

М | О | ↑ | И | В

Система оперативного управления компанией

## Руководство программиста

2.5

# Оглавление

1. Введение .....	7
2. Использование API-функций системы Мотив.....	8
2.1. Структура входного XML.....	8
2.2. testConnection – тестирование соединения.....	9
2.3. Вызов API-функции.....	9
2.4. Результат выполнения API-функции.....	10
2.5. Работа API-функций с файлами.....	10
2.6. Работа с Rest API.....	11
2.7. Описание параметров вызова API-функций.....	12
2.7.1. Функции для работы с подразделениями.....	12
2.7.1.1. getpermissionsforemployer – получение списка пользователей подразделения.....	12
2.7.1.2. departmentMove – перемещение подразделения.....	14
2.7.1.3. getDepartmentHead - получение руководителя подразделения.....	14
2.7.1.4. getEmployersOGV - получение органа власти пользователя .....	15
2.7.2. Функции для работы с пользователями.....	16
2.7.2.1. userAdd – создание пользователя.....	16
2.7.2.2. userRemove – удаление пользователя .....	19
2.7.2.3. getEmployerID – получение ID пользователя по его логину и паролю.....	20
2.7.2.4. userGetInfo – получение информации о пользователях из справочника «Адресная книга».....	21
2.7.2.5. getPicture – получение изображения, прикрепленного к записи справочника .....	23
2.7.2.6. getEmployerDepartmentUserTree – получение информации о пользователях и их полномочиях.....	24
2.7.3. Функции для работы с задачами.....	25
2.7.3.1. addTask – создание задачи.....	25
2.7.3.2. addTaskByTemplate – создание задачи по шаблону.....	31
2.7.3.3. editTask – редактирование задачи.....	40
2.7.3.4. addAction – добавление действия в задачу.....	42
2.7.3.5. editAction – редактирование действия .....	43
2.7.3.6. gettaskbuttons – получение списка действий по задаче .....	44

2.7.3.7. getTaskActions – постраничный вывод действий по задаче .....	45
2.7.3.8. getTaskAttachments – получение списка прикрепленных к действиям файлов.....	48
2.7.3.9. MakeSubtask – перевод задач в подзадачи.....	49
2.7.3.10. checkpointCreate – создание контрольной точки.....	50
2.7.3.11. checkPointGetInfo – получение списка контрольных точек задач.....	52
2.7.3.12. getIsolatedGroups – получение списка изолированных групп задачи и их состава .....	53
2.7.3.13. setTaskMarker – назначение маркера для задачи.....	55
2.7.3.14. taskHide – скрывание задач.....	56
2.7.3.15. taskExamine – ознакомление пользователей с задачей.....	58
2.7.3.16. closetask – закрытие задачи.....	58
2.7.3.17. rejectTask – отправка задачи на доработку.....	59
2.7.3.18. taskReOpen – восстановление задачи из архива.....	60
2.7.3.19. getTasksSystemInfo - получение задач для iOS-приложения .....	60
2.7.3.20. getTasksWorkGroupInfo - получение рабочих групп задач для iOS-приложения.....	61
2.7.3.21. getTasksFiles - получение файлов задач для iOS-приложения .....	61
2.7.4. Функции для работы со списком задач.....	62
2.7.4.1. setNewInAction – установка/сброс признака «Новые действия» .....	62
2.7.4.2. setNewInTask – установка/сброс признака «Новая задача».....	63
2.7.4.3. resetNewInTask – сброс признаков обновления при открытии задачи.....	64
2.7.4.4. setNewActions – установка/сброс признака «Новые действия» в REST.....	64
2.7.5. Функции для работы с проектами.....	65
2.7.5.1. copyProject – копирование проекта со всеми его задачами.....	65
2.7.5.2. getProjectsInfo – получение информации о проекте.....	66
2.7.6. Функции для работы с документами.....	67
2.7.6.1. documentCreate – создание документа.....	67
2.7.6.2. getDocumentInfo – получение информации о карточке документа .....	69
2.7.6.3. getTasksAndDocData – получение информации о полях задачи и документа.....	71
2.7.6.4. getRoutetemplateNodes – получение информации о маршруте рассмотрения документа .....	74
2.7.6.5. documentConsider – рассмотрение документа .....	77
2.7.6.6. coordDocument – принудительная смена статуса документа на «Рассмотрен».....	78

2.7.6.7. sendDocumentOnReject – отправка документа на доработку.....	78
2.7.6.8. sendDocumentToArchive – отправка документа в архив.....	79
2.7.6.9. sendDocumentToArchiveManual – принудительная отправка документа в архив.....	80
2.7.6.10. createNewStageOfDocumentConsideration – создание нового этапа рассмотрения документа.....	81
2.7.6.11. deleteDocument – удаление документа.....	82
2.7.6.12. approveResolution - утверждение проектов резолюций.....	83
2.7.6.13. getDocumentParents - получение связанных документов.....	84
2.7.7. Функции для работы с библиотекой.....	85
2.7.7.1. addLibraryFolder – создание папки библиотеки.....	85
2.7.7.2. APIfolderSetRightsOnEmployer – назначение пользователю прав на папку библиотеки.....	85
2.7.7.3. renameLibraryFolder – переименование папки библиотеки.....	86
2.7.7.4. moveLibraryFolder – перемещение папки в библиотеке .....	87
2.7.7.5. deleteLibraryFolder – удаление папки библиотеки.....	88
2.7.7.6. addLibraryFile – добавление файла в библиотеку.....	88
2.7.7.7. copyLibraryFile – копирование файла из одной папки библиотеки в другую.....	89
2.7.7.8. moveLibraryFile – перемещение файлов библиотеки из одной папки в другую .....	90
2.7.7.9. deleteLibraryFile – удаление файла из библиотеки.....	91
2.7.8. Функции для работы со справочниками, типами задач и типами таблиц.....	91
2.7.8.1. catalogCreate – создание справочника/типа задачи.....	91
2.7.8.2. catalogRename – переименование справочника/типа задачи.....	93
2.7.8.3. catalogDelete – удаление справочника/типа задачи.....	94
2.7.8.4. taskTypeGetList – получение списка доступных пользователю типов задач.....	95
2.7.8.5. catalogGetList – получение списка доступных пользователю справочников.....	96
2.7.8.6. catalogAddField – добавление поля в справочник/тип задачи.....	97
2.7.8.7. catalogEditField – редактирование поля справочника/типа задачи.....	99
2.7.8.8. catalogDeleteField – удаление поля справочника/типа задачи.....	100
2.7.8.9. taskTypeGetFields – получение списка полей типа задачи.....	101
2.7.8.10. catalogAddRecord – добавление записи в справочник.....	103
2.7.8.11. catalogEditRecord – редактирование записи справочника.....	105
2.7.8.12. catalogDeleteRecord – удаление записи справочника.....	107

