примере предполагается получить информацию только из колонок представления C\_DATE\_BEGIN, C\_NUM\_DOG, C\_SH\_SUM, C\_SH\_CUR#CUR\_SHORT

Если нужно получить данные по всем колонкам представления, запрос выглядит так:

```
http://book187vt1:8095/webproxy/REST/v1/SEARCH?FILTER=id eq 4686238012
```

Результат выполнения запроса:

```
{
  "VIEW": {
    "NAME": "Полный список",
    "SHORT_NAME": "VW_CRIT_ALL_PLPLUS"
  },
  "DATA": {
    "FILTER": "where ID = :0 and rownum < 20",
    "BINDS": [
      "4686238012"
    ],
    "VALS": [
      {
        "C_DATE_BEGIN": {
          "VALUE": "17/10/06"
        },
        "C_NUM_DOG": {
          "VALUE": "6777"
        },
        "C_SH_SUM": {
          "VALUE": 1200000
        },
        "C_SH_CUR_CUR_SHORT": {
          "VALUE": "RUB"
        }
      }
    ]
  }
}
```



```

    ],
  }
}

```

**Внимание!** Если имя колонки в представлении содержит символ "#", то его нужно заменить на нижнее подчеркивание "\_".

В описанном примере `C_SH_CUR#CUR_SHORT` заменено на `C_SH_CUR_CUR_SHORT`

### Пример 3. Редактирование записи, добавленной в примере 1

Нужно выбрать метод **POST**, в заголовках **content-type** и **accept** указать значение **application/json**,

в Request URL указать URL, например:

<http://book187vt1:8095/webproxy/REST/v1/SEARCH?id=294686238012>

В теле запроса json указать значения параметров:

```

{"P_NUM_DOG":"1777"
,"P_CREATE_USER":3257227842
,"P_FILIAL":1985880
,"P_DEPART":4008483
,"P_SH_SUM":1200000
,"P_SH_CUR":43353
,"P_SH_VID":4677792047
,"P_SH_CONTENT":"Содержание операции 6001"}

```

В результате выполнения запроса запись будет обновлена.

**Внимание!** В теле запроса должны быть заданы значения для всех параметров, даже если они не меняются. Иначе значения в записи обнулятся.

### Пример 4. Выполнение операции CHNG\_ST, которая зарегистрирована для ТБП как свободная операция

Нужно выбрать метод **POST**, в заголовках **content-type** и **accept** указать значение **application/json**,

в Request URL указать URL, например:

[http://book187vt1:8095/webproxy/REST/SEARCH/ACTIONS/CHNG\\_ST?id=294686238012](http://book187vt1:8095/webproxy/REST/SEARCH/ACTIONS/CHNG_ST?id=294686238012)

В теле запроса json указать значение параметра операции:

```

{"P_NEW_STATUS":3027208272}

```

В результате будет выполнена операция CHNG\_ST, которая изменит статус записи.

### Пример 5. Удаление записи

Нужно выбрать метод **DELETE**

<http://book187vt1:8095/webproxy/REST/v1/SEARCH?id=4686238012>

В результате выполнения запроса запись будет удалена.

### 3 Работа с REST-сервисами в режиме потребителя

Приложение ["REST. Потребитель веб-служб"](#) предоставляет возможность обращаться к REST-сервисам из модулей Системы, написанных на PL/Plus, затрачивая минимальное время на обработку конкретно интеграционных вещей.

Использование Приложения возможно только совместно с Приложением ["ЦФТ Веб-шлюз"](#) не ниже версии 1.6.2.

### 3.1 Возможности приложения

Приложение ["REST. Потребитель веб-служб"](#) позволяет зарегистрировать REST-сервис и обращаться к нему из прикладного модуля. Для зарегистрированного REST-сервиса есть возможность заполнить массивы заголовков и параметров по умолчанию. Кроме того, при обращении к прикладному ядру указываются:

- ссылка на зарегистрированный сервис
- HTTP-метод, с которым необходимо обратиться
- таблица заголовков
- таблица параметров, которые будут передаваться в URL
- таймаут

Результатом работы функции является HTTP статус. После обработки можно получить http заголовки, а также тело ответа.

