

Руководство  
по настройке источника  
для приложения  
«Мобильный монитор  
показателей»

2014

# ОГЛАВЛЕНИЕ

<i>Введение</i> .....	3
<i>Общее описание</i> .....	5
<i>Основные определения и сокращения</i> .....	7
<i>Общая последовательность действий по настройке источника</i> .....	9
Подготовительный этап.....	9
Объединение конфигураций пользователя и макета источника.....	9
Общая структура макета конфигурации и структура вызовов методов web-сервиса.....	9
Пример добавления нового показателя.....	9
Публикация web-сервиса и проверка работы .....	10
<i>Объединение конфигураций</i> .....	11
<i>Описание структуры конфигурации</i> .....	17
<i>Описание структуры протокола взаимодействия и последовательности методов вызова web- сервиса</i> .....	21
<i>Добавление нового показателя</i> .....	21
Постановка задачи.....	25
Реализация поставленной задачи.....	25
<i>Публикация web-сервиса</i> .....	36
<i>Ограничения прямого подключения к базе 1С:Предприятие</i> .....	38

## **ВВЕДЕНИЕ**

Это Руководство представляет собой описание макета конфигурации источника данных для приложения «Мобильный монитор показателей» и инструкции по настройке нового источника данных.

Руководство рассчитано на то, что Вы знакомы с конфигурированием и программированием для системы 1С:Предприятие 8.

Необходимы следующие знания и умения:

- понимание механизмов работы web-сервисов;
- умение опубликовать web-сервис из 1С:Предприятия;
- опыт сравнения и объединения конфигураций;
- умение формировать запросы к данным.

В данном руководстве в качестве примера источника используется типовая конфигурация «Управление торговлей, 11».



## ОБЩЕЕ ОПИСАНИЕ

«Мобильный монитор показателей» – программный комплекс для оперативного слежения и отображения основных показателей деятельности предприятия на мобильных устройствах.

Основные возможности:

- создание любого количества показателей с возможностью их группировки и совместного анализа. Настройка периодов агрегации и отображения данных;
- формирование информационных рассылок для сотрудников компании – новости, события календаря, напоминания, автоматические оповещения об отклонениях в показателях и т.п.;
- возможность проведения согласовательных процедур – заявки, поручения и другие бизнес-процессы, которые требуют принятия решения через выбор решения из набора вариантов.

Продукт представляет собой комплекс, состоящий из:

- мобильных приложений для устройств на базе Android и iOS;
- web-сервис для обмена данными;
- универсальную конфигурацию (макет) на 1С;
- web-сервер синхронизации;
- протокол обмена.

Возможны две модели реализации системы:

- универсальная конфигурация встраивается в любую типовую, отраслевую конфигурацию на базе 1С:Предприятие 8. Мобильное приложение обменивается данными через web-сервис, встроенный в конфигурацию (Рисунок 1);

- взаимодействие происходит через web-сервер, где пользователи могут регистрировать доступ и производить обмен данными по стандартному протоколу между любыми источниками и мобильными приложениями данного программного комплекса (Рисунок 2).

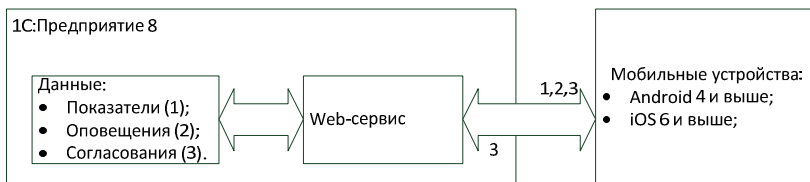


Рисунок 1. Прямое подключение «Мобильного монитора показателей» к базе данных 1С:Предприятие 8.

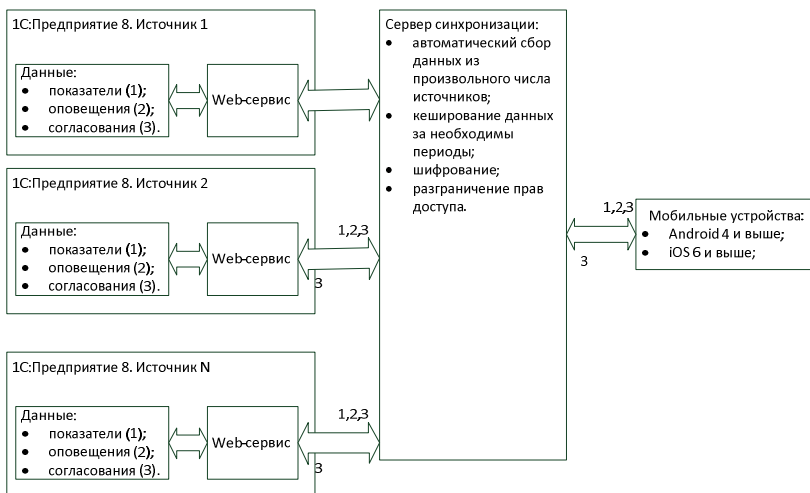


Рисунок 2. Подключение «Мобильного монитора показателей» через сервер синхронизации.

---

При работе с сервером синхронизации возможно использование двух вариантов подключения:

1. Использование прямого протокола – инициатором обмена является сервер синхронизации и он вызывает методы веб-сервиса опубликованного на стороне источника (базы данных 1С:Предприятие).
2. Использование обратного протокола – инициатором является источник (база данных 1С:Предприятия или любая другая система с поддержкой SOAP).

Выбор варианта подключения зависит от поставленных задач.

## ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

**Мобильный монитор показателей (ММП)** – программный комплекс для оперативного слежения и отображения основных показателей деятельности предприятия на мобильных устройствах.

**Источник данных (Источник)** – информационная база 1С:Предприятия, хранящая данные для отправки на мобильные устройства, в которой опубликован специальным образом описанный web-сервис для обмена данными. На данный момент поддерживается возможность публиковать несколько источников внутри одной информационной базы. Данная возможность будет отключена после реализации механизмов передачи аналитической информации по показателям.

**Потребитель данных (Потребитель)** – потребителем данных может выступать либо сервер синхронизации, либо мобильное устройство. Зависит от способа организации взаимодействия.

**Показатель** – любые данные, необходимые для анализа (например, дебиторская задолженность). Может быть развернут по времени или детализирован аналитиками.

---

**Аналитика** – детализированная информация по конкретному значению показателя (например, дебиторская задолженность может иметь аналитику – контрагент, организация).

**Шкала** – объект данных, позволяющий указать свойства для измерения показателя (единица измерения, минимальное и максимальное значения), а также указать, что разные показатели сравнимы между собой (например, если указать для показателей «Выручка от продаж» и «Себестоимость продаж» одинаковую шкалу, то тогда эти показатели на мобильном устройстве можно будет отображать в единых координатных осях). Шкала может быть числовая (значения шкалы имеют числовые значения) или перечисляемая (значения являются ограниченным набором строковых значений. Например, набор статусов документов). На данный момент реализована поддержка только числовых шкал.

**Единица измерения** – показывает, в какой единице измерения исчисляется показатель.

**Канал** – предназначен для разграничения прав доступа на сервере. Все показатели могут быть распределены по различным каналам данных и эти каналы уже назначаются пользователям. Пользователь видит только те показатели, которые включены в доступные ему каналы.

**Пользователь** – определяется логином и паролем. Служит для авторизации и определения прав доступа на сервере синхронизации.

**Сервер синхронизации** – сервер, предназначенный для сбора и хранения данных из всех источников с последующей передачей этих данных на мобильные устройства.

**Представление показателей** – элемент группировки для отображения на мобильном устройстве любого количества показателей. По умолчанию представления формируются по шкалам, т.е. все показатели с одинаковой шкалой будут включены в одно представление.



---

## ОБЩАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ ПО НАСТРОЙКЕ ИСТОЧНИКА

### **Подготовительный этап**

На этом этапе необходимо установить (если не установлено):

- «Модуль расширения web-сервисов» для 1С:Предприятие 8.
- Web-сервер.

Установка и настройка данных модулей подробно описана в комплекте документации, поставляемой с вашей редакцией 1С:Предприятия 8.

При использовании обратного протокола данный этап можно опустить.

### **Объединение конфигураций пользователя и макета источника**

На данном этапе необходимо произвести сравнение объединение поставляемого макета источника и рабочей конфигурацией базы данных 1С:Предприятие.

### **Общая структура макета конфигурации и структура вызовов методов web-сервиса**

На данном этапе необходимо изучить описание объектов метаданных макета источника, назначение и последовательность вызова методов web-сервиса.

При использовании обратного протокола методы web-сервиса использовать не нужно.

### **Пример добавления нового показателя**

Изучить представленный в документации и макете источника пример. Разработать и опубликовать собственный показатель.

### Публикация web-сервиса и проверка работы

Опубликовать web-сервис из базы данных 1С:Предприятия и проверить его работоспособность.

При использовании обратного протокола публикация web-сервиса не требуется. Проверка работоспособности осуществляется вызовом методов обратного протокола.

## ОБЪЕДИНЕНИЕ КОНФИГУРАЦИЙ

Последовательность действий по объединению конфигураций:

1. Запустить базу данных в режиме «Конфигуратор» (в примере описывается процесс объединения с демо-базой «Управление торговлей, редакция 11»).
2. Сделать архив копии базы данных: выбрать в главном меню пункт «Администрирование» | «Выгрузить информационную базу».
3. Включить возможность изменения конфигурации: выбрать в главном меню пункт «Конфигурация» | «Поддержка» | «Настройка поддержки». В появившемся окне нажать кнопку «Включить возможность изменения» (Рисунок 3).

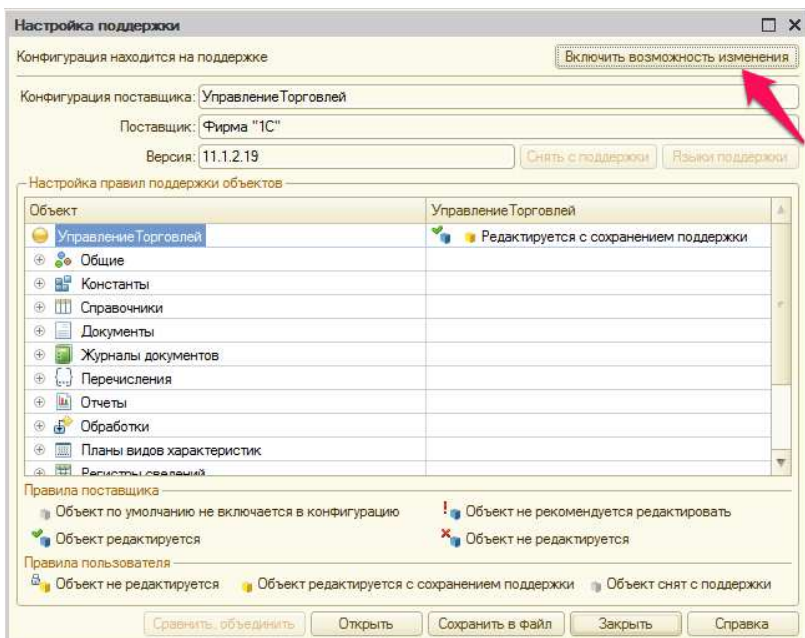


Рисунок 3. Настройка поддержки конфигурации.