2.7.8.13. catalogGetRecord – получение записи справочника .....	108
2.7.8.14. getCatalogRecordsExt – получение списка (расширенного) записей справочника .....	109
2.7.9. Функции для работы со словарями.....	110
2.7.9.1. editDictionary – создание, редактирование и удаление словаря.....	110
2.7.9.2. editDictionaryWord – редактирование элемента словаря.....	111
2.7.10. Функции для работы со счётчиками документов/ задач.....	112
2.7.10.1. editDocCounter – редактирование счётчика документа.....	112
2.7.10.2. getDocCounter – получение информации о счётчике документа .....	113
2.7.11. Функции для работы с внешними пользователями/доменами.....	114
2.7.11.1. addOuterUser – добавление внешнего пользователя/домена в опрашиваемый ящик пользователя.....	114
2.7.11.2. editOuterUser – редактирование внешнего пользователя/домена.....	115
2.7.11.3. deleteOuterUser – удаление внешнего пользователя/домена .....	115
2.7.11.4. deleteMailbox – удаление почтового ящика пользователя .....	116
2.7.12. Функции для работы с папками задач .....	117
2.7.12.1. taskFolderCreate – создание папки задач.....	117
2.7.12.2. taskFolderRename – переименование папки задач.....	117
2.7.12.3. taskFolderDelete – удаление папки задач.....	118
2.7.12.4. getUserTaskFolders – получение списка папок задач пользователя.....	118
2.7.13. Функции для работы с напоминаниями.....	119
2.7.13.1. taskNoticeCreate – создание напоминания.....	119
2.7.13.2. taskNoticeView – ознакомление с напоминанием.....	123
2.7.13.3. taskNoticeDelete – удаление напоминания по его ID.....	124
2.7.14. Функции для работы с шаблонами задач.....	125
2.7.14.1. createTaskTemplate – создание шаблона задачи.....	125
2.7.14.2. editTaskTemplate – редактирование шаблона задачи.....	127
2.7.14.3. deleteTaskTemplate – удаление шаблона задачи.....	129
2.7.14.4. getTaskTemplate – получение информации о шаблоне задачи.....	130
2.7.15. Функции для работы с шаблонами изолированных рабочих групп.....	131
2.7.15.1. saveIWGTemplate – создание шаблона изолированной рабочей группы.....	131
2.7.15.2. getIWGForTemplate – получение списка изолированных групп в шаблоне.....	132

2.7.16. Функции для работы с шаблонами карточек документов.....	132
2.7.16.1. getDocumentTemplateList – получение списка шаблонов карточек документов.....	132
2.7.17. Функции для работы с полномочиями пользователей.....	133
2.7.17.1. setEmpPermission - назначение полномочий на пользователя.....	133
2.7.17.2. getEmpPermission - возврат полномочий на пользователя.....	134
2.7.17.3. getEmpPermissionMultiData - возврат полномочий на нескольких пользователей.....	135
2.7.18. API для делегирования полномочий заместителю.....	137
2.7.19. saveColorSettings – сохранение цветовых настроек пользователя.....	144
2.7.20. Создание объектов в системе «Мотив» на основе данных, вводимых на стороннем сайте.....	145
<b>3. Создание обработчиков.....</b>	<b>151</b>
3.1. Создание выполняемых файлов для обработчиков типа «Вычисление значений полей»... ..	159

## 1. Введение

**API (Application Programming Interface)** – набор функций, который программист может использовать для расширения возможностей системы «Мотив» (далее – Системы). API-функции позволяют осуществлять интеграцию Системы с другими приложениями. Функции написаны на языке разметки XML. Данное руководство предполагает наличие у программиста навыков работы с этим форматом. В руководстве описаны правила построения, вызова и получения результатов выполнения API-функций Системы.

## 2. Использование API-функций системы Мотив

### 2.1. Структура входного XML

Входной XML имеет следующую структуру:

1. **Заголовок** `<?xml version="1.0" encoding="UTF-8"?>` – Первая строка XML обязательно должна содержать заголовок документа, в котором определяется версия XML и кодировка, в которой будут представлены данные;
2. **Тег Документ** `<document>` – Корневой тег, остальные теги являются вложенными в него;
3. **Тег название функции** `<function>` – предназначен для описания функции. Имя функции указывается в качестве значения атрибута «**name**».
4. **Тег Параметры** `<param>` – описание параметров функции. Имя параметра указывается в качестве значения атрибута «**name**»;
5. **Тег Клиент** `<client>` – описание клиента. Имя клиента указывается в качестве значения атрибута «**name**», номер версии — атрибута «**version**»;

Пример структуры входного XML:

```
<document>
  <function name="имя функции">
    <param name="имя параметра">значение параметра</param>
  </function>
  <client name="имя клиента" version="номер версии"/>
</document>
```

Передаваемые функцией параметры указываются в тегах `<param>`, вложенных в тег `<function>`.

Название параметра задаётся в качестве значения атрибута «**name**» тега `<param>`. Во избежание проблем со специальными символами все строковые параметры следует передавать заключёнными в тег `<CDATA>`.

Пример записи для передачи строкового параметра:

```
<param name="название параметра"><![CDATA[строковое значение параметра]]></param>
```

Необязательные параметры можно передавать без указания значений. Например, необязательный параметр `parentId` в функции «**addTask**» передаётся следующим образом:

```
<param name="parentId"/>
```

Помимо параметров необходимо передавать информацию о программе-клиенте, использующей API. Для этого во входном файле XML в тег **<document>** должен быть вложен тег **<client>**. Тег **<client>** имеет два обязательных атрибута:

- **name** – имя программы-клиента;
- **version** – версия программы-клиента.

## 2.2. testConnection – тестирование соединения

Функция для тестирования соединения. Вызывайте ее первой, чтобы удостовериться, что соединение с сервером Системы установлено.

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="testConnection">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка).

При успешном соединении возвращается ID пользователя, логин и пароль которого были переданы, а также название компании, владеющей лицензией.

## 2.3. Вызов API-функции

Для аутентификации пользователя от имени которого выполняется функция во входном XML задайте два обязательных параметра: имя пользователя (**login**) и его пароль (**pass**), закодированный по алгоритму md5. Если эти параметры не указаны, то функция не будет выполнена и вернет сообщение "No authorization". По логину и паролю функция идентифицирует пользователя, проверит его права на запрашиваемую операцию и при их наличии вернет результат. Вызов API-функций осуществляется передачей в файл **functions.php** методом POST переменной **input\_xml**, содержащей сформированный xml. Файл **functions.php** доступен по адресу:

`http://HOST/api/functions.php`, где `HOST` – адрес Системы.

#### 2.4. Результат выполнения API-функции

Результат выполнения функции возвращается в формате xml (выходной XML), который имеет следующую структуру: главный тег `<document>` имеет вложенные теги `<info>` и `<result>`. В теге `<info>` передаётся служебная информация:

- `version` – текущая версия Системы;
- `api_version` – текущая версия API;
- `time` – текущее время на сервере;
- `execution_time` – время выполнения функции на сервере.

В теге `<result>` возвращаются запрашиваемые функцией данные. Он всегда имеет атрибут `value`, который может принимать одно из следующих значений:

`value=0`. Функция выполнена успешно.

`value=1`. Пользователя с переданным логином и паролем в Системе нет.

`value=2`. Заданы некорректные параметры. Под заданием некорректных параметров подразумевается передача параметров другого формата или превышение длины параметра. Например, передача строкового параметра вместо числового либо превышение допустимой длины строкового параметра.

`value=3`. В процессе выполнения SQL-запроса произошла ошибка. Текст ошибки выводится в значении тега `<result>`.

`value=4`. Ошибка в процессе выполнения. Текст ошибки выводится в значении тега `<result>`.