## 3.2 Действия, необходимые для установки приложения и начала работы с ним

Для работы с приложением ["REST. Потребитель веб-служб"](#) предварительно нужно выполнить действия:

1. Установить приложение ["ЦФТ Веб - шлюз"](#).
2. В конфигурационном файле webproxy-settings.xml указать значение параметра settings\_name:
 

```
<entry key="settings_name"> WEB_EXTENSIONS:PLP_DADS:DISTR_PLP_DADS</entry>
```

 - если в лицензии на веб-шлюз установлено ограничение на выполняемые методы (опция 2MCA.WEBPROXY.METHODS = WEB\_EXTENSIONS.DISTR\_CALLS), т.е. с помощью веб-шлюза можно взаимодействовать только дистрибутивными точками вызова или
 

```
<entry key="settings_name"> WEB_EXTENSIONS:PLP_DADS:PLP_DADS</entry>
```

 - если ограничение на выполняемые методы не установлено
3. Настроить ЦФТ-Интегратор. Подробнее см. в [Приложении 4](#)<sup>48</sup>.
4. Для запуска веб-шлюза пользователю необходимо предоставить права на:
 

ТБП:

  - WEB\_EXTENSIONS
  - REST\_CONS
  - REST\_PROV

Операции:

  - WEB\_EXTENSIONS.PLP\_DADS
  - WEB\_EXTENSIONS.DISTR\_CALLS
  - REST\_CONS.WS\_MAIN
  - REST\_PROV.WS\_MAIN

Кроме того, при работе с представлениями в случае GET запросов, пользователю необходимо предоставить права на эти представления.

### Внимание!

Для работы с приложением необходимо наличие приложений:

- для "ЦФТ - Банк" - ["Интеграция на основе механизмов сопряжения Oracle"](#)
- для "ЦФТ-Ритейл банк" - "Универсальный шлюз - Базовый функционал"
- для "ЦФТ-Страховая компания" - ["Интеграция на основе механизмов сопряжения Oracle"](#)

Также необходимо убедиться, что установлен pl/json. Pl/json устанавливается автоматически при установке приложения с помощью АРМ "Программа установки обновления".

### 3.3 Особенности текущей версии приложения

Текущая версия приложения имеет следующие особенности:

1. Параметры, записанные в массив или переданные в функцию в качестве параметра, записываются в URL.
2. В качестве тела запроса принимаются JSON и XML. Также в качестве тела запроса можно передать CLOB. В этом случае над телом запроса не будут выполняться дополнительные преобразования.
3. Если в URI зарегистрированного сервиса есть знак «?», считается, что в сервисе проставлены все параметры. Массив параметров сервиса и таблица, переданная в качестве параметра, при этом игнорируются.

## 3.4 Использование приложения

Для использования приложения нужно:

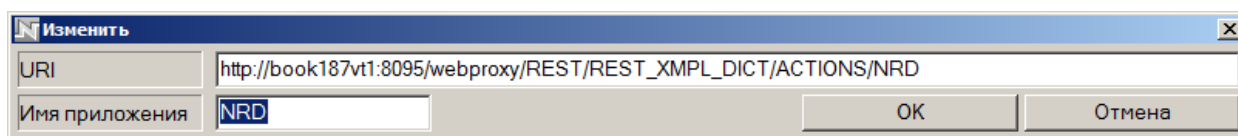
1. [Зарегистрировать сервис](#) <sup>[37]</sup>
2. [Разработать прикладную операцию, вызывающую сервис](#) <sup>[37]</sup>

### 3.4.1 Регистрация сервиса

Для регистрации REST-сервиса необходимо указать базовый URL (без параметров) и имя приложения.

Режим работы: *"Веб-расширения" -> "Веб-службы. Потребитель REST"*

Операции *"Добавить"* и *"Изменить"*



После регистрации REST-сервиса необходимо в массивах HTTP-заголовков и параметров URI заполнить нужные заголовки (такие, как content-type) и, при необходимости, параметры командной строки по умолчанию (например, APIKEY). Значения будут подставляться по умолчанию, также их можно переопределить при вызове прикладного ядра.