4. Подтвердить свои действия нажатием кнопки «Да» в окне вопроса (Рисунок 4).

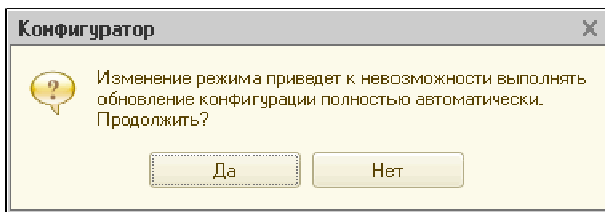


Рисунок 4 – Изменение режима поддержки

5. В открывшемся окне настройки правил поддержки установить флажки в соответствии с представленными на Рисунке 5:

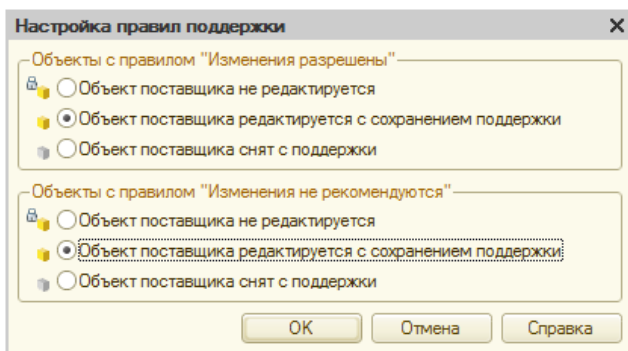


Рисунок 5 - Правила настройки поддержки

6. Для объединения с конфигурацией макета источника необходимо в главном меню выбрать пункт «Конфигурация» | «Сравнить, объединить с конфигурацией из файла» (Рисунок 6).

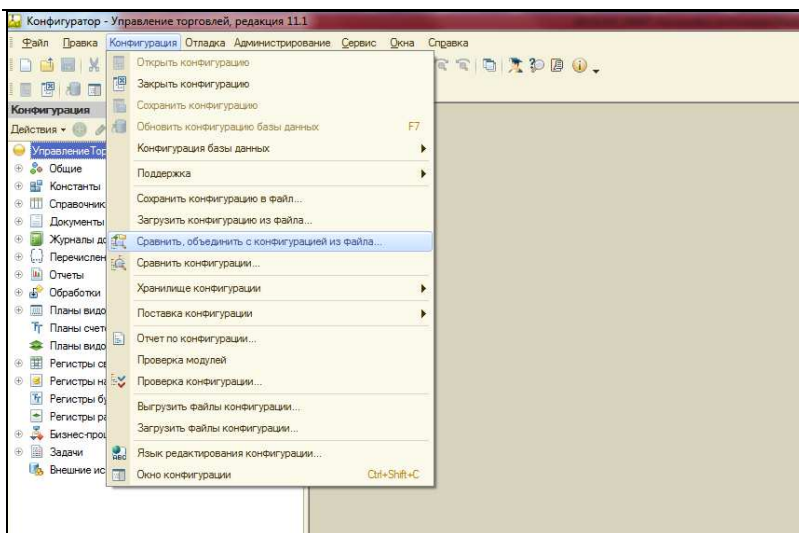


Рисунок 6. Выбор файла макета источника.

7. В открывшемся окне надо выбрать файл конфигурации макета источника, нажать «Открыть».
8. После выполнения этих действий будет выведен список объектов объединяемых конфигураций (Рисунок 7). В списке необходимо снять галочку с пункта «Свойства» и нажать кнопку «Выполнить».

**Важно!**

Данная операция может занять несколько минут.

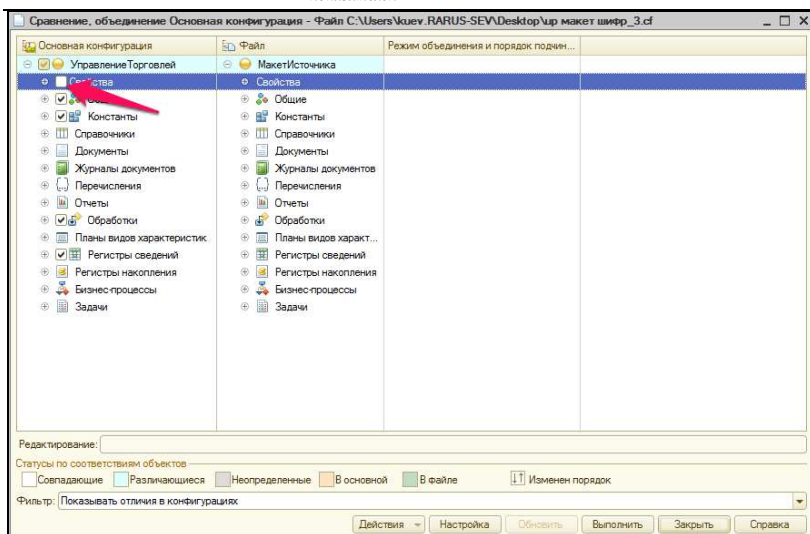


Рисунок 7. Сравнение и объединение конфигураций.

9. По окончании операции в окне конфигурации появятся новые объекты с префиксом «mm» (можно отобразить по подсистеме «mmMobileMonitor» - пункт меню «Действия» | «По подсистемам») (Рисунок 8)

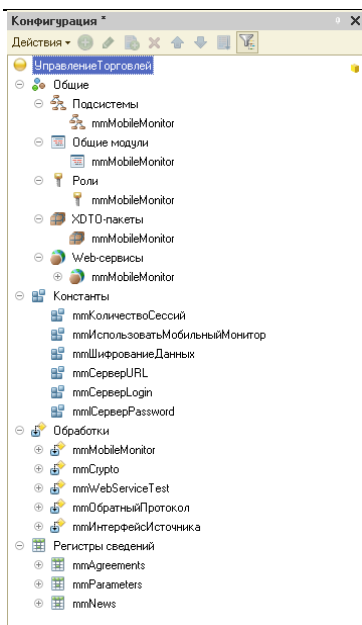


Рисунок 8. Объекты подсистемы «mmMobileMonitor».

10. В зависимости от используемого способа получения данных пользователю необходимо назначать права на получение данных из объектов конфигурации.

При получении данных из конфигурации при обращении к web-сервису у пользователя должны быть права на все объекты подсистемы «mmMobileMonitor» и права на чтение данных, которые участвуют в запросах на получение необходимых показателей, оповещений. На объекты участвующие в формировании согласований, необходимы права на чтение и запись.

11. Создать пользователя с именем и паролем «mobileUser» и назначить ему роль «Mobile monitor». Для этого выбрать пункт меню «Администрирование» | «Пользователи» и добавить нового пользователя в список (Рисунок 9).

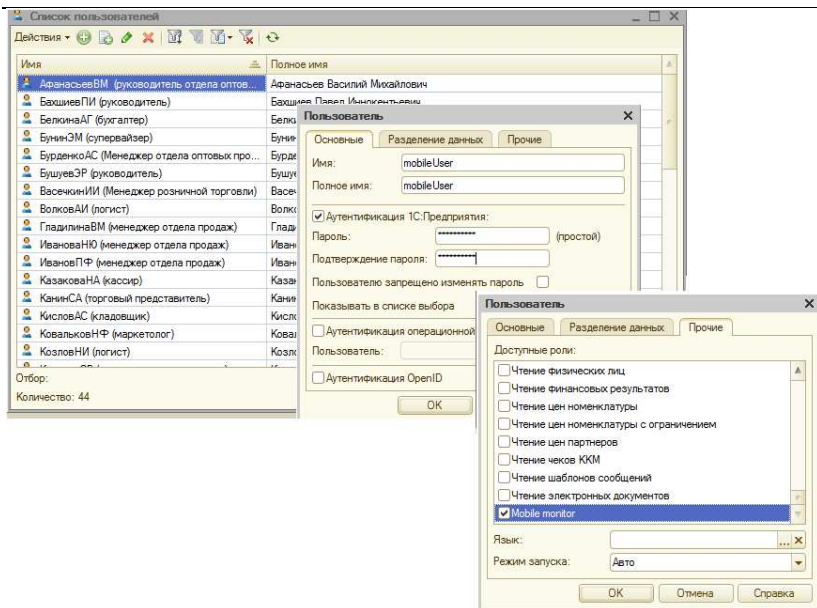


Рисунок 9. Добавление нового пользователя.



## ОПИСАНИЕ СТРУКТУРЫ КОНФИГУРАЦИИ

Для обеспечения легкого встраивания и дальнейшей поддержки подсистемы «mmMobileMonitor» в составе сторонних конфигураций приняты следующие решения:

1. Все объекты необходимые для функционирования «Мобильного монитора показателей» прифигируются и их название начинается с «mm». Также все объекты включены в подсистему «mmMobileMonitor».
2. Обязательно должен существовать пользователь для авторизации в web-сервисе. При прямом подключении имя пользователя и пароль должны быть mobileUser, в ином случае (с использованием сервера мобильного монитора показателей) могут быть произвольными.
3. Этому пользователю должна быть установлена роль «Mobile monitor» и даны права на чтение данных необходимых для формирования запросов на получение показателей, оповещений и согласований. Для объектов, работающих с механизмом согласований необходимо установить права на запись. Для этого необходимо пользователю включить соответствующие роли.
4. Все методы для работы прямого протокола «Мобильного монитора показателей» определены в модуле web-сервиса «mmMobileMonitor», но их реализация переопределена в модуле менеджера обработки «mmMobileMonitor». Эти методы не редактируются, любые изменения необходимые для реализации собственных показателей, оповещений и согласований нужно реализовывать в модуле менеджера обработки «mmИнтерфейсИсточника».
5. Все методы для работы обратного протокола «Мобильного монитора показателей» определены в модуле менеджера обработки «mmОбратныйПротокол». Этот модуль не

---

редактируется, любые изменения необходимые для реализации собственных показателей, оповещений и согласований нужно реализовывать в модуле менеджера обработки «mmИнтерфейсИсточника».