`value=5`. Нарушена структура XML. В возвращаемом XML будет указана подробная информация об ошибке в процессе парсинга XML.

`value=6`. Нет прав на выполнение требуемой операции. Например, передан ID удалённого или не существующего в Системе пользователя.

#### 2.5. Работа API-функций с файлами

Для работы с файлами, прикрепленными к объектам системы, используйте секцию `"files"`.

Пример фрагмента кода XML для взаимодействия с файлами:

```
<param name="files">  
<id>123</id>  
<tmp_name><![CDATA[C:/azxc.xml]]></tmp_name>
```

```
<name><![CDATA[azxc.xml]]></name>  
<description><![CDATA[Приложение]]></description>  
<idtask>314</idtask>  
<filesize>1852</filesize>  
</param>
```

Описание входных параметров:

**id** – ID файла в базе файлов (ATTACHMENT); является ключевым параметром, в случае копирования в библиотеку файла, прикрепленного к действию (целое положительное число);

**tmp\_name** – путь к файлу-источнику (строка);

**name** – название файла (строка);

**description** – описание файла (строка);

**idtask** – ID задачи (целое положительное число);

**filesize** – размер файла в байтах (целое положительное число).

## 2.6. Работа с Rest API

Дополнительно к основному API-вызову функций:

**http://HOST/api/functions.php**, где **HOST** – адрес Системы.

есть REST - интерфейс, позволяющий передавать параметры и данные в формате json:

**https://HOST/rest/<functionName>**

Общий алгоритм использования:

### 1. Авторизация

**https://<hostname>/rest/login?login=<login>&pass=<password>**

Успешная авторизация возвращает cookie **PHPSESSID**, этот параметр сессии нужно будет указывать в последующих запросах.

### 2. Вызов функций GET или POST запросами в формате

**https://<hostname>/rest/<functionName>**

Параметры можно передать как обычные переменные get - запроса (пример авторизации выше).

Также параметры api функции возможно передавать в теле запроса в формате объекта json вида (наименования параметров указаны в спецификации функции):

```
{  
  <parameter 1>: <value 1>,  
  <parameter 2>: <value 2>,  
  ...  
  <parameter N>: <value N>  
}
```

Для этого в заголовке запроса необходимо указать **Content-Type == 'application/json'**.  
Ответ будет возвращен в формате json.

### 3. Выход из системы

**https://<hostname>/rest/logout**

## 2.7. Описание параметров вызова API-функций

### 2.7.1. Функции для работы с подразделениями

#### 2.7.1.1. getpermissionsforemployer – получение списка пользователей подразделения

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>  
<document>  
  <function name="getpermissionsforemployer">  
    <param name="login">ivanov</param>  
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>  
    <param name="idDepartment">2037</param>  
    <param name="idEmployer">1567</param>  
  </function>  
  <client name="SomeProgram" version="1.0"/>  
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idDepartment** – ID подразделения (целое положительное число);

**idEmployer** – ID пользователя, для которого выполняется функция (целое положительное число).

**Замечание** – Получение информации о пользователях подразделения доступно при наличии на них полномочий «Полные права» или «Только свои задачи» у пользователя, для которого выполняется функция.

Пример выходного XML:

```
<code>
<?xml version="1.0" encoding="utf-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>11:59:49</time>
    <execution_time>0.046087980270386</execution_time>
  </info>
  <result value="0">
    <employer>
      <id>1</id>
      <name><![CDATA[Иванов Иван Иванович]]></name>
      <isboss>1</isboss>
      <manpost>Инженер-программист</manpost>
    </employer>
    <employer>
      <id>20178</id>
      <name><![CDATA[Петров Петр Петрович]]></name>
      <isboss>1</isboss>
      <manpost>Дизайнер</manpost>
    </employer>
  </result>
</document>
</code>
```

### 2.7.1.2. departmentMove – перемещение подразделения

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="departmentMove">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idEmployer">1000</param>
    <param name="iddepartmentmove">22</param>
    <param name="iddepartmentnewparent">10</param>
    <param name="issaveemployersrights">0</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который перемещает подразделение (целое положительное число);

**iddepartmentmove** – ID перемещаемого подразделения (целое положительное число);

**iddepartmentnewparent** – ID нового родительского подразделения для перемещаемого подразделения (целое положительное число);

**issaveemployersrights** – может принимать два значения: 1 – сохранять дополнительные полномочия пользователей; 0 – не сохранять дополнительные полномочия пользователей.

### 2.7.1.3. getDepartmentHead - получение руководителя подразделения

Возвращает руководителя подразделения пользователя.

Пример входного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <function name="getDepartmentHead">
```

```
<param name="login">ivanov</param>
<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="employer">1</param>
<param name="isogv">1</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**employer** - id пользователя, для которого нужно получить руководителя;

**isogv** - выводить руководителя органа власти (1- да; 0 - нет).

**Внимание!** Для работы функции при настройке подразделения должны быть добавлены поля:

- **Руководитель** (тип поля: сотрудник; идентификатор: HEAD).

- **Орган власти** (тип поля: логический; идентификатор: OGV).

#### 2.7.1.4. getEmployersOGV - получение органа власти пользователя

Возвращает орган власти пользователя.

Пример входного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <function name="getEmployersOGV">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="employer">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание параметров:

**login** – логин пользователя (строка);  
**pass** – пароль пользователя (строка);  
**employer** - id пользователя.

**Внимание!** Для работы функции при настройке подразделения должны быть добавлены поля:

- **Руководитель** (тип поля: сотрудник; идентификатор: HEAD).
- **Орган власти** (тип поля: логический; идентификатор: OGV).

## 2.7.2. Функции для работы с пользователями

### 2.7.2.1. userAdd – создание пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="userAdd">
    <param name="login">ivanov</param>
    <param name="pass">b3776e576ba4a84aee9b974e03741f59</param>
    <param name="IDEmployer">1</param>
    <param name="IDDepartment">140</param>
    <param name="user_login">mapic_user_256</param>
    <param name="user_pass">123</param>
    <param name="edition">complete</param>
    <param name="isBoss">1</param>
    <param name="addNumber"/>
    <param name="sortOrder"/>
    <param name="nameLast"><![CDATA[Фамилия]]></param>
    <param name="nameFirst"><![CDATA[Имя]]></param>
    <param name="nameMiddle"><![CDATA[Отчество]]></param>
    <param name="birthDate"><![CDATA[31.01.1955]]></param>
    <param name="gender"><![CDATA[Мужской]]></param>
    <param name="country"><![CDATA[Россия]]></param>
```

```

<param name="zipCode">308036</param>
<param name="area"><![CDATA[Белгородская]]></param>
<param name="city"><![CDATA[Белгород]]></param>
<param name="address"><![CDATA[Спортивная 234к]]></param>
<param name="phoneWork">380088</param>
<param name="phoneHome">514747</param>
<param name="phoneCell">89051700011</param>
<param name="email"><![CDATA[email@mail.mail]]></param>
<param name="company"><![CDATA[Мотив]]></param>
<param name="position"><![CDATA[программист]]></param>
<param name="dictrecords">
  <onedictvalue>
    <fieldid>65</fieldid>
    <values>
      <value><![CDATA[значение для поля 23]]></value>
    </values>
  </onedictvalue>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**IDEmployer** – ID пользователя, который выполняет функцию (целое положительное число);