### 3.4.2 Реализация прикладной операции, вызывающей сервис

В зависимости от того, в каком формате необходимо отправить запрос, прикладная операция должна обращаться к одной из функций:

- тело запроса в формате XML - `::[WEBEXT_INT].[INT_REST_001].SendXML`
- тело запроса в формате JSON - `::[WEBEXT_INT].[INT_REST_001].SendJSON`
- тело запроса в формате CLOB - `::[WEBEXT_INT].[INT_REST_001].SendReq`

Функции имеют одинаковые параметры:

1. Ссылка на зарегистрированный сервис.
2. Тело запроса.
3. HTTP метод, с которым обращаются к сервису. Например, GET, POST, PUT, DELETE.
4. Таймаут в секундах. По умолчанию равен 120.
5. Таблица параметров. Параметры будут записаны в URL. Например, `tParams('FILTER') := 'id eq 123'`. При этом в URL при вызове будет передано «...NRD?FILTER=id eq 123».
6. Таблица HTTP заголовков. Например, `tHeaders('Content-type') := 'application/json'`.

Если один и тот же параметр определен и в массиве параметров, и в таблице, которая передана в функцию, приоритет имеет параметр в таблице, в URL будет именно его значение. То же самое касается и HTTP заголовков. Результат функции – строка – HTTP статус запроса, например, 200 (успешное выполнение запроса, 404 – не найдено, 500 – общая ошибка сервера, ...). Если ответ от веб-шлюза не получен за указанное в параметре "Таймаут" время, генерируется исключение ::[REST\_CONS].[LIB\_API].REQ\_TIMEOUT.

Если запрос отработал без ошибок, можно получить заголовки ответа с помощью функций `getHeaders` для получения всех заголовков или `getHeader` для получения конкретного заголовка по имени. Кроме того, можно получить тело ответа на запрос. Для этого есть функции `getBody`, `getBodyBLOB`, `getBodyJSON`, `getBodyJS_LIST`, `getBodyXML`. Они возвращают CLOB , BLOB, JSON , JSON\_LIST, XMLType соответственно.

При этом стоит иметь в виду, что, даже если статус запроса – один из неуспешных, например, 404 или 500, в заголовках или теле ответа вполне может содержаться дополнительная информация, которая может помочь в идентификации причины ошибки.

Пример работы с прикладным ядром можно посмотреть в операциях `REST_CONS.SEND_JSON_TST`, `REST_CONS.SEND_XML_TST`.

Кроме того, можно воспользоваться этими операциями для тестирования сервиса вручную. Для этого на зарегистрированном сервисе нужно выполнить операцию «Отправка запроса» или «Отправка запроса XML».

**Режим работы: "Веб-расширения" -> "Веб-службы. Потребитель REST"**

**Операция "Отправка запроса"**



Отправка запроса

Тело запроса

```
{
  "METACLASS_REF": {
    "URI": "/REST/MAIN_DOCUM/",
    "ID": "MAIN_DOCUM",
    "NAME": "Платёжные документы"
  },
  "API_VERSIONS": [
    {
      "VERSION_NUM": "1.1"
    }, {
      "VERSION_NUM": "1.2",
      "ACTIONS": {
        "POST": {
          "PARAMS": [
            {
              "NAME": "P_DATE_VAL",
              "DIR": "IN",
              "TYPE": "DATE"
            }, {
              "NAME": "P_NUM",
              "DIR": "IN",
              "TYPE": "NUMBER"
            }
          ]
        }, {
          "PUT": {
```

Метод **POST** Таймаут(сек) 50 OK Отмена

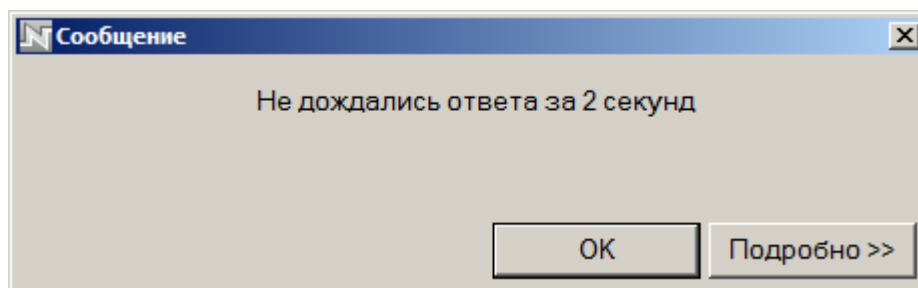
вариант результата

☒ json ☐ json\_list

Результат выводится в буфер сессии:

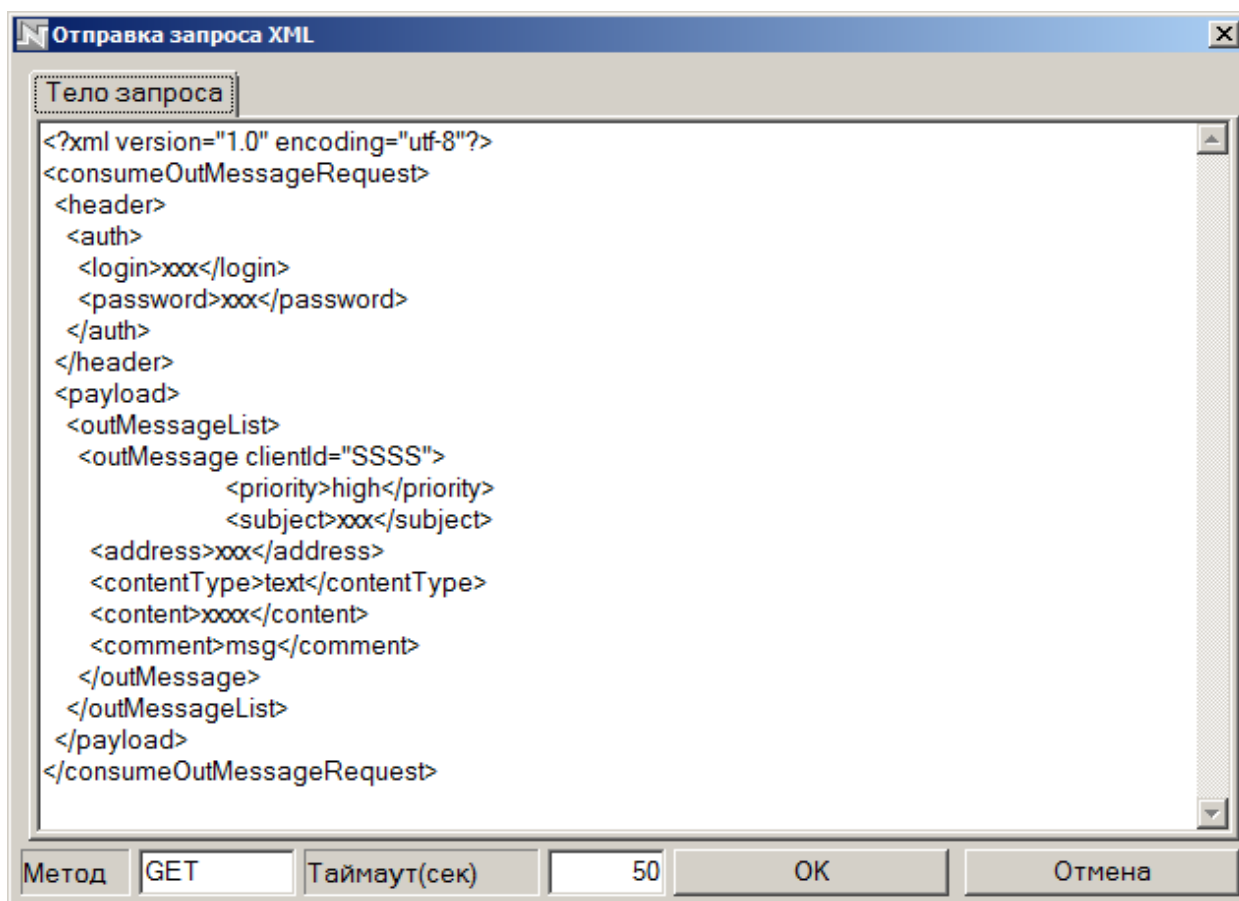


Если происходит таймаут, соответствующая ошибка перехватывается, выдаётся сообщение об ошибке:



Режим работы: "Веб-расширения" -> "Веб-службы. Потребитель REST"

### Операция "Отправка запроса XML"



Результат работы операции **"Отправка запроса XML"** аналогичен результату работы операции **"Отправка запроса"**.

### 3.4.3 Использование приложения в асинхронном режиме

Приложение ["REST. Потребитель веб-служб"](#) позволяет работать в асинхронном режиме.

Для этого необходимо:

1. При отправке запроса указать пустое значение в параметре для таймаута.
2. В таблицу заголовков добавить заголовок с кодом "SCRIPT\_NAME". В качестве значения указать URI к веб-сервису в ТБП "Веб-службы. Поставщик REST".