Обратный протокол используется тогда, когда инициатором подключения является информационная база. С его помощью можно отправить на сервер мобильного монитора значения и настройки показателей, список оповещений и согласований, а также получить ответы на согласования.

Используя обратный протокол нельзя создать новые источники, но можно обновить существующие. Для этого надо отредактировать изменяемые поля, но идентификаторы оставить прежними.

6. Для работы механизмов шифрования реализована обработка «mmСrypto». В модуле менеджера реализованы методы работы с подключением компоненты шифрования и вспомогательные методы для вызова соответствующих обработчиков компоненты. Сама компонента хранится в макетах данной обработки.
7. Для ускорения работы методов по выборке данных, через web-сервис реализованы кэширующие регистры сведений «mmParameters», «mmNews» и «mmAgreements». Их структура позволяет простым запросом к данным выгрузить все необходимые показатели, оповещения и согласования. Запись необходимых данных в этот регистр реализуется любым удобным способом (например, при проведении конкретных документов, или организацией подписки на конкретное событие в системе, или регламентным заданием). Реализуется в конкретном случае индивидуально, согласно конкретной задаче. Если в кэшировании нет необходимости, то данные можно выбирать напрямую из соответствующих объектов 1С:Предприятия, для этого необходимо

---

реализовать пользовательский обработчик получения данных в модуле «mmИнтерфейсИсточника».

8. Для настроек работы «Мобильного монитора показателей» используются константы:

- «mmИспользоватьМобильныйМонитор» – определяет необходимость использования «Мобильного монитора показателей». Если установлена **ЛОЖЬ**, то система будет отключена, и web-сервис не будет отвечать на внешние запросы. По умолчанию отключено.

**Важно!**

Если не включить эту константу, то при подключении макета к вашей конфигурации web-сервис не будет публиковать данные. Для включения константы в демо-примере на базе «Управление торговлей, редакция 11» необходимо зайти в основное меню «Все операции» | «Константы» и установить галочку для константы «Использовать мобильный монитор показателей».

- «mmКоличествоСессий» – ограничивает количество одновременных подключений к базе данных источника. Это ограничение позволяет предотвратить DDoS атаку на источник данных. Если значение рано нулю, то проверка на количество подключений не выполняется.
- «mmШифрованиеДанных» – определяет, будет ли использоваться шифрование данных или нет. По умолчанию отключено. Более подробную информацию можно найти в разделе «Шифрование данных».
- mmСерверURL – URL адрес сервера мобильного монитора. Используется в обратном протоколе.
- mmСерверLogin – имя пользователя мобильного монитора показателей. Используется в обратном протоколе.

- 
- `mmlСерверPassword` – пароль пользователя мобильного монитора показателей. Используется в обратном протоколе.

## ПОСЛЕДОВАТЕЛЬНОСТИ МЕТОДОВ ВЫЗОВА WEB- СЕРВИСА (ПРЯМОЙ ПРОТОКОЛ)

Для реализации возможности работы как напрямую с базой данных 1С:Предприятие, так и через сервер синхронизации был реализован симметричный протокол обмена данными, т.е. как при работе источников данных напрямую с мобильными устройствами, так и при обмене с сервером синхронизации используется одна и та же последовательность команд. В протоколе реализован механизм последовательной выгрузки каждого показателя, а не выгрузка всего пакета данных в целом. Это позволяет организовать передачу данных на слабых каналах связи между источником и потребителем. При разрыве связи обмен происходит с места разрыва, а не передачей всего пакета заново.

Общая структура протокола обмена данными представлена на рисунке 10.

Назначение и описание методов протокола обмена данными приведено в таблице 1.

Во всех методах работы с данными передается структура loginStruct – она предназначена для передачи параметров авторизации (логин и пароль пользователя). При авторизации на источнике данных используется предопределенный пользователь mobileUser, при авторизации на сервере синхронизации (запросы с мобильных устройств) передается конкретный пользователь сервиса (для разграничения прав доступа на данные пользователя).

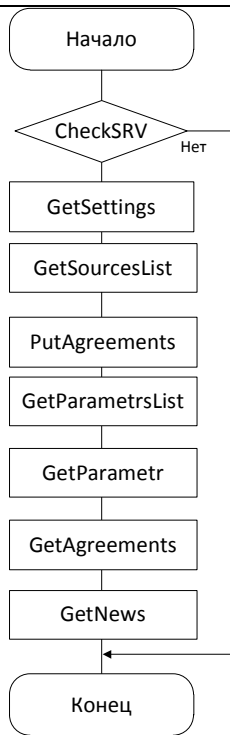


Рисунок 10. Структура протокола обмена данными.

Таблица 1 – Описание методов web-сервиса

Метод	Назначение	Описание
CheckSrv (checkSrvData)	Определяет доступность источника данных.	Если при вызове метода ответ не приходит, то все остальные методы не будут выполняться. Для корректной работы сервиса переопределение (изменение) метода не рекомендуется.
GetSettings (loginStruct, getSettingsData)	Передает настройки показателей по умолчанию	Метод возвращает начальные настройки для мобильного приложения и служебную информацию. В параметре getSettingsData передается служебная информация.
GetSourcesList (sources)	Возвращает структуру источников из информационной базы.	На данный момент реализована поддержка множественных источников в одной базе данных 1С:Предприятие. В дальнейшем данный подход не будет поддерживаться. Рекомендуется для последующей совместимости использовать один источник для одной информационной базы 1С:Предприятие.
PutAgreements (putAgreementsData, Result)	Возвращает на источник реакцию пользователей на согласования.	В параметре putAgreementsData передается список согласований и вариант ответа пользователя с сервера (мобильного приложения). Метод вызывается до формирования списка новых согласований и отправки его на сервер (мобильное приложение).
GetParametersList (loginS	Формирование	Метод предназначен для

<code>truct, getParamsData)</code>	списка доступных показателей.	того, чтобы сформировать перечень всех доступных показателей, согласований и оповещений. Позволяет потребителю данных определить структуру всех данных и сопоставить одинаковые данные из разных источников.
<code>GetParameter(loginStruct, getParamData)</code>	Выгрузка конкретного показателя	Потребитель данных в цикле по списку параметров полученным методом <code>GetParametersList()</code> запрашивает данные конкретного показателя. В параметре <code>getParamData</code> передается информация о показателе, источнике и периоде выборки данных.
<code>GetAgreements(loginStruct, getAgreesData)</code>		Аналогично методу <code>GetParameter()</code> происходит выборка согласований за передаваемый в <code>getAgreesData</code> период времени.
<code>GetNews(loginStruct, getNewsData)</code>		Аналогично методу <code>GetParameter()</code> происходит выборка согласований за передаваемый в <code>getNewsData</code> период времени.

Подробное описание методов и параметров протокола описано в разделе «Описание протокола обмена данными».



## ДОБАВЛЕНИЕ НОВОГО ПОКАЗАТЕЛЯ

### Постановка задачи.

Необходимо добавить новый показатель «Продажи за день» для конфигурации «Управление торговлей, редакция 11» (демо-база из комплекта поставки). В качестве источника использовать организацию «Торговый дом «Комплексный».

### Реализация поставленной задачи (Прямой протокол)

Для возможности публикации данных необходимо в пользовательском режиме установить истинное значение константы «ММ: Использовать мобильный монитор показателей», иначе web-сервис не будет отвечать на внешние запросы.

Для добавления показателя необходимо выполнить следующие действия:

1. Открыть модуль менеджера обработки «mmИнтерфейсИсточника»;
2. Для определения списка источников необходимо модифицировать функцию ПолучитьСписокИсточников(), введя параметры своего источника данных в таблицу значений:

```
Функция ПолучитьСписокИсточников() Экспорт
Источники = Новый ТаблицаЗначений;
Источники.Колонки.Добавить ("Наименование");
Источники.Колонки.Добавить ("ИдИсточника");

Источник = Источники.Добавить();
Источник.Наименование = "Торговый дом "Комплексный" ";
Источник.ИдИсточника = "TD";

Возврат Источники;
КонецФункции
```

### 3. Далее необходимо сформировать правильную таблицу показателей с колонками ИдПоказателя, ИдИсточника, Обработчик, возвращаемую методом ПолучитьСписокПоказателейИсточников():

```

Функция ПолучитьСписокПоказателейИсточников() Экспорт
Соответствия = Новый ТаблицаЗначений;
Соответствия.Колонки.Добавить("ИдПоказателя");
Соответствия.Колонки.Добавить("ИдИсточника");
Соответствия.Колонки.Добавить("Обработчик");

ПоказательИсточника = Соответствия.Добавить();
ПоказательИсточника.ИдПоказателя = "salesParameterID";
ПоказательИсточника.ИдИсточника = "TD";
ПоказательИсточника.Обработчик =
"ПолучитьЗначенияПоказателяИзКонфигурации";
КонецФункции

```

### 4. Если у показателя указано название обработчика, то в этом же модуле надо определить функцию с таким же названием, входными параметрами которой является структура. Данная функция должна возвращать таблицу значений с данными показателя и быть экспортной.