**IDDepartment** – ID подразделения, в которое добавляется пользователь (целое положительное число);

**user\_login** – логин добавляемого пользователя (строка);

**user\_pass** – пароль добавляемого пользователя (строка);

**edition** – права пользователя на модули, может принимать следующие значения: complete; workflow; docflow;

**isBoss** – может принимать следующие значения: 0 – статус «Сотрудник»; 1 – статус «Начальник»; 2 – статус «Делопроизводитель»;

**addNumber** – дополнительный номер пользователя (целое положительное число);

**sortOrder** – номер пользователя при принудительной сортировке (целое положительное число);

**nameLast** – фамилия пользователя (строка);

**nameFirst** – имя пользователя (строка);

**nameMiddle** – отчество пользователя (строка);

**birthDate** – дата рождения пользователя (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**gender** – пол пользователя (строка);

**country** – страна (строка);

**zipCode** – индекс (строка);

**area** – область (строка);

**city** – город (строка);

**address** – адрес пользователя (строка);

**phoneWork** – рабочий телефон пользователя (строка);

**phoneHome** – домашний телефон пользователя (строка);

**phoneCell** – мобильный телефон пользователя (строка);

**email** – адрес электронной почты пользователя (строка);

**company** – название компании, в которой работает пользователь (строка);

**position** – должность пользователя (строка).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <info>
```

```
    <api_version>1.0</api_version>
```

```
    <time>12:35:00</time>
```

```
    <execution_time>6.0676908493</execution_time>
```

```
  </info>
```

```
<result value="0">
  <user_id>20427</user_id>
</result>
</document>
```

#### 2.7.2.2. userRemove – удаление пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="userRemove">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="iduser">2000</param>
    <param name="areplacegroup">
      <group>
        <id_group>0</id_group>
        <id_user_group>22412</id_user_group>
        <is_object>0</is_object>
      </group>
      <objects>
        <object>
          <object_id>123</object_id>
          <id_user>20001</id_user>
          <is_object>1</is_object>
        </object>
      </objects>
    </param>
    <param name="disablesystemactions">1</param>
  </function>
</document>
<client name="MotiwAutoTest" version="1.0"/>
```

</document>

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idUser** – ID удаляемого пользователя;

**disableSystemActions** – признак *Не уведомлять об изменении рабочей группы*.

Описание входных параметров:

**areplacegroup** – параметры перенаправления объектов

**id\_group** – целое, id группы,

**id\_user\_group** – целое, id пользователя, на которого перенаправляем всю группу объектов или -1, если группу нужно удалить

**is\_object** – 0/1, признак, группа (0) или объект (1), здесь = 0,

**objects** – список объектов для перенаправления

**object\_id** – целое, id объекта,

**id\_user** – целое, id пользователя, на которого перенаправляем объект или -1, если объект нужно удалить,

**is\_object** – 0/1, признак, группа (0) или объект (1), здесь = 1

2.7.2.3. `getEmployerID` – получение ID пользователя по его логину и паролю

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<document>
```

```
  <function name="getEmployerID">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```
  </function>
```

```
  <client name="SomeProgram" version="1.0" />
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

---

**pass** – пароль пользователя (строка).

Если пользователь с переданным логином и паролем в системе существует, то его ID возвращается в качестве значения тега **<id>**, вложенного в тег **<result>**.

Описание входных параметров:

**areplacegroup** – параметры перенаправления объектов

**id\_group** – целое, id группы,

**id\_user\_group** – целое, id пользователя, на которого перенаправляем всю группу объектов или -1, если группу нужно удалить

**is\_object** – 0/1, признак, группа (0) или объект (1), здесь = 0,

**objects** – список объектов для перенаправления

**object\_id** – целое, id объекта,

**id\_user** – целое, id пользователя, на которого перенаправляем объект или -1, если объект нужно удалить,

**is\_object** – 0/1, признак, группа (0) или объект (1), здесь = 1

2.7.2.4. **userGetInfo** – получение информации о пользователях из справочника «Адресная книга»

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="userGetInfo">
    <param name="login">ivanov</param>
    <param name="pass">b3776e576ba4a84aee9b974e03741f59</param>
    <param name="aUserIDs">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**aUserIDs** – массив ID пользователей, для которых требуется получить информацию (целое положительное число; через запятую).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>17:57:45</time>
    <execution_time>0.284126996994</execution_time>
  </info>
  <result value="0">
    <empls>
      <p1>
        <id>1</id>
        <name><![CDATA[admin]]></name>
        <id_department>1</id_department>
        <isdeleted>0</isdeleted>
        <isboss>1</isboss>
        <id_rec_dict>21</id_rec_dict>
      </p1>
    </empls>
    <cat_recs>
      <p1>
        <id>58</id>
        <id_mf>45</id_mf>
        <id_record>21</id_record>
        <value_char><![CDATA[admin]]></value_char>
        <isdeleted>0</isdeleted>
      </p1>
      <p2>
        <id>59</id>
        <id_mf>59</id_mf>
        <id_record>21</id_record>
        <value_char><![CDATA[alexandrova@motiw.ru]]></value_char>
      </p2>
    </cat_recs>
  </result>
</document>
```

```
<isdeleted>0</isdeleted>
</p2>
</cat_recs>
</result>
</document>
```

2.7.2.5. `getPicture` – получение изображения, прикреплённого к записи справочника

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getPicture">
    <param name="login">ivanov</param>
    <param name="pass">b3776e576ba4a84aee9b974e03741f59</param>
    <param name="idPicEmp">1325</param>
    <param name="checkSum">getinfo</param>
    <param name="saveAs">1.jpg</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idPicEmp** – ID пользователя, для которого требуется получить системное фото;

**checkSum** – контрольная сумма: если равна текущей, то файл не возвращается; если значением параметра является `getinfo`, то в любом случае будет получена информация о файле;

**saveAs** – новое название файла, которое будет передано в параметре **filename** заголовка **Content-Disposition** (если **false**, то будет передано текущее название).

Пример выходного XML (возвращается только в том случае, если файл не был изменён):

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
```

```

<info>
  <version>3.7</version>
  <api_version>1.0</api_version>
  <time>12:49:30</time>
  <execution_time>0.0551228523254</execution_time>
</info>
<result value="0">
  <id>119</id>
  <id_record>1325</id_record>
  <id_mf>50</id_mf>
  <filename><![CDATA[12-php5A3.jpg]]></filename>
  <isdeleted>0</isdeleted>
  <id_employer>1</id_employer>
  <displayfilename><![CDATA[php5A3.jpg]]></displayfilename>
  <url><![CDATA[Z:/_CVS/HEAD/Motiw/files/Dicts/motiw/1325/php5A3.jpg]]></url>
  <checksum><![CDATA[492b0bf1e30c43a61a8870c97fcae486]]></checksum>
</result>
</document>

```

**Замечание** – Получение выходного XML недоступно, если файл был изменён.

2.7.2.6. getEmployerDepartmentUserTree – получение информации о пользователях и их полномочиях