При этом прикладная операция при отправке запроса не ждёт ответа, получить HTTP статус или тело ответа будет нельзя.

При получении ответа от удалённого сервиса будет вызван соответствующий поставщик REST, в котором можно обработать ответ на запрос.

## 4 Справочник "Веб-сервисы. Журнал выполнения"

Режим работы: "Справочники" -> "Веб-сервисы. Журнал выполнения"

Справочник содержит информацию об обращениях к REST-сервисам как в режиме поставщика, так и в режиме потребителя:

- дату и время начала выполнения запроса
- дату и время окончания выполнения запроса
- ссылку на Rest-сервис
- статус запроса
- текст запроса
- текст ответа
- идентификатор запроса при работе в режиме потребителя

К таблице справочника должен быть применен профиль секционирования WEBEXT\_LOGS (Веб-расширения. Журналы).

## 5 Приложение 1

### Поддерживаемые возможности oData при обработке HTTP метода GET

При передаче запроса, в строке соединения можно задать параметры, которые будут транслироваться в запрос к представлению в Системе.

Для передачи условий запроса необходимо указывать их в параметре запроса FILTER. Например, MAIN\_DOCUM?FILTER=C\_NUM\_DT LIKE 40817%

Доступны:

1. AND транслируется в AND запроса
2. OR транслируется в OR запроса
3. EQ транслируется в "="
4. NE транслируется в "!="
5. GT транслируется в ">"
6. GE транслируется в ">="
7. LT транслируется в "<"
8. LE транслируется в "<="
9. LIKE транслируется в LIKE запроса

Реализована поддержка параметров TOP – количество строк, которые необходимо возвращать, и SKIP – количество строк, которые нужно пропустить. Допустим, необходимо в GET запросе прочитать строки с 100 по 150. Тогда TOP=50, SKIP=100

Кроме того, реализована поддержка параметров запроса INLINECOUNT и SELECT. Первый трансформируется в условие rownum < (значение параметра).

Второй – список колонок, которые нужно отображать.

Если в представлении есть ссылочная колонка, вывод имеет вид

```
"CUR2": {
  "REF": 43379,
  "VALUE": "USD"
}
```

В таком случае для отображения поля необходимо в параметре SELECT указывать следующее: CUR2.REF. Для отображения значения поля (поле value) нужно просто указывать название колонки.

В общем случае URL будет иметь следующий вид:

[http://book187vt1:8095/webproxy/REST/v1/REST\\_XMPL\\_DICT?FILTER=id<sup>43</sup>eq 4686412966 OR id EQ 4686412985&INLINECOUNT=10&SELECT=CUR2.REF,CUR2,ID&TOP=10&SKIP=50](http://book187vt1:8095/webproxy/REST/v1/REST_XMPL_DICT?FILTER=id<sup>43</sup>eq 4686412966 OR id EQ 4686412985&INLINECOUNT=10&SELECT=CUR2.REF,CUR2,ID&TOP=10&SKIP=50)

( book187vt1:8095/webпроху приведено в качестве примера )

Результат выполнения запроса:

```
{
  "VIEW": {
    "NAME": "Полный список ",
    "SHORT_NAME": "VW_CRIT_REST_XMPL_DICT"
  },
  "DATA": {
    "FILTER": "where ID = :0 OR ID = :4 and rownum < 10",
    "BINDS": [
      "4686412966", "4686412985"
    ],
    "VALS": [
      {
        "CUR2": {
          "REF": 43379,
          "VALUE": "USD"
        },
        "ID": 4686412985
      }, {
        "CUR2": {
          "REF": 43379,
          "VALUE": "USD"
        },
        "ID": 4686412966
      }
    ]
  }
}
```

## 6 Приложение 2

### Основные HTTP коды результата выполнения запроса

Для всех кодов, кроме 200, более подробный текст ошибки находится в теге ErrorMessage