Функция ПолучитьЗначенияПоказателяИзКонфигурации(ВходящиеДанные) Экспорт

```

ЗначенияПоказателя = Новый ТаблицаЗначений;
ЗначенияПоказателя.Колонки.Добавить("Дата");
ЗначенияПоказателя.Колонки.Добавить("Значение");
ЗначенияПоказателя.Колонки.Добавить("Аналитика1");
//ЗначенияПоказателя.Колонки.Добавить("Аналитика2");
//ЗначенияПоказателя.Колонки.Добавить("Аналитика3");

НовоеЗначение = ЗначенияПоказателя.Добавить();
НовоеЗначение.Дата = Дата(2014, 01, 01);
НовоеЗначение.Значение = 200;
НовоеЗначение.Аналитика1 = "extValue-Розница";

НовоеЗначение = ЗначенияПоказателя.Добавить();
НовоеЗначение.Дата = Дата(2014, 01, 02);
НовоеЗначение.Значение = 250;
НовоеЗначение.Аналитика1 = "extValue-Розница";

НовоеЗначение = ЗначенияПоказателя.Добавить();
НовоеЗначение.Дата = Дата(2014, 01, 03);
НовоеЗначение.Значение = 150;
НовоеЗначение.Аналитика1 = "extValue-Розница";

НовоеЗначение = ЗначенияПоказателя.Добавить();
НовоеЗначение.Дата = Дата(2014, 01, 04);
НовоеЗначение.Значение = 300;
НовоеЗначение.Аналитика1 = "extValue-Розница";

```

---

Возврат ЗначенияПоказателя;

КонецФункции

## 5. В методе ПолучитьТаблицуОписанийПоказателей() формируем структуру параметра, согласно колонкам таблицы значений ОписанияПоказателей:

```

Функция ПолучитьТаблицуОписанийПоказателей() Экспорт
ОписанияПоказателей = Новый ТаблицаЗначений;
ОписанияПоказателей.Колонки.Добавить("ТипСущности");
ОписанияПоказателей.Колонки.Добавить("Наименование");
ОписанияПоказателей.Колонки.Добавить("Описание");
ОписанияПоказателей.Колонки.Добавить("ИдПоказателя");
ОписанияПоказателей.Колонки.Добавить("Шкала");
ОписанияПоказателей.Колонки.Добавить("Ответственный");

Показатель = ОписанияПоказателей.Добавить();
Показатель.ТипСущности = 1; // Показатель
Показатель.Наименование = "Продажи по дням";
Показатель.Описание = "Продажи по дням";
Показатель.ИдПоказателя = "salesParameterID";
Показатель.Шкала = ПолучитьОписаниеШкалыПродаж();
Показатель.Ответственный = ПолучитьОтветственногоЗаПоказатели();
КонецФункции

```

## 6. Если у показателя могут быть аналитики, то их значения надо описать в процедуре ПолучитьОписаниеАналитикПоказателей().

```

Функция ПолучитьОписаниеАналитикПоказателей() Экспорт
ОписаниеАналитик = Новый Структура("ТипыАналитик, ЗначенияАналитик");

ТипыАналитик = Новый ТаблицаЗначений;
ТипыАналитик.Колонки.Добавить("ИдТипа");
ТипыАналитик.Колонки.Добавить("ИмяТипа");

НоваяСтрока = ТипыАналитик.Добавить();
НоваяСтрока.ИдТипа = "extType-Подразделения";
НоваяСтрока.ИмяТипа = "Подразделения";

НоваяСтрока = ТипыАналитик.Добавить();
НоваяСтрока.ИдТипа = "extType-ТипНоменклатуры";
НоваяСтрока.ИмяТипа = "Тип номенклатуры";

ЗначенияАналитик = Новый ТаблицаЗначений;
ЗначенияАналитик.Колонки.Добавить("ИдТипа");
ЗначенияАналитик.Колонки.Добавить("ИдЗначения");
ЗначенияАналитик.Колонки.Добавить("ИмяЗначения");

НоваяСтрока = ЗначенияАналитик.Добавить();
НоваяСтрока.ИдТипа = "extType-Подразделения";

```

---

```

НоваяСтрока.ИдЗначения = "extValue-Розница";
НоваяСтрока.ИмяЗначения = "Точки розничной продажи";

НоваяСтрока = ЗначенияАналитик.Добавить();
НоваяСтрока.ИдТипа = "extType-Подразделения";
НоваяСтрока.ИдЗначения = "extValue-Интернет";
НоваяСтрока.ИмяЗначения = "Интернет-магазин";

НоваяСтрока = ЗначенияАналитик.Добавить();
НоваяСтрока.ИдТипа = "extType-ТипНоменклатуры";
НоваяСтрока.ИдЗначения = "extValue-Товары";
НоваяСтрока.ИмяЗначения = "Товары";

НоваяСтрока = ЗначенияАналитик.Добавить();
НоваяСтрока.ИдТипа = "extType-ТипНоменклатуры";
НоваяСтрока.ИдЗначения = "extValue-Услуги";
НоваяСтрока.ИмяЗначения = "Услуги";

ОписаниеАналитик.ТипыАналитик = ТипыАналитик;
ОписаниеАналитик.ЗначенияАналитик = ЗначенияАналитик;
Возврат ОписаниеАналитик;

```

КонецФункции

## 7. Для формирования шкалы показателя используем вспомогательный метод ПолучитьОписаниеШкалыПродаж():

```

Функция ПолучитьОписаниеШкалыПродаж()
    Шкала = Новый Структура();
    Шкала.Вставить("Периодичность", 0);
    Шкала.Вставить("Идентификатор", "scaleID-sales");
    Шкала.Вставить("Наименование", "Продажи");
    Шкала.Вставить("ЕдиницаИзмерения", "млн руб.");
    Шкала.Вставить("ЕдиницаИзмеренияИд", "unitID-millionRUB");
    Шкала.Вставить("ТипАналитики1", "extType-Подразделения");
    Шкала.Вставить("ТипАналитики2", "extType-ТипНоменклатуры");

    Возврат Шкала;
КонецФункции

```

## 8. Для формирования ответственного за показатель используем метод ПолучитьОтветственногоЗаПоказатели():

```

Функция ПолучитьОтветственногоЗаПоказатели()
    Ответственный = Новый Структура;
    Ответственный.Вставить("Имя", "Петров П.П.");
    Ответственный.Вставить("Телефон", "+7(123)444-55-66");
    Ответственный.Вставить("ЭлектроннаяПочта", "petrov.mmp@rarus.ru");
    Ответственный.Вставить("ЭлектронныйАдрес", "http://rarus.ru/mobile-monitor");

    Возврат Ответственный;
КонецФункции

```

9. Для использования шифрования данных в модуле менеджера обработки «mmСгурто» отредактируем текст функции «СформироватьТаблицуСоответствийКлючейШифрования», где опишем соответствия между ключами и идентификаторами источников, показателей и аналитик.

Функция СформироватьТаблицуСоответствийКлючейШифрования() Экспорт

```
Таблица = Новый ТаблицаЗначений;

Таблица.Колонки.Добавить("ИдКлюча");
Таблица.Колонки.Добавить("ИдИсточника");
Таблица.Колонки.Добавить("ИдПоказателя");
//Таблица.Колонки.Добавить("ИдАналитики1");
//Таблица.Колонки.Добавить("ИдАналитики2");
//Таблица.Колонки.Добавить("ИдАналитики3");

НовСтрока = Таблица.Добавить();
НовСтрока.ИдКлюча = "scale.sales";
НовСтрока.ИдИсточника = "TD";
НовСтрока.ИдПоказателя = "salesParameterID";
//НовСтрока.ИдАналитики1 = "";
//НовСтрока.ИдАналитики2 = "";
//НовСтрока.ИдАналитики3 = "";
```

Возврат Таблица;

КонецФункции

10. Настройки начального отображения показателей на мобильном устройстве надо отредактировать в макете НастройкиИсточника обработки «mmИнтерфейсИсточника». Для шкалы показателя «Продажи за день» установим:

- Формат даты – дд.мм.гггг;
- Отображение значения серии – значения показателей;
- Внешний вид серии – график;
- Видимая шкала – последние полгода;
- Размер окна - последний месяц;
- Период агрегации – день;
- Агрегирующая функция - сумма.

Также настроим внешний вид графика:

- Отображать точки – истина
- Сглаживание точек – истина
- Заливка – истина
- Порядок - 1

<!-- НАСТРОЙКИ ШКАЛ И ПОКАЗАТЕЛЕЙ, НЕОБЯЗАТЕЛЬНЫЙ ЭЛЕМЕНТ -->

```
<scaleSettings>
    <scale source_id="TD" scale_id="scaleID-sales"
    date_format="dd.мм.yyyy" series_type="value">
```

---

```

series_appearance_type="line" visible_scale="6monthes" window="month"
aggregation_period="day" aggregation_function="sum">

<parameterSettings>
  <parameter source_id="TD" parametr_id="salesParameterID"
    points="yes" smoothing="yes" fill="yes" order=1/>
</parameterSettings>
</scale>
</scaleSettings>

```

В модуле менеджера обработки «mmMobileMonitor» данные для показателя выбираются из вспомогательного регистра, где они уже заранее подготовлены в необходимом формате (для тестирования примера необходимо данные в регистр забить вручную). Алгоритмы подготовки данных в макете не реализованы и остаются для реализации под нужды конкретной системы. Возможны следующие варианты реализации:

- Прямая запись в регистр «mmParameters» при проведении первичных документов (Например, при проведении документа «РеализацияТоваровУслуг»);
- Реализация механизма подписки на события с последующей записью в регистр «mmParameters», при возникновении события (Например, отлавливаем события изменения регистра «ВыручкаИСебестоимостьПродаж» и при его изменении записываем данные в регистр «mmParameters»).
- Реализация фонового задания записывающего в регистр «mmParameters» данных в определенное время (Например, каждый день в конце рабочего дня получаем данные из регистра «ВыручкаИСебестоимостьПродаж» за этот день и записываем в «mmParameters»).
- Использование процедур  
 ЗаписатьЗначенияПоказателяВРегистрММ(),  
 ЗаписатьОповещениеВРегистрММ(),  
 ЗаписатьСогласованияВРегистрММ() (Например, для записи данных показателя «Продажи по дням» в процедуре ЗаписатьЗначенияПоказателяВРегистрММ() формируем структуру заголовка и таблицу с значениями показателя,

---

вызываем одноименную процедуру из модуля обработки «mmMobileMonitor» и передаем ей эти параметры)

- В модуле менеджера обработки «mmИнтерфейсИсточника» в функции ПолучитьСписокПоказателейИсточников() при добавлении показателя указать название функции - обработчика, которая будет выполняться при обращении к конфигурации. Данная функция должна быть описана в модуле менеджера этой обработки с ключевым словом Экспорт. Входными параметрами является структура с полями ИдПоказателя, ИдИсточника, ДатаОбновления, а результатом должна быть таблица значений с данными показателя.

Возможно также получение данных напрямую из исходных данных (в данном случае, например, из регистра «ВыручкаИСебестоимостьПродаж»). Для этого нужно отредактировать текст запроса в процедуре «ПолучитьЗначенияПоказателя()» в модуле менеджера «mmMobileMonitor», но структура и названия полей запроса должны соответствовать приведенному в примере запросу. Установить условие на начало и конец периода, значения которых получаются с мобильного устройства. Для преобразования границ периода в формат даты используется функция «ПреобразоватьДатуИЗФорматаПротоколаММ()». На каждую запись результата запроса создается значение XDTO и добавляется в массив точек. Для корректного обновления данных на мобильном устройстве необходимо в timestamp записывать самую позднюю дату из полученного пакета.