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="getEmployerDepartmentUserTree">
    <param name="login">admin</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="iddepartment">7</param>
    <param name="onlydepartments">1</param>
  </function>
<client name="MApiC" version="1.0" />

```

</document>

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**onlydepartments** – только подразделения;

**iddepartment** – ID подразделения (целое положительное число).

Если параметр **onlydepartments** не передан или равен 0, функция возвращает информацию о сотрудниках и подразделениях, иначе – только дерево подразделений.

Если параметр **iddepartment** не передан, функция возвращает информацию обо всем дереве подразделений со всеми доступными пользователями. Если **iddepartment = 0**, функция возвращает информацию о корневых подразделениях, если **iddepartment > 0** – обо всех дочерних подразделениях и пользователях указанного подразделения.

### 2.7.3. Функции для работы с задачами

#### 2.7.3.1. addTask – создание задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="AddTask">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```
    <param name="taskName"><![CDATA[Забрать принтер]]></param>
```

```
    <param name="idAuthor">409</param>
```

```
    <param name="authorsids">
```

```
      <id>4</id>
```

```
      <id>5</id>
```

```
      <id>6</id>
```

```
    </param>
```

```
    <param name="respPersonIds">
```

```
      <id>4</id>
```

```
      <id>5</id>
```

```
<id>6</id>
</param>
<param name="idTaskInitiator">409</param>
<param name="idRealAuthor">409</param>
<param name="groupIds">
  <id>1</id>
  <id>2</id>
  <id>3</id>
</param>
<param name="controllerIds">
  <id>4</id>
  <id>5</id>
  <id>6</id>
</param>
<param name="endDate">23.08.2018</param>
<param name="isWithReport">1</param>
<param name="description"><![CDATA[Описание задачи]]></param>
<param name="isSecret">0</param>
<param name="parentId" />
<param name="priority">0</param>
<param name="idDocumentExec" />
<param name="isOnlyForView">0</param>
<param name="isUseParentDocs">0</param>
<param name="idProject">14</param>
<param name="newProjectName" />
<param name="newProjectDesc" />
<param name="newProjectDesc2" />
<param name="newProjectEnddate">30.12.2020</param>
<param name="newProjectIdDepartment">0</param>
```

```

<param name="receive_sms">0</param>
<param name="receive_mail">1</param>
<files>
<file>
  <displayname><![CDATA[appstar2.ani]]></displayname>
  <description><![CDATA[Описание файла1]]></description>
</file>
<file>
  <displayname><![CDATA[appstar3.ani]]></displayname>
  <description><![CDATA[Описание файла2]]></description>
</file>
</files>
</function>
<client name="SomeProgram" version="1.0" />
</document>

```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**taskName** – название задачи (строка);

**idAuthor** – ID автора задачи (целое положительное число);

**idRealAuthor** – ID постановщика задачи (целое положительное число);

**authorsids** – массив ID авторов задачи (целое положительное число);

**respPersonIds** – массив ID ответственных руководителей задачи (целое положительное число);

**groupIds** – массив ID исполнителей задачи (целое положительное число);

**controllerIds** – массив ID контролёров задачи (целое положительное число);

**endDate** – дата окончания задачи (строка);

**isWithReport** – задача с докладом; может принимать два значения: 1 – с докладом; 0 – без доклада;

**description** – описание задачи (строка);

**isSecret** – признак *Секретная задача*; может принимать два значения: 1 – секретная задача; 0 – обычная задача;

**parentId** – ID родительской задачи (целое положительное число);

**priority** – приоритет задачи; может принимать два значения: 1 – важная задача; 0 – обычная задача;

**idDocumentExec** – ID документа по исполнению (целое положительное число);

**isOnlyForView** – признак *Только для ознакомления*; может принимать два значения: 1 – задача с признаком *Только для ознакомления*; 0 – обычная задача;

**isUseParentDocs** – признак наследования документов; может принимать два значения: 1 – задача с признаком наследования документов; 0 – обычная задача;

**idProject** – ID проекта, к которому прикрепляется задача (целое число);

**newProjectName** – имя нового проекта (строка);

**newProjectDesc** – описание нового проекта (строка);

**newProjectDesc2** – заказчик нового проекта (строка);

**newProjectEnddate** – дата окончания нового проекта (строка);

**newProjectIdDepartment** – ID подразделения, к которому относится новый проект (целое положительное число).

Прикрепление задачи к проекту.

Если при постановке задача прикрепляется к новому проекту, то значение параметра **idProject** должно быть отрицательным. Если значение этого параметра равно нулю либо данный тег отсутствует или является пустым, то задача прикрепляется к проекту «Задачи вне проектов». Для прикрепления задачи к определённом проекту в параметре **idProject** следует указать значение его ID.

**receive\_sms** – получать обновления задачи через SMS (необязательный параметр)

**receive\_mail** – получать обновления задачи по почте (необязательный параметр)

Прикрепление к задаче файлов.

Если к задаче нужно прикрепить файлы, то информация о них указывается в теге **<files>**, вложенному в тег **<function>**. Каждому файлу соответствует свой тег **<file>**.

В тег **<file>** должны быть вложены теги **<displayname>** и **<description>**. В теге **<displayname>** передаётся название файла, которое будет отображаться в задаче, в теге **<description>** – его описание. Передача файлов осуществляется отдельно.

Пример постановки задачи с прикреплением 2-х файлов с помощью библиотеки curl:

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $sTaskName = trim($_POST["taskname"]);
    if (!empty($sTaskName)) {
        $url="http://motiw.local/api/functions.php";
        $sch = curl_init();

        $aPostData = array();

        // Изначальная структура XML
        $dom = new DOMDocument("1.0", "UTF-8");
        $document = $dom->createElement('document');
        $function = $dom->createElement('function');
        $client = $dom->createElement('client');

        // Инициализация названия функции и клиента
        $function->setAttribute('name', "addTask");
        $client->setAttribute('name', "test");
        $client->setAttribute('version', "0.1");

        // Поле login
        $param = $dom->createElement("param");
        $param->setAttribute("name", "login");
        $param->nodeValue = "admin";
        $function->appendChild($param);

        // Поле pass
        $param = $dom->createElement("param");
        $param->setAttribute("name", "pass");
        $param->nodeValue = md5("admin");
        $function->appendChild($param);

        // Поле idAuthor
        $param = $dom->createElement("param");
        $param->setAttribute("name", "idAuthor");
        $param->nodeValue = 1;
        $function->appendChild($param);

        // Поле idRealAuthor
        $param = $dom->createElement("param");
        $param->setAttribute("name", "idRealAuthor");
        $param->nodeValue = 1;
        $function->appendChild($param);

        // Поле idTaskInitiator
        $param = $dom->createElement("param");
        $param->setAttribute("name", "idTaskInitiator");
        $param->nodeValue = 1;
        $function->appendChild($param);

        // Поле taskname
        $param = $dom->createElement("param");
        $param->setAttribute("name", "taskName");

```

```

$param->nodeValue = $sTaskName;
$function->appendChild($param);

// Секция files - прикрепляемые файлы
$files = $dom->createElement("files");

// file1 - первый файл
$file = $dom->createElement("file");

// Название файла
$node = $dom->createElement("displayname");
$node->nodeValue = "file1.txt";
$file->appendChild($node);

// Описание (Отображаемое имя)
$node = $dom->createElement("description");
$node->nodeValue = "File One";
$file->appendChild($node);
$files->appendChild($file);

// file2 - второй файл
$file = $dom->createElement("file");