- 200 – OK – успешный запрос.
- 404 – NOT\_FOUND – запрошенный ресурс не может быть найден. Может выводиться в случаях: не найден запрашиваемый ТБП, не определен запрашиваемый метод для переданной версии.
- 405 – METHOD\_NOT\_ALLOWED – HTTP-метод запроса не поддерживается для запрашиваемого ресурса, отображается для неподдерживаемых HTTP-методов, например, метод PATCH. Также возникает при попытке вызвать свободное действие через HTTP метод, отличный от POST.
- 409 – CONFLICT – отображается, если ТБП определен неоднозначно (несколько записей с одинаковым URI).
- 415 – UNSUPPORTED\_MEDIA\_TYPE – объект запроса имеет тип, который сервер не поддерживает. Подробнее об используемых в данной версии значениях http заголовков ACCEPT, CONTENT\_TYPE описано в пункте [Требования к прикладным операциям, функциям, процедурам](#)<sup>13</sup>.
- 500 – Internal Server Error – прочие ошибки. Конкретный текст ошибки необходимо смотреть в теге ErrorMessage.

## 7 Приложение 3

### Выполнение настроек для работы Приложения ["REST. Поставщик веб-служб"](#)

В справочнике "Настройки." (FP\_TUNE) должны быть настроены следующие записи:

1. Настройка с кодом REST\_JOBS\_CNT - управляет количеством движков веб-шлюза, обрабатывающих исходящие запросы. Если настройка отсутствует в справочнике или имеет пустое значение, по умолчанию количество движков равно 10.

Изменить

Изменить | Описание

Группа: Веб-расширения

Наименование: Веб-расширения. Потребитель REST. Количество движков

Код: REST\_JOBS\_CNT

Тип значения:

- ☐ Строка
- ☒ Число
- ☐ Дата
- ☐ Логика
- ☐ Ссылка
- ☐ Мемо
- ☐ CLOB
- ☐ Функция

Значение: 10

☐ Использовать критерии выбора значения

☒ Кэшировать получение значений

Кэширование рекомендуется использовать для редко изменяемых настроек

Критерии

OK Отмена

2. Настройка с кодом REST\_JOBS\_INT - управляет интервалом в секундах, через который движок веб-шлюза обращается к БД для получения запросов. Если настройка отсутствует в справочнике или имеет пустое значение, по умолчанию интервал обращения к БД равен 5 секунд.



Изменить

Изменить | Описание

Группа: Веб-расширения

Наименование: Веб-расширения. Потребитель REST. Интервал обращения в БД

Код: REST\_JOBS\_INT

Тип значения:

- ☐ Строка
- ☒ Число
- ☐ Дата
- ☐ Логика
- ☐ Ссылка
- ☐ Мемо
- ☐ CLOB
- ☐ Функция

Значение: 5

☐ Использовать критерии выбора значения

☒ Кэшировать получение значений

Кэширование рекомендуется использовать для редко изменяемых настроек

Критерии

OK Отмена

## 8 Приложение 4

### Настройка ЦФТ-Интегратора для работы Приложения ["REST. Потребитель веб-служб"](#)

Для функционирования приложения используется система-абонент ЦФТ-Интегратор с кодом «REST\_CONS».

Необходимо создать соответствующую систему-абонент, настроить её, как показано на скриншоте ниже.

Режим работы: *"Справочники" -> Группа справочников "Интегратор" -> "Интегратор. Системы-абоненты"*

Операции *"Добавить"* и *"Изменить"*

**Изменить**

Система-абонент

Код:  ☐ Своя система

Наименование:

Имя линка:  Владелец:

Кодировка XML:  NAMESPACE:

Тип очереди:  Код абонента получателя:

☐ Применять XSLT для входящих  
☐ Применять XSLT для исходящих  
☐ Не создавать элементы с пустыми значениями  
☐ Использовать отзывы  
☐ Взаимодействие с IBM WebSphere MQ  
☐ Использовать альтернативный график продвижения

☒ Не использовать файловый обмен  
☐ Использовать приоритеты

Пользователь для работы:

Интерфейс вызова внешней системы:

Количество обработчиков очереди:

Используемый XML-парсер:  
☐ Xerces ☒ Oracle XML

Использовать дополнительные обработчики: ☐

После непрерывной обработки в течение:  с.

Запускать по:

OK Отмена

После этого с помощью операции *"Управление окружением"* создать и запустить входящую и исходящую очереди.

**Внимание!** Обработчики при этом запускать нельзя. Иначе обработчики будут вычитывать

задания из входящей очереди, и ответы на запросы на запросы будут теряться, в связи с чем в прикладных операциях будут возникать таймауты.