### **Реализация поставленной задачи (Обратный протокол)**

Для добавления показателя необходимо выполнить следующие действия:

1. Открыть модуль менеджера обработки «mmИнтерфейсИсточника»;

2. Так как мы не рассматриваем в примере работу с оповещениями и согласованиями, то методы `ОтправитьНаСерверММСогласования()`, `ПолучитьОтветыНаСогласованияССервераММ()` и `ОтправитьНаСерверММОповещения()` модификации не подлежат.
3. Для определения списка источников необходимо модифицировать процедуру `ОтправитьНаСерверММСписокИсточников()`, введя параметры своего источника данных:

```
Процедура ОтправитьНаСерверММСписокИсточников (
    Источники = Новый ТаблицаЗначений;
    Источники.Колонки.Добавить ("Наименование");
    Источники.Колонки.Добавить ("ИдИсточника");

    Источник = Источники.Добавить();
    Источник.Наименование = "Торговый дом " "Комплексный" " ";
    Источник.ИдИсточника = "TD";
    Результат =
    Обработки.ммОбратныйПротокол.ОтправитьНаСерверММСписокИсточников(Источники);

    Если ЗначениеЗаполнено(Результат) Тогда
        // Если не пустая строка, то возникла ошибка
        Сообщить(Результат);
    КонецЕсли;
КонецПроцедуры
```

4. Далее необходимо сформировать правильный список показателей передаваемых на сервер в методе `ОтправитьНаСерверММСписокПоказателей()`:

```
Процедура ОтправитьНаСерверММСписокПоказателей (
    Соответствия = Новый ТаблицаЗначений;
    Соответствия.Колонки.Добавить ("ИдПоказателя");
    Соответствия.Колонки.Добавить ("ИдИсточника");

    ПоказательИсточника = Соответствия.Добавить();
    ПоказательИсточника.ИдПоказателя = "salesParameterID";
    ПоказательИсточника.ИдИсточника = "TD";

    Результат =
    Обработки.ммОбратныйПротокол.ОтправитьНаСерверММСписокПоказателей(
    Соответствия);
    Если ЗначениеЗаполнено(Результат) Тогда
        // Если не пустая строка, то возникла ошибка
```



---

```
Сообщить (Результат) ;
КонецЕсли ;
КонецПроцедуры
```

5. В методе `postParametersList()` вызываем вспомогательный метод `ПолучитьParameterHeaderПоказателя()`, который формирует название и структуру параметра согласно структуре протокола:

Функция `ПолучитьParameterHeaderПоказателя(Прокси, СерверURI)`

```
ЗаголовокПараметра =
Прокси.ФабрикаXDTO.Создать(Прокси.ФабрикаXDTO.Тип(СерверURI,
"ТParameterHeader"));
ЗаголовокПараметра.parameterType = 1;
ЗаголовокПараметра.parameterName = "Продажи по дням";
ЗаголовокПараметра.shortName = "Sale";
ЗаголовокПараметра.parameterContent = "Продажи по дням. Торговый дом
"Комплексный" ";
ЗаголовокПараметра.parameterID = "salesParameterID";
ЗаголовокПараметра.addData = "";
ЗаголовокПараметра.sourceID = "TD";
ЗаголовокПараметра.scale = ПолучитьШкалуПродажи(Прокси,
СерверURI);
ЗаголовокПараметра.person = ПолучитьОтветственного(Прокси,
СерверURI);

Возврат ЗаголовокПараметра;
КонецФункции
```

6. Для формирования шкалы показателя и ответственного за данный показатель используем вспомогательные методы `ПолучитьШкалуПродажи()`, и `ПолучитьОтветственного()` соответственно:

Функция `ПолучитьШкалуПродажи(Прокси, СерверURI)`

```
Шкала = Прокси.ФабрикаXDTO.Создать(Прокси.ФабрикаXDTO.Тип(СерверURI,
"ТScaleData"));

Шкала.scaleType = 1;
Шкала.period = 6;
Шкала.minValue = 0;
Шкала.maxValue = 65536;
Шкала.unitName = "млн руб.";
Шкала.unitID = "unitID-millionRUB";
Шкала.scaleID = "scaleID-sales";
Шкала.scaleName = "Продажи";
Шкала.addData = "";
Шкала.extDimensionType1 = "";
Шкала.extDimensionType2 = "";
Шкала.extDimensionType3 = "";

Возврат Шкала;
КонецФункции
```

```

Функция ПолучитьОтветственного(Прокси, СерверURI)
    PersonData =
        Прокси.ФабрикаXDTO.Создать(Прокси.ФабрикаXDTO.Тип(СерверURI,
            "TPersonData"));
    PersonData.name = "Петров П.П.";
    PersonData.phone = "+7(123)111-22-33";
    PersonData.email = "mob.dev@rarus.ru";
    PersonData.url = "http://rarus.ru/mobile-monitor";
    Возврат PersonData;
КонецФункции

```

11. Далее необходимо исправить метод `ОтправитьНаСерверММЗначенияПоказателя()`, добавив код получения конкретных значений конкретных показателей - в нашем случае показатель «Продажи по дням» и его значение на каждый день:

```

Процедура ОтправитьНаСерверММЗначенияПоказателя()
    ЗаголовокПоказателя = Новый Структура("ИдПоказателя, ИдИсточника");
    ЗаголовокПоказателя.ИдПоказателя = "salesParameterID";
    ЗаголовокПоказателя.ИдИсточника = "TD";

    ЗначенияПоказателя = Новый ТаблицаЗначений;
    ЗначенияПоказателя.Колонки.Добавить("Дата");
    ЗначенияПоказателя.Колонки.Добавить("Значение");
    ЗначенияПоказателя.Колонки.Добавить("Аналитикал");

    НовоеЗначение = ЗначенияПоказателя.Добавить();
    НовоеЗначение.Дата = Дата(2014, 01, 01);
    НовоеЗначение.Значение = 200;
    НовоеЗначение.Аналитикал = "extValue-Розница";

    НовоеЗначение = ЗначенияПоказателя.Добавить();
    НовоеЗначение.Дата = Дата(2014, 01, 02);
    НовоеЗначение.Значение = 250;
    НовоеЗначение.Аналитикал = "extValue-Розница";

    НовоеЗначение = ЗначенияПоказателя.Добавить();
    НовоеЗначение.Дата = Дата(2014, 01, 03);
    НовоеЗначение.Значение = 150;
    НовоеЗначение.Аналитикал = "extValue-Розница";

    НовоеЗначение = ЗначенияПоказателя.Добавить();
    НовоеЗначение.Дата = Дата(2014, 01, 04);
    НовоеЗначение.Значение = 300;
    НовоеЗначение.Аналитикал = "extValue-Розница";

    Результат =
        Обработки.ммОбратныйПротокол.ОтправитьНаСерверММЗначенияПоказателя(За
        головокПоказателя, ЗначенияПоказателя);
    Если ЗначениеЗаполнено(Результат) Тогда
        Сообщить(Результат);
    КонецЕсли;
КонецПроцедуры

```



## ПУБЛИКАЦИЯ WEB-СЕРВИСА (ПРЯМОЙ ПРОТОКОЛ)

Перед публикаций web-сервиса необходимо убедиться, что установлен модуль расширения web-сервисов 1С:Предприятие и HTTP-сервер, например Apache (инструкция по установке и настройке приведена в документации и комплекта поставки 1С:Предприятие).

Для публикации web-сервиса мобильного монитора необходимо:

1. В конфигурации источнике добавить пользователя mobileUser с паролем mobileUser (Значения по умолчанию. Используются для прямого подключения мобильного приложения к источнику. В серверном варианте работы могут быть другими. Должны совпадать на сервере и на клиенте).
2. Пользователь должен получить права на все объекты подсистемы mmMobileMonitor и права на чтение данных, которые участвуют в запросах на получение показателей и оповещений. На объекты, участвующие в формировании согласований, необходимы права на чтение и запись.
3. Через меню «Администрирование»/ «Публикация на web-сервере» публикуем источник данных (Рисунок 11).

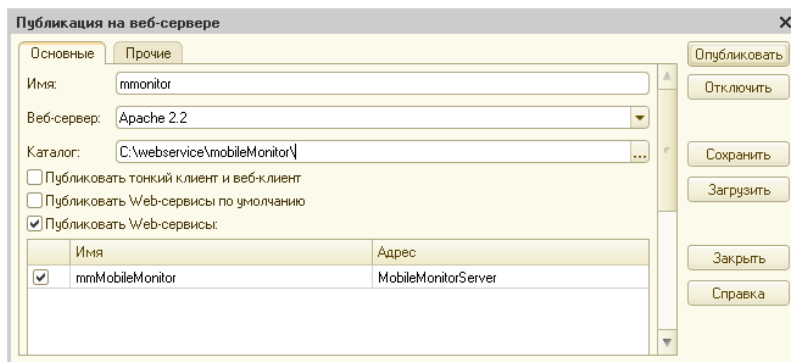


Рисунок 11. Публикация данных на web-сервисе.

4. После публикации web-сервиса необходимо проверить, что публикация прошла корректно и web-сервис доступен для запросов. Для этого необходимо в браузере набрать строку подключения к web-сервису:
5. <http://xx.xx.xx.xx/mmonitor/ws/mmMobileMonitorServer?wsdl>  
где xx.xx.xx.xx – адрес сервера, на котором развернут web-сервис.

Если web-сервис работает корректно, то браузер отобразит xml описание web-сервиса, похожее на представленное на рисунке 12.

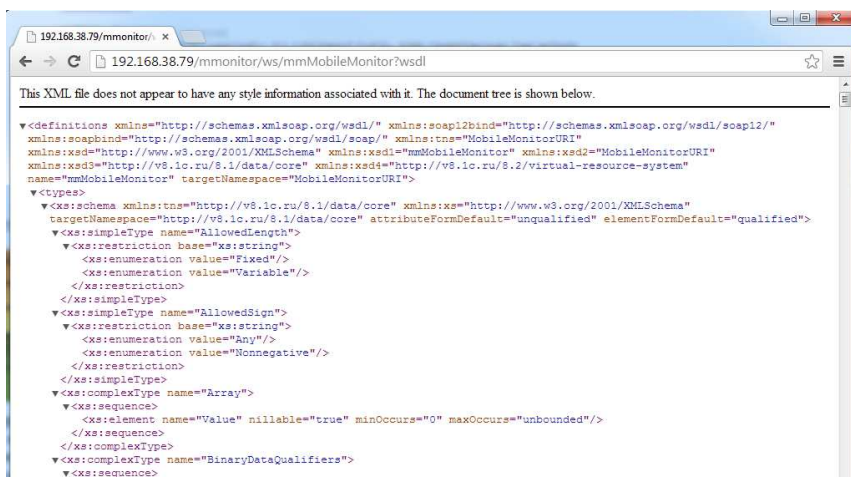


Рисунок 12. Ответ web-сервиса.