// Название файла
$node = $dom->createElement("displayname");
$node->nodeValue = "file2.txt";
$file->appendChild($node);

// Описание (Отображаемое имя)
$node = $dom->createElement("description");
$node->nodeValue = "File Two";
$file->appendChild($node);
$files->appendChild($file);

// Добавляем файлы в функцию
$function->appendChild($files);

$sFilePath1 = "file1.txt"; // Файл, описание которого расположено в 1-м те
$sFilePath2 = "file2.txt"; // Файл, описание которого расположено в 2-м те

// Содержимое файлов
file_put_contents($sFilePath1, "Hello, World! From file one!");
file_put_contents($sFilePath2, "Hello, World! From file two!");

curl_setopt($ch, CURLOPT_SAFE_UPLOAD, FALSE); // php 5.6+

// Добавляем файлы в запрос
$aPostData['file1'] = '@' . realpath($sFilePath1);
$aPostData['file2'] = '@' . realpath($sFilePath2);

// Формируем конечный XML
$document->appendChild($function);
$document->appendChild($client);

```

```

$dom->appendChild($document);
$dom->formatOutput = TRUE;
$aPostData['input_xml'] = $dom->saveXML(); // В элементе массива $aPostData

// Настройки cURL
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $aPostData);

// Выполняем
$store = curl_exec($ch);
curl_close ($ch);

if ($store) {
    echo "Задача: {$sTaskName} добавлена!";
}
}
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta charset="UTF-8">
    <title>Add task</title>
</head>
<body>
    <form name="form1" method="post" action="<?php echo $_SERVER['PHP_SELF'] ?>" ENCTY
        <label>Task name: <input type="text" name="taskname"></label>
        <input type="submit">
    </form>
</body>
</html>

```

### 2.7.3.2. addTaskByTemplate – создание задачи по шаблону

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
    <function name="addTaskByTemplate">
        <param name="login">admin</param>
        <param name="pass">21232f297a57a5a743894a0e4a801fc3</param>
        <param name="idEmployer">1</param>
        <param name="idTemplate">258</param>
        <param name="parentTask">14948</param>
        <param name="idDocument">2454</param>
    </function>

```

```
<param name="rule" type="array_assoc">
  <string>nazv</string>
</param>
<param name="add_task_params" type="array_assoc">
  <is_isolated>1</is_isolated>
  <iwg_name>iwg task name</iwg_name>
  <task_type_id>612</task_type_id>
  <task_name>taskk name from param</task_name>
  <project_id>11</project_id>
  <save_sys_actions_to_parent>1</save_sys_actions_to_parent>
  <authors type="array">
    <id>20002</id>
    <id>20005</id>
  </authors>
  <controllers>
    <id>20003</id>
    <id>20006</id>
  </controllers>
  <resppersons>
    <id>20005</id>
    <id>20006</id>
  </resppersons>
  <workers>
    <id>20454</id>
    <id>20417</id>
  </workers>
  <description>task desc from param</description>
  <workers_from_doc type="array">
    <id>20064</id>
    <id>20067</id>
  </workers_from_doc>
```

```
<isNeedCheckForSubstitution>1</isNeedCheckForSubstitution>
<parentroutetreeid>9654</parentroutetreeid>
<iddocumentexec>2455</iddocumentexec>
<document_author>20418</document_author>
<duration>15</duration>
<period_id>3</period_id>
<route_description>descr for route</route_description>
<field1 type="array_assoc">
  <name>name</name>
  <value>task name from field</value>
</field1>
<field2 type="array_assoc">
  <name>start_date</name>
  <value>11.11.2018 11:11:11</value>
</field2>
<field3 type="array_assoc">
  <name>end_date</name>
  <value>12.12.2018 12:12:12</value>
</field3>
<field4 type="array_assoc">
  <name>project</name>
  <value>test project name</value>
</field4>
<field5 type="array_assoc">
  <name>description</name>
  <value>task descr from field</value>
</field5>
<field6 type="array_assoc">
  <name>authors</name>
  <value type="array">
    <id>20015</id>
```

```
<id>20016</id>
</value>
</field6>
<field7 type="array_assoc">
  <name>controllers</name>
  <value type="array">
    <id>20009</id>
    <id>20008</id>
  </value>
</field7>
<field8 type="array_assoc">
  <name>resppersons</name>
  <value type="array">
    <id>20020</id>
    <id>20021</id>
  </value>
</field8>
<field9 type="array_assoc">
  <name>workers</name>
  <value type="array">
    <id>20022</id>
    <id>20024</id>
  </value>
</field9>
<field10 type="array_assoc">
  <name>2558</name> <!-- Пример строкового значения -->
  <value>string data</value>
</field10>
<field11 type="array_assoc">
  <name>2559</name> <!-- Пример целого значения -->
  <value>3</value>
```

```
</field11>
<field12 type="array_assoc">
  <name>2560</name> <!-- Пример вещественного значения -->
  <value>3.53</value>
</field12>
<field13 type="array_assoc">
  <name>2561</name> <!-- Пример значения типа дата -->
  <value>10.10.2018 10:10:10</value>
</field13>
<field14 type="array_assoc">
  <name>2562</name> <!-- Пример значения типа ссылка на справочник -->
  <value type="array">
    <text>Запись 1</text>
    <text>Запись 2</text>
  </value>
</field14>
<field15 type="array_assoc">
  <name>2563</name> <!-- Пример логического значения -->
  <value>да</value>
</field15>
<field16 type="array_assoc">
  <name>2564</name> <!-- Пример значения типа ссылка на библиотеку -->
  <value>dummy.txt</value>
</field16>
<field17 type="array_assoc">
  <name>2565</name> <!-- Пример значения типа ссылка на задачу -->
  <value>test_task_name</value>
</field17>
</param>
<param name="ignorenecessaryfields">1</param>
<fields>
```

```
<field>
  <name>name</name>
  <value>task name from field</value>
</field>
<field>
  <name>start_date</name>
  <value>11.11.2018 11:11:11</value>
</field>
<field>
  <name>end_date</name>
  <value>12.12.2019 12:12:12</value>
</field>
<field>
  <name>project</name>
  <value>test project name</value>
</field>
<field>
  <name>description</name>
  <value>task descr from field</value>
</field>
<field>
  <name>2558</name> <!-- Пример строкового значения -->
  <value>string data</value>
</field>
<field>
  <name>2559</name> <!-- Пример целого значения -->
  <value>3</value>
</field>
<field>
  <name>2560</name> <!-- Пример вещественного значения -->
  <value>3.53</value>
```

```

</field>
<field>
  <name>2561</name> <!-- Пример значения типа дата -->
  <value>10.10.2018 10:10:10</value>
</field>
<field>
  <name>2563</name> <!-- Пример логического значения -->
  <value>да</value>
</field>
<field>
  <name>2564</name> <!-- Пример значения типа ссылка на библиотеку -->
  <value>any.txt</value>
</field>
<field>
  <name>2565</name> <!-- Пример значения поля типа ссылка на задачу -->
  <value>test_task_name</value>
</field>
</fields>
<files>
  <file>
    <name>pic.png</name>
    <displayname>picture</displayname>
    <description>picture description</description>
    <signature></signature>
  </file>
</files>
</function>
<client name="some_program" version="1.0"/>
</document>

```

Описание параметров:

**login** – логин пользователя, от имени которого выполняется функция (строка);  
**pass** – пароль пользователя (строка);  
**idEmployer** – ID пользователя;  
**idTemplate** – ID шаблона задачи в системе;  
**parentTask** – ID родительской задачи;  
**idDocument** – ID документа по которому будет создана задача;  
**rule** – правило копирования полей в задачу из документа, задается в формате <ПОЛЕ ЗАДАЧИ>ПОЛЕ ДОКУМЕНТА</ПОЛЕ ЗАДАЧИ>, поля указываются идентификаторами;  
**add\_task\_params** – дополнительные параметры задачи, если задана секция **add\_task\_params**, то секция **fields** игнорируется, иначе – берется в запрос;  
**ignorenecessaryfields** – игнорировать ли заполненность обязательных полей задачи при создании.