## ОГРАНИЧЕНИЯ ПРЯМОГО ПОДКЛЮЧЕНИЯ К БАЗЕ «С:ПРЕДПРИЯТИЕ (ПРЯМОЙ ПРОТОКОЛ)

В случае, если подключение мобильного приложения выполняется непосредственно к информационной базе, без использования сервера ММП, существуют некоторые ограничения:

1. Публикация должна иметь адрес `http://localhost/mmonitor/ws/mmMobileMonitorServer`

Вместо localhost можно использовать любой ip-адрес или доменное имя, а также номер порта.

Например, <http://trade11/mmonitor/ws/mmMobileMonitorServer>

Если необходима публикация большего количества баз или по нестандартному адресу, то необходимо использовать сервер ММП.

---

## ОПИСАНИЕ ПРОТОКОЛА ОБМЕНА ДАННЫМИ

Прямой протокол обмена данными

---

### Типы данных

---

В этом разделе описаны все типы данных используемые в протоколе.

#### ***LoginStruct***

- login: String – Логин пользователя
- password: String – Пароль пользователя
- deviceID: String – Идентификатор мобильного устройства
- sessionID: String – Идентификатор сеанса связи устройства с сервером

Структура предназначена для авторизации на сервере. При авторизации при прямом подключении к источнику IC:Предприятия данные из *LoginStruct* не анализируются. Параметры *login* и *password* используются для идентификации пользователя и определения его прав, *deviceID* и *sessionID* для учета лицензий.

#### ***GetSettingsData***

- sources: array of string

Структура с массивом уникальных идентификаторов источников.

#### ***GetSettingsResult***

- GetSettingsResult: sourceSettingsData
- loginResult: LoginResultStruct

Структура содержит настройку по умолчанию для данного веб-сервиса и описание ошибки, если таковая возникла при формировании настроек.

#### ***sourceSettingsData***

- sourceSettings: array of sourceSettings

Тип данных содержит массив из элементов *sourceSettings* с настройками по умолчанию для всех источников, опубликованных с помощью данного веб-сервиса.

#### ***sourceSettings***

- sourceID: String
- XMLSettings: String

---

Тип данных содержит идентификатор источника и настройку визуального представления по умолчанию для показателей данного источника.

### ***GetSourcesResult***

- sources: array of SourceData
- loginResult: LoginResultStruct

Структура содержит перечень всех источников опубликованных в данной публикации веб-сервиса и описание ошибки, если таковая возникла при формировании списка источников.

### ***SourceData***

- name: string
- sourceID: string
- addData: string

Структура содержит информацию об источнике данных - имя источника, его уникальный идентификатор и вспомогательную информацию об источнике.

### ***GetParamsListResult***

- data: ParamsListData
- loginResult: LoginResultStruct

Структура содержит перечень всех параметров для данного веб-сервиса и описание ошибки, если таковая возникла при формировании списка параметров.

### ***ParamsListData***

- headers: array of ParameterHeader

Структура содержит массив заголовков всех параметров.

### ***ParameterHeader***

- parameterType: int – тип параметра (1-показатель, 2-оповещение, 3- согласование)
- parameterName:string – наименование параметра
- shortName:string – короткое наименование
- parameterContent:string – детальное описание параметра
- parameterID:string – уникальный идентификатор параметра
- addData: string – дополнительная информация (не используется, введено для расширения протокола)
- sourceID:string – уникальный идентификатор источника данных



- 
- scale:ScaleData – шкала, на которой происходит группировка параметра
  - person:PersonData – данные об ответственном за параметр лице

Тип данных содержит заголовочную информацию о параметре.

### ***ScaleData***

- type:int – тип шкалы (1 – числовая, 2 – перечисляемая)
- period: int – период агрегации данных в шкале (
  - 0 - не выполняется;
  - 1 – секунда;
  - 2 – минута;
  - 3 – час;
  - 4 – сутки;
  - 5 – неделя;
  - 6 – месяц;
  - 7 – квартал;
  - 8 - пол года;
  - 9 – год)
- minValue:float – минимальное значение шкалы, не обязательно к заполнению
- maxValue:float – максимальное значение шкалы, не обязательно к заполнению
- unitName:string – единица измерения шкалы
- unitID:string – идентификатор единицы измерения
- scaleID:string – идентификатор шкалы
- scaleName:string – наименование шкалы

Тип данных содержит информацию о шкале.

### ***GetParamData***

- sourceID: String – уникальный идентификатор источника
- parametrID: String – уникальный идентификатор параметра
- dateFrom: date – дата начала для выборки данных показателя
- dateTo: date – дата окончания для выборки данных показателя

### ***GetParameterResult***

- data: ParameterData
- loginResult: LoginResultStruct

Структура содержит информацию о параметре и описание ошибки, если таковая возникла при формировании параметра.

### ***ParameterData***

- header: ParameterHeader – конкретный показатель, для которого нужно передать значения
- points: array of ParamPoint – массив точек

Структура, содержащая все точки для конкретного показателя

### ***ParamPoint***

- dateTime:dateTime – Дата и время значения точки
- valueFloat:float – значение точки
- valueStr:string – строковое значение точки (для шкал с перечисляемыми значениями, пока не поддерживается)
- extDimensionValue1: String – значение аналитики 1 (пока не поддерживается)
- extDimensionValue2: String – значение аналитики 2 (пока не поддерживается)
- extDimensionValue3: String – значение аналитики 3 (пока не поддерживается)

Тип данных содержащий значение конкретной точки.

### ***GetNewsData***

- parametrID: String
- sourceID: String
- timestamp: String

Структура содержит информацию о группе оповещений и источнике, для которых необходимо получить оповещения. В поле timestamp передаётся информация об дате и времени последнего обновления оповещений.

### ***GetNewsResult***

- data: NewsData
- loginResult: LoginResultStruct

Структура содержит информацию о сформированных оповещениях и описание ошибки, если таковая возникла при формировании оповещений.

### ***NewsData***

- news: array of NewsStruct
- count: int

Структура содержит массив оповещений и их количество.

### ***NewsStruct***

- newsID: string – уникальный идентификатор оповещения
- parametrID: String – уникальный идентификатор параметра (группа оповещений)

- 
- sourceID: String – уникальный идентификатор источника
  - body: string – текст оповещения
  - date: dateTime – дата и время оповещения
  - link: string – ссылка на источник оповещения (адрес сервера и сайта)
  - topic: string – тема сообщения
  - source: string – не используется
  - addData: String – поле для передачи дополнительной информации (для расширения протокола)
  - alertType: int – уровень важности предупреждения (
    - 0 - обычное оповещение. Поле никак не обрабатывается.
    - 1 - Информационное извещение. Не ошибка.
    - 2 - Предупреждение.
    - 3 - Информация (например, к размышлению)
    - 4 - Ошибка первого уровня.
    - 5 - Критическая ошибка)
  - alertParametrID: String – уникальный идентификатор параметра, который вызвал предупреждение
  - alertSourceID: String – уникальный идентификатор источника параметр которого вызвал предупреждение

Тип данных для хранения информации об оповещении.

### ***GetAgreesData***

- parametrID: String – уникальный идентификатор параметра (типа согласования)
- sourceID: String – уникальный идентификатор источника
- timestamp: String – дата и время последнего получения согласований
- lastAgreement: String – не используется

Структура содержит информацию о группе оповещений и источнике, для которых необходимо получить оповещения. В поле timestamp передаётся информация об дате и времени последнего обновления оповещений.

### ***GetAgreesResult***

- data: AgreesData
- loginResult: LoginResultStruct

Структура содержит информацию о сформированных согласованиях и описание ошибки, если таковая возникла при формировании оповещений.

### ***AgreesData***

- news: array of AgreementStruct
- count: int

---

Структура для хранения массива всех согласований и их количества.

### ***AgreementStruct***

- agreeID: string – уникальный идентификатор согласования
- parametrID: String – уникальный идентификатор параметра (вида согласования)
- sourceID: String – уникальный идентификатор источника согласования
- dateTime: dateTime – дата и время согласования
- topic: string – заголовок согласования
- content: string – текст согласования
- buttons: string – набор вариантов принятия решений для согласований
- isProcessed: Boolean – признака обработанности согласования
- answer: string – выбранный пользователем вариант ответа на согласование
- userAnswer: string – идентификатор пользователя ответившего на согласование
- timestamp: string – временная отметка о моменте согласования
- addData: String – дополнительная информация (для расширения протокола)
- alertType: int – уровень важности согласования (
  - 0 - Обычное оповещение. Поле никак не отрабатывается.
  - 1 - Информационное извещение. Не ошибка.
  - 2 - Предупреждение.
  - 3 - Информация (например, к размышлению)
  - 4 - Ошибка первого уровня.
  - 5 - Критическая ошибка)

Тип данных для хранения информации о согласовании.

### ***PutAgreementsData***

- answers: array of AgreementAnswer

Структура, в которой хранятся все согласования подлежащие обработке.

### ***AgreementAnswer***

- parametrID: String – уникальный идентификатор параметра (типа согласования)
- sourceID: String – уникальный идентификатор источника согласований
- agreeID: string – уникальный идентификатор согласования
- button: string – принятое решение
- comment: string – комментарий пользователя

- 
- `userAnswer: string` – пользователь обработавший согласование

Тип данных для описания результат согласования.

### ***PutAgreementsResult***

- `data: ResultData`
- `loginResult: LoginResultStruct`

Структура для возврата результата загрузки согласований. Содержит все по тем или иным причинам не загруженные согласования и описания ошибок загрузки.

### ***ResultData***

- `success: int` – количество успешно загруженных согласований
- `fail: int` – количество не загруженных согласований
- `fails: array of PutAgreementsFails` – массив не загруженных согласований и описания ошибок по загрузке.

Структура для хранения информации об не обработанных при загрузке согласований.

### ***PutAgreementsFails***

- `agreementID: string` – уникальный идентификатор согласования
- `comment: string` – описание ошибки обработки согласования на источнике
- `errorCode: int` – код ошибки загрузки согласования

Тип данных для хранения ошибок загрузки согласований на источнике.

## **Методы**

---

В этом разделе описаны методы работы с веб-сервисом

### **CheckSrv**

#### **Синтаксис:**

*CheckSrv(checkSrvData): CheckSrvResultData*

#### **Описание:**

Предназначен для проверки доступности сервера (источника) данных. Результат формируется сервером (при прямом подключении к источнику – пустое значение) автоматически и предназначен для распознавания подлинности сервера данных.

#### **Параметры:**

*checkSrvData: string*

Открытый ключ для идентификации сервера мобильным устройством. На основании него, по предопределенному алгоритму генерируется ответ сервера.

**Возвращаемое значение:**

*CheckSrvResultData: string*

Результат идентификации сервера. По данному значению мобильное устройство определяет подлинность сервера.

**GetSettings****Синтаксис:**

*GetSettings(loginStruct: LoginStruct, getSettingsData: GetSettingsData):  
GetSettingsResult*

**Описание:**

С помощью данного метода возможно передать начальные настройки (по умолчанию) представления данных с источника на мобильные устройства. Например, период агрегации данных и вид представления в виде гистограммы.

**Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

*getSettingsData: GetSettingsData*

Массив источников данных, для которых необходимо передать настройки.

**Возвращаемое значение:**

*GetSettingsResult: GetSettingsResult*

Структура, возвращающая массив настроек по умолчанию для каждого источников и описание ошибки, если таковая возникла при выполнении метода.

**GetSourcesList****Синтаксис:**

*GetSourcesList(loginStruct: LoginStruct): GetSourcesResult*

**Описание:**

Метод предназначен для формирования списка доступных источников в опубликованном веб-сервисе.

**Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

**Возвращаемое значение:**

---

***GetSourcesResult: GetSourcesResult***

Структура, возвращающая массив доступных в публикации источников и описание ошибки, если таковая возникла при выполнении метода.

**GetParametersList****Синтаксис:**

*GetParametersList(loginStruct: LoginStruct, getParamsData: GetParamsData): GetParamsListResult*

**Описание:**

Метод предназначен для формирования списка доступных параметров в опубликованном веб-сервисе.

**Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

*getParamsData: GetParamsData*

Не используется. Введено для возможности расширения протокола.

**Возвращаемое значение:**

*GetParamsListResult: GetParamsListResult*

Структура, возвращающая массив доступных в публикации параметров и описание ошибки, если таковая возникла при выполнении метода.

**GetParameter****Синтаксис:**

*GetParameter(loginStruct: LoginStruct, getParamData: GetParamData): GetParameterResult*

**Описание:**

Метод предназначен для формирования массива точек и их значений для конкретного показателя.

**Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

*getParamData: GetParamData*

Структура для передачи данных о параметре и периоде формирования выборки.

**Возвращаемое значение:**

*GetParameterResult: GetParameterResult*

Структура, возвращающая массив значений показателя и описание ошибки, если таковая возникла при выполнении метода.

---

## **GetNews**

### **Синтаксис:**

*GetNews(loginStruct: LoginStruct, getNewsData: GetNewsData):*

*GetNewsResult*

### **Описание:**

Метод предназначен для формирования списка доступных оповещений в опубликованном веб-сервисе.

### **Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

*getNewsData: GetNewsData*

Структура содержит информацию о необходимых к выборке оповещениях и времени последнего обращения за оповещениями.

### **Возвращаемое значение:**

*GetNewsResult: GetNewsResult*

Структура, возвращающая массив доступных в публикации оповещений и описание ошибки, если таковая возникла при выполнении метода.

## **GetAgreements**

### **Синтаксис:**

*GetAgreements(loginStruct: LoginStruct, getAgreesData: GetAgreesData):*

*GetAgreesResult*

### **Описание:**

Метод предназначен для формирования списка доступных согласований в опубликованном веб-сервисе.

### **Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

*getAgreesData: GetAgreesData*

Структура содержит информацию о необходимых к выборке согласованиях и времени последнего обращения за согласованиями.

### **Возвращаемое значение:**

*GetAgreesResult: GetAgreesResult*

Структура, возвращающая массив доступных в публикации согласований и описание ошибки, если таковая возникла при выполнении метода.

## **PutAgreements**

### **Синтаксис:**



---

*PutAgreements(loginStruct: LoginStruct, putAgreementsData: PutAgreementsData): PutAgreementsResult*

**Описание:**

Метод предназначен для передачи на источник (с сервера) или на сервер (с мобильного устройства) списка обработанных согласований.

**Параметры:**

*loginStruct: LoginStruct*

Структура для проверки авторизации

*putAgreementsData: PutAgreementsData*

Структура содержит информацию о всех согласованиях, которые подлежат обработке на источнике (сервере).

**Возвращаемое значение:**

*PutAgreementsResult: PutAgreementsResult*

Структура, возвращающая массив не обработанных при загрузке согласований и описания ошибок загрузки.

Обратный протокол

---

## Типы данных

---

В этом разделе описаны все типы данных используемые в протоколе

***TLoginStruct***

- login: String – Логин пользователя
- password: String – Пароль пользователя
- deviceID: String – Идентификатор мобильного устройства. Резервировано для расширения, не используется.
- sessionID: String – Идентификатор сеанса связи устройства с сервером. Резервировано для расширения, не используется.

Структура предназначена для авторизации на сервере.

***TPostSourcesListData***

- sources: TSourcesArray – описание источников

Структура предназначена для передачи на сервер списка источников.

***TSourcesArray***

- TSourceData: array of TSourceData

Структура с массивом уникальных идентификаторов источников.

***TSourceData***

- name: string
- sourceID: string
- addData: string

Структура содержит информацию об источнике данных - имя источника, его уникальный идентификатор и вспомогательную информацию об источнике.

***TPostSourcesResult***

- loginResult: TLoginResultStruct

Структура содержит результат выполнения метода *PostSourcesList*

***TLoginResultStruct***

- errorCode: int – Код ошибки
- errorMessage: String – описание ошибки
- version: String – версия сервера
- timeStamp: dateTime – текущая дата сервера

Структура содержит результат авторизации при передаче данных на сервер

***TPostSettingsResult***

- loginResult: TLoginResultStruct

Структура содержит результат выполнения метода *PostSettings*

***TPostSettingsData***

- data: TPostSettingsStruct

***TPostSettingsStruct***

- settings: TSourceSettingsArray

Структура содержит массив настроек мобильного приложения в разрезе источников

***TSourceSettingsArray***

- TSourceSettings: array of TSourceSettings

***TSourceSettings***

- sourceID: String – идентификатор источника для которого передаются настройки
- XMLSettings: String – XML настроек

Тип данных содержит идентификатор источника и настройку визуального представления по умолчанию для показателей данного источника.

---

***TPostParamsListResult***

- loginResult: TLoginResultStruct

***TPostParamsListData***

- data: TParamsListData

***TParamsListData***

- headers: TParameterHeadersArray

***TParameterHeadersArray***

- TParameterHeader: array of TParameterHeader

***TParameterHeader***

- parameterType: int – тип параметра (1-показатель, 2-оповещение, 3– согласование)  
- parameterName:string  
- shortName:string  
- parameterContent:string  
- parameterID:string  
- addData: string  
- sourceID:string  
- scale:TScaleData  
- person:TPersonData

***TScaleData***

- scaleType:int  
- period: int  
- minValue:float  
- maxValue:float  
- unitName:string  
- unitID:string  
- scaleID:string  
- scaleName:string  
- extDimensionType1:string  
- extDimensionType2:string  
- extDimensionType3:string  
- scaleValues: array of string  
- addData: string

***TPersonData***

- name:string

- phone:string
- email:string
- url:string

### ***TPostParameterResult***

- loginResult: TLoginResultStruct

### ***TPostParamData***

- header: TParameterHeader – Заголовок параметра
- points: TPointsArray – Структура, содержащая массив точек значений
- timestamp: string -

Структура, содержащая все точки для конкретного показателя

### ***TParamsPointsArray***

- TPointsArray: array of TParamPoint – массив точек параметра

### ***TParamPoint***

- dateTime:string – дата точки
- paramValue:string – значение точки
- extDimensionValue1: String – Не используется. Зарезервировано для расширения
- extDimensionValue2: String – Не используется. Зарезервировано для расширения
- extDimensionValue3: String – Не используется. Зарезервировано для расширения
- responseTime: float – время получения значения

Структура предназначена для описания точки параметра

### ***TPostNewsResult***

- loginResult: TLoginResultStruct

### ***TPostNewsData***

- data: TNewsData
- timestamp: string

### ***TNewsData***

- news: TNewsArray
- count: int

### ***TNewsArray***

- TNewsStruct: array of TNewsStruct

---

***TNewsStruct***

- newsID: string – уникальный идентификатор оповещения
- parametrID: String – уникальный идентификатор параметра (группа оповещений)
- sourceID: String – уникальный идентификатор источника
- body: string - текст оповещения
- date: string - дата и время оповещения
- link: string – ссылка на источник оповещения (адрес сервера и сайта)
- topic: string – тема сообщения
- source: string – не используется
- addData: String – поле для передачи дополнительной информации (для расширения протокола)
- alertType: int – уровень важности предупреждения (
  - 0 - обычное оповещение. Поле никак не обрабатывается.
  - 1 - Информационное извещение. Не ошибка.
  - 2 - Предупреждение.
  - 3 - Информация (например, к размышлению)
  - 4 - Ошибка первого уровня.
  - 5 - Критическая ошибка)
- alertParametrID: String – уникальный идентификатор параметра, который вызвал предупреждение
- alertSourceID: String – уникальный идентификатор источника параметр которого вызвал предупреждение

Структура предназначена для описания оповещения

***TPostAgreesResult***

- loginResult: TLoginResultStruct

***TPostAgreesData***

- data: TAgreesData
- timestamp: string

***TAgreesData***

- agrees: TAgreesArray
- count: int

***TAgreesArray***

- *TAgreementStruct*: array of *TAgreementStruct*

***TAgreementStruct***

- 
- agreeID: string – уникальный идентификатор согласования
  - parametrID: String – уникальный идентификатор параметра (вида согласования)
  - sourceID: String – уникальный идентификатор источника согласования
  - dateTime: String – дата и время согласования
  - topic: string – заголовок согласования
  - content: string – текст согласования
  - buttons: string – набор вариантов принятия решений для согласований
  - isProcessed: Boolean – признака обработанности согласования
  - answer: string – выбранный пользователем вариант ответа на согласование
  - userAnswer: string – идентификатор пользователя ответившего на согласование
  - timestamp: string – временная отметка о моменте согласования
  - addData: String – дополнительная информация (для расширения протокола)
  - alertType: int – уровень важности согласования (
    - 0 - обычное оповещение. Поле никак не отрабатывается.
    - 1 - Информационное извещение. Не ошибка.
    - 2 - Предупреждение.
    - 3 - Информация (например, к размышлению)
    - 4 - Ошибка первого уровня.
    - 5 - Критическая ошибка)

Структура предназначена для описания согласования

### ***TGetAgreementAnswersResult***

- loginResult: TLoginResultStruct
- answers: TAnswersArray

### ***TAnswersArray***

- TAgreementAnswer: array of TAgreementAnswer

### ***TAgreementAnswer***

- sourceID: String – уникальный идентификатор источника согласований
- parameterID: String – уникальный идентификатор параметра (типа согласования)
- agreeID: string – уникальный идентификатор согласования
- button: string – принятое решение
- comment: string – комментарий пользователя

---

- userAnswer: string – пользователь обработавший согласование

### ***TGetAgreementsAnswersData***

- sourceID: String

---

## Методы

---

В этом разделе описаны методы работы с веб-сервисом

### **PostSourcesList**

#### **Синтаксис:**

*PostSourcesList(loginStruct: TLoginStruct, sourcesData: TPostSourcesListData): TPostSourcesResult*

#### **Описание:**

Метод предназначен для передачи на сервер списка доступных источников в опубликованном веб-сервисе.