Параметры секции **add\_task\_params**:

**is\_isolated** – признак задачи ИРГ;  
**iwg\_name** – название ИРГ;  
**task\_type\_id** – ID типа задачи ИРГ;  
**task\_name** – название задачи;  
**project\_id** – ID проекта;  
**save\_sys\_actions\_to\_parent** – отсылать ли системные действия в родительскую для ИРГ задачу;  
**authors** – ID авторов для задачи ИРГ;  
**controllers** – ID контроллеров для задачи ИРГ;  
**resppersons** – ID ответственных руководителей для задачи ИРГ;  
**workers** – ID исполнителей для задачи ИРГ;  
**description** – описание задачи;  
**workers\_from\_doc** – исполнители документа;  
**isNeedCheckForSubstitution** – выполнять ли пересчет заместителей;  
**parentroutetreeid** – ID родительского узла маршрута документа, для которого создается задача;  
**iddocumentexes** – ID документа, по исполнению которого создается задача;  
**document\_author** – ID автора документа;

**duration** – длительность;  
**period\_id** – ID периода длительности;  
**route\_description** – описание для узла маршрута с задачей.

Поля задачи передаются в виде:

```
<field type="array_assoc">  
  <name>поле</name>  
  <value>значение</value>  
</field>
```

В параметре **name** указывается:

- для системных полей – текстовый идентификатор поля (например, **STARTDATE**);
- для пользовательских полей – числовой идентификатор поля (например, 12345).

Идентификаторы системных полей:

**NAME** – название задачи, если не задано в расширенных настройках;

**START\_DATE** – дата начала задачи;

**END\_DATE** – дата окончания задачи;

**PROJECT** – проект задачи (название проекта), значение используется даже если проект указан в **add\_task\_params**;

**DESCRIPTION** – описание задачи, значение используется, если описание не задано в **add\_task\_params**;

**AUTHORS** – авторы задачи, если задача не ИРГ;

**CONTROLLERS** – контроллеры задачи, если задача не ИРГ;

**RESPPERSONS** – ответственные руководители задачи, если задача не ИРГ;

**WORKERS** – исполнители задачи, если задача не ИРГ.

Параметры секции **files**:

**name** – имя файла;

**displayname** – отображаемое имя файла;

**description** – описание файла;

**signature** – подпись файла.

### 2.7.3.3. editTask – редактирование задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="editTask">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">105674</param>
    <param name="taskName"><![CDATA[Позвонить поставщикам]]></param>
    <param name="authorsIds">
      <del>
        <id>3</id>
        <id>4</id>
      </del>
      <add>
        <id>206</id>
        <id>55</id>
      </add>
    </param>
    <param name="endDate">null</param>
    <param name="isWithReport">0</param>
    <param name="description"><![CDATA[Выяснить насчет поставок]]></param>
    <param name="priority">1</param>
    <param name="resppersonsIds">
      <del>
        <id>3</id>
        <id>4</id>
      </del>
      <add>
        <id>206</id>
        <id>55</id>
      </add>
    </param>
  </function>
</document>
```

```
</add>
</param>
<param name="groupIds">
  <del>
    <id>3</id>
    <id>4</id>
  </del>
  <add>
    <id>206</id>
    <id>55</id>
  </add>
</param>
<param name="controllerIds">
  <del>
    <id>3</id>
    <id>4</id>
  </del>
  <add>
    <id>206</id>
    <id>55</id>
  </add>
</param>
<param name="idProject">-1</param>
<param name="newProjectName"><![CDATA[Новый проект]]></param>
<param name="newProjectDesc" />
<param name="newProjectDesc2" />
<param name="newProjectEnddate">31.12.2018</param>
<param name="newProjectIdDepartment">95</param>
</function>
<client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID редактируемой задачи (целое положительное число);

**taskName** – название редактируемой задачи (строка);

**authorsids** – ID авторов задачи (целое положительное число);

**endDate** – дата окончания задачи (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**isWithReport** – задача *С докладом*; может принимать два значения: 1 – с докладом; 0 – без доклада;

**description** – описание задачи (строка);

**priority** – приоритет задачи; может принимать два значения: 1 – важная задача; 0 – обычная задача;

**respPersonIds** – ID ответственных руководителей задачи (целое положительное число);

**groupIds** – ID исполнителей задачи (целое положительное число);

**controllerIds** – ID контролёров задачи (целое положительное число);

**idProject** – ID проекта задачи (целое число);

**newProjectName** – название нового проекта (строка);

**newProjectDesc** – описание нового проекта (строка);

**newProjectDesc2** – заказчик нового проекта (строка);

**newProjectEnddate** – дата окончания нового проекта (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**newProjectIdDepartment** – подразделение, к которому относится новый проект (целое положительное число).

**Замечание** – Обязательными параметрами при передаче являются **login**, **pass**, **idTask**. Во входном XML передаются только те параметры задачи, которые необходимо отредактировать.

2.7.3.4. addAction – добавление действия в задачу

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="addaction">
    <param name="login">ivanov</param>
```

```

<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="idTask">106195</param>
<param name="idEmployer">1337</param>
<param name="message"><![CDATA[Ознакомлен]]></param>
<param name="alarm">0</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи, в которую добавляется действие (целое положительное число);

**idEmployer** – ID пользователя, от имени которого добавляется действие (целое положительное число);

**message** – текст действия (строка);

**alarm** – тип действия; может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение; 3 – системное действие.

В результате выполнения функции в теге **<id>**, вложенном в тег **<result>**, возвращается значение ID добавленного действия.