#### **Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*sourcesData: TPostSourcesListData*

Структура содержит информацию о всех источниках передаваемых на сервер.

#### **Возвращаемое значение:**

*TPostSourcesResult*

Результат выполнения метода.

### **PostSettings**

#### **Синтаксис:**

*PostSettings(loginStruct: TLoginStruct, postSettingsData: TPostSettingsData): TPostSettingsResult*

#### **Описание:**

Метод предназначен для передачи на сервер начальных настроек мобильного приложения

#### **Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*postSettingsData: TPostSettingsData*

---

Структура содержит информацию о настройках мобильного приложения.

**Возвращаемое значение:***TPostSettingsResult*

Результат выполнения метода.

**PostParametersList****Синтаксис:**

*PostParametersList(loginStruct: TLoginStruct, postParamsListData: TPostParamsListData): TPostParamsListResult*

**Описание:**

Метод предназначен для передачи на сервер списка параметров

**Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*postParamsListData: TPostParamsListData*

Структура содержит информацию о списке параметров.

**Возвращаемое значение:***TPostParamsListResult*

Результат выполнения метода.

**PostParameter****Синтаксис:**

*PostParameter(loginStruct: TLoginStruct, postParamData: TPostParamData): TPostParameterResult*

**Описание:**

Метод предназначен для передачи на сервер значений параметра

**Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*postParamData: TPostParamData*

Структура содержит информацию о списке точек параметра.

**Возвращаемое значение:***TPostParameterResult*

Результат выполнения метода.

**PostNews****Синтаксис:**

*PostNews(loginStruct: TLoginStruct, postNewsData: TPostNewsData): PostNewsResult*



---

**Описание:**

Метод предназначен для передачи на сервер списка оповещений

**Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*postNewsData: TPostNewsData*

Структура содержит список оповещений передаваемых на сервер.

**Возвращаемое значение:**

*PostNewsResult*

Результат выполнения метода.

**PostAgreements****Синтаксис:**

*PostAgreements(loginStruct: TLoginStruct, postAgreesData: TPostAgreesData): TPostAgreesResult*

**Описание:**

Метод предназначен для передачи на сервер списка согласований

**Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*postAgreesData: TPostAgreesData*

Структура содержит список согласований передаваемых на сервер.

**Возвращаемое значение:**

*TPostAgreesResult*

Результат выполнения метода.

**GetAgreementsAnswers****Синтаксис:**

*GetAgreementsAnswers(loginStruct: TLoginStruct, getAgreementsAnswersData: TGetAgreementsAnswersData): TGetAgreementsAnswersResult*

**Описание:**

Метод предназначен для получения с сервера результатов согласований

**Параметры:**

*loginStruct: TLoginStruct*

Структура для выполнения авторизации

*getAgreementsAnswersData: TGetAgreementsAnswersData*

Структура содержит список результатов согласований.

**Возвращаемое значение:**

---

## *TGetAgreementsAnswersResult*

Результат выполнения метода.

---

## Предопределенные настройки

---

В этом разделе описаны предопределенные настройки отображения показателей на мобильном устройстве. Текст настройки находится в макете «НастройкиИсточника» обработки «mmИнтерфейсИсточника».

### Таблица привязок ключей

В данном пункте описываются настройки привязок идентификаторов источников, показателей и ключей.

*source\_id* – идентификатор источника;

*param\_id* – идентификатор показателя;

*key\_id* – идентификатор ключа;

*extDimension1\_id*, *extDimension2\_id*, *extDimension3\_id* –  
идентификаторы аналитик.

### Ключи шифрования

Ключи шифрования можно получить с помощью утилиты MMKeysEdit.exe – «Редактор хранилища ключей мобильного монитора».

Для этого надо ввести наименование ключа и указать начало действия ключа, программа сгенерирует идентификатор и бинарные ключи на каждую указанную дату.

*id* – идентификатор ключа;

*name* – наименование ключа;

*from* – начало действия ключа в формате «уууу-ММ-дд»;

*binary* – бинарный ключ.

### Настройки шкал и показателей

В данном пункте описываются настройки внешнего вида шкал, периода агрегации и т.п.

*date\_format* – определяет представление даты на мобильном устройстве

---

Возможные значения:

- dd.MM.yyyy;
- yyyy;
- dd.MMM.yyyy;
- dd.MM.yy;
- dd.MM;
- dd.MMM;
- MMMM;
- MMMM.yyyy;
- MMMM.yy;
- hh:mm:ss;
- dd.MM.yy hh.mm.ss.

***series\_appearance\_type*** – определяет внешний вид серии показателей.

Возможные значения:

- line – график;
- bar – гистограмма;
- line\_bar – график + гистограмма;
- pie\_chart – круговая диаграмма;
- stack\_bar – гистограмма с накоплением;
- stack\_line – график с накоплением;
- stack\_bar\_line – вертикальная гистограмма с накоплением;
- bar\_3d – гистограмма 3D;
- pie\_chart\_3d – круговая диаграмма 3D;

***series\_value*** – определяет аналитический разрез показателя (серии)

Возможные значения:

- source – источник;
- parameter – параметр;
- extDimension1 – аналитика1;
- extDimension2 – аналитика2;
- extDimension3 – аналитика3.

***series\_type*** - хранит тип выводимого значения.

Возможные значения:

- value – значения показателей;
- sum – сумма показателей;
- value\_sum – значения + сумма показателей;
- increase\_value – значения с нарастающим итогом;

- 
- `increase_value_and_sum` – значения + сумма с нарастающим итогом;
  - `increase_sum` – сумма с нарастающим итогом.

***visible\_scale*** – определяет период шкалы, отображаемой на мобильном устройстве.

Возможные значения:

- `day` – последний день;
- `week` – последняя неделя;
- `month` – последний месяц;
- `quarter` – последний квартал;
- `6monthes` – последние полгода;
- `year` – последний год;
- `3year` – последние три года;
- `all` – все значения.

***window*** – определяет период шкалы, на котором будут отображаться значения показателей.

Возможные значения:

- `day` – последний день;
- `week` – последняя неделя;
- `month` – последний месяц;
- `quarter` – последний квартал;
- `6monthes` – последние полгода;
- `year` – последний год;
- `3year` – последние три года.

***aggregation\_period*** – определяет период агрегации (группировки значений).

Возможные значения:

- `hour` – час;
- `day` – день;
- `week` – неделя;
- `month` – месяц;
- `quarter` – квартал;
- `6monthes` – полгода;
- `year` – год.

***aggregation\_function*** – агрегирующая функция. Определяет функцию, по которой собираются значения за период.

Возможные значения:

- `min` – минимум;

- 
- *max* – максимум;
  - *avg* – среднее;
  - *sum* – сумма.

*order* – порядковый номер представления показателя в списке, создаваемого на основании шкалы.

## Настройки показателей

В этом пункте определяются настройки, относящиеся к конкретному показателю.

*points* – определяет отображать или нет точки на графиках.

Возможные значения:

- *yes* – отображать;
- *no* – не отображать.

*smoothing* – определяет сглаживание точек на графиках

Возможные значения:

- *yes* – сглаживать;
- *no* – не сглаживать.

*fill* – определяют заливку на графиках

Возможные значения:

- *yes* – заливка включена;
- *no* – заливка не включена.

*order* – определяет порядковый номер показателя внутри представления.

---

## Шифрование данных

Для использования шифрования данных надо выполнить следующее:

1. В модуле менеджера обработки «*mmИнтерфейсИсточника*» отредактировать текст функции «СформироватьТаблицуСоответствийКлючейШифрования», где описать соответствия между ключами и идентификаторами источников, показателей, аналитик. Идентификатор источника должен быть заполнен, идентификатор показателя может

оставаться пустым, в таком случае ключ будет выбираться по идентификатору источника.

**ИдКлюча** - идентификатор ключа, которым будет зашифрованы значения;

**ИдИсточника** - идентификатор источника, значения из которого будут шифроваться данным ключом;

**ИдПоказателя** - идентификатор показателя из источника, значения из которого будут шифроваться данным ключом. Не обязательный элемент.

**ИдАналитики1**, **ИдАналитики2**, **ИдАналитики3** - идентификаторы значения аналитик. Не обязательный элемент.

2. Запустить утилиту MMKeysEdit.exe – «Редактор хранилища ключей мобильного монитора». Ввести наименование ключа и указать начало действия, программа сгенерирует идентификатор и бинарные ключи на каждую указанную дату. Сохранить сгенерированные ключи на диск. По умолчанию файл с ключами должен находиться в каталоге "C:\Rarus\mmCrypto\keys" или в его подкаталогах. Если поместить в другую папку, то надо отредактировать путь к схемам в вызовах процедуры ЗагрузитьСхемуКлючей() в модуле менеджера обработки «mmCrypto».

Поиск файла с таблицей соответствия ключей происходит в каталоге и всех его подкаталогах, поэтому надо контролировать, чтобы в подкаталогах хранились только корректные файлы с расширением .xml.

Если у пользователя есть ограничения на доступ к зашифрованным показателям, то в таблицу соответствия записываются только разрешенные комбинации. Можно поместить в каталог несколько файлов с ключами, и в зависимости от прав доступа выбирать соответствующий файл для каждого пользователя.

3. Установить истинное значение константы «mmШифрованиеДанных».
4. В режиме предприятия открыть обработку «mmCrypto» - «ММ: Шифрование», выбрать файл, в который сохранили ключи в предыдущем пункте, и нажать кнопку «Сохранить».