2.7.3.5. editAction – редактирование действия

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="editAction">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idAction">887294</param>
    <param name="message"><![CDATA[Сделаю послезавтра]]></param>
    <param name="alarm">0</param>
  </function>
<client name="SomeProgram" version="1.0"/>

```

**</document>**

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idAction** – ID редактируемого действия (целое положительное число).

**message** – новый текст действия (строка);

**alarm** – тип действия; может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение.

2.7.3.6. **gettaskbuttons** – получение списка действий по задаче

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="gettaskbuttons">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```
    <param name="idtask">2037</param>
```

```
    <param name="idEmployer">1567</param>
```

```
  </function>
```

```
  <client name="SomeProgram" version="1.0"/>
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи (целое положительное число);

**idEmployer** – ID пользователя (целое положительное число).

В результате выполнения данной функции возвращается название операции и ID действия.

ID действий:

1. Сохранить;
2. Завершить выполнение;
3. Вернуть задачу на доработку;

4. Согласовать документ;
  5. Отправить документ в архив;
  6. Переслать на исполнение;
  7. Вернуть документ на доработку;
  8. Сохранить от своего имени;
  9. Ознакомиться (в задачах по ознакомлению).
  10. Согласовать с замечаниями;
  11. Отправить доклад;
  12. Выставить оценку задачи;
  13. Отказаться согласовать.
- 2.7.3.7. `getTaskActions` – страничный вывод действий по задаче

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getTaskActions">
    <param name="login">ivanov</param>
    <param name="pass">b3776e576ba4a84aee9b974e03741f59</param>
    <param name="idTask">1544</param>
    <param name="pageNum">1</param>
    <param name="pageSize">3</param>
    <param name="countSys">1</param>
    <param name="idLastAct"></param>
    <param name="aActionIDs">1,2,3,4,5</param>
    <param name="sort">DESC</param>
    <param name="sDate">20.04.2018 18:00:00</param>
    <param name="eDate">20.05.2018 18:00:00</param>
    <param name="filter"><![CDATA[Ознакомлен]]></param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи, список действий которой требуется получить (целое положительное число);

**pageNum** – номер страницы (целое положительное число);

**pageSize** – количество действий, отображаемых на странице (целое положительное число);

**countSys** – может принимать два значения: 1 – учитывать системные действия; 0 – не учитывать системные действия;

**idLastAct** – ID действия, для которого в любом случае будет получена информация; данный параметр требуется для сравнения последнего действия в локальной базе клиента (целое положительное число);

**aActionIDs** – ID действий, которые не нужно возвращать (целое положительное число);

**sort** – флаг сортировки действий по дате регистрации; может принимать два значения: ASC – от более поздних к новым действиям; DESC – от новых к более поздним действиям.

**\$sDate** – дата начала интервала времени, для которого нужны действия

**\$eDate** – дата окончания интервала времени, для которого нужны действия

**\$filter** – строка поиска.

Если не нужно выводить действия постранично, опустите параметры **pageNum** и **pageSize** или установите их значение равным -1.

Если не нужно учитывать временной интервал, опустите параметры **sDate** и **eDate**.

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <info>
```

```
    <api_version>1.0</api_version>
```

```
    <time>10:31:08</time>
```

```
    <execution_time>0.451254844666</execution_time>
```

```
  </info>
```

```
  <result value="0">
```

```
    <actions>
```

```
      <p1>
```

```
<id>51828</id>
<id_task>1544</id_task>
<id_employer>1</id_employer>
<name><![CDATA[Сделал фото местности]]></name>
<adate><![CDATA[28.04.2018 18:02:58]]></adate>
<class>0</class>
<isfirst>1</isfirst>
<examined>0</examined>
<is_resolution>0</is_resolution>
<lastchange><![CDATA[28.04.2018 18:02:58]]></lastchange>
<rejected>0</rejected>
<num_in_task>1</num_in_task>
<means_num_in_task>1</means_num_in_task>
</p1>
</actions>
<attachments>
  <p1>
    <id>198</id>
    <id_task>1544</id_task>
    <filename><![CDATA[17-image001.tif]]></filename>
    <isdeleted>0</isdeleted>
    <id_employer>1</id_employer>
    <id_action>51828</id_action>
    <displayfilename><![CDATA[image001.tif]]></displayfilename>
    <filesize_bytes><![CDATA[1097170]]></filesize_bytes>
    <add_date><![CDATA[28.04.2018 18:02:58]]></add_date>
  </p1>
</attachments>
</result>
</document>
```

**Замечание** – Данная функция выводит список действий с прикрепленными к ним файлами с помощью функции «**getTaskAttachments**».

2.7.3.8. **getTaskAttachments** – получение списка прикрепленных к действиям файлов

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getTaskAttachments">
    <param name="login">ivanov</param>
    <param name="pass">b3776e576ba4a84aee9b974e03741f59</param>
    <param name="idTask">1</param>
    <param name="flag">inarray</param>
    <param name="aActionIDs">1,2,3</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя (целое положительное число);

**idTask** – ID задачи (целое положительное число);

**aActionIDs** – массив ID действий, для которых требуется получить информацию о файлах (или которые нужно исключить) (целое положительное число; через запятую);

**flag** – может принимать следующие значения:

**DESCRIPTION** – информация о файлах, прикрепленных к описанию задачи (**aActionIDs** не учитывается);

**ALL** – информация обо всех файлах в задаче (**aActionIDs** не учитывается);

**INARRAY** – информация о файлах, прикрепленных к действиям, ID которых указаны в **aActionIDs**;

**NOTINARRAY** – информация о файлах, прикрепленных к действиям, за исключением тех, ID которых указаны в **aActionIDs**.

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>12:06:29</time>
    <execution_time>0.247888803482</execution_time>
  </info>
  <result value="0">
    <attachments>
      <p1>
        <id>1</id>
        <id_task>1</id_task>
        <filename><![CDATA[1-Rukovodstvo polzovatelya.pdf]]></filename>
        <isdeleted>0</isdeleted>
        <id_employer>20013</id_employer>
        <id_action>2</id_action>
        <description><![CDATA[Руководство пользователя]]></description>
        <displayfilename><![CDATA[Rukovodstvo polzovatelya.pdf]]></displayfilename>
        <filesize_bytes><![CDATA[0]]></filesize_bytes>
        <add_date><![CDATA[25.08.2018 14:31:14]]></add_date>
      </p1>
    </attachments>
  </result>
</document>
```

**Замечание** – Если задача создана по документу, то возвращается информация о файле последней версии.

2.7.3.9. MakeSubtask – перевод задач в подзадачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
```

```

<function name="makesubtask">
  <param name="login">ivanov</param>
  <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
  <param name="idTask">1000</param>
  <param name="idEmployer">200</param>
  <param name="subtaskids">
    <id>10</id>
    <id>11</id>
    <id>12</id>
  </param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID родительской задачи для задач, ID которых указаны в параметре **subtaskids** (целое положительное число);

**idEmployer** – ID пользователя, который переводит задачи в подзадачи (целое положительное число);

**subtaskids** – массив ID задач, которые станут подзадачами задачи **idTask** (целое положительное число).

2.7.3.10. **checkpointCreate** – создание контрольной точки

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="checkpointcreate">
    <param name="idTask">2000</param>
    <param name="idEmployer">1000</param>
    <param name="checkpoint">
      <checkpoint>

```

```

<date>
  <![CDATA[19.06.2018 00:00]]>
</date>
<description>
  <![CDATA[Позвонить директору]]>
</description>
<reminds>
  <remind days="0">
    <system>1</system>
    <mail>1</mail>
    <sms>0</sms>
  </remind>
</reminds>
</checkpoint>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**idTask** – ID задачи, в которой создаётся контрольная точка (целое положительное число);

**idEmployer** – ID пользователя, который создаёт контрольную точку (целое положительное число);

**date** – дата наступления контрольной точки (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**description** – описание контрольной точки (строка);

**days** – количество дней, за которое должно прийти напоминание о наступлении контрольной точки; если **days=0**, то напоминание отправляется в момент наступления контрольной точки (целое положительное число);

**system** – отправка напоминания о наступлении контрольной точки сообщением системы «Мотив»; может принимать два значения: 1 – отправлять; 0 – не отправлять;

**mail** – отправка напоминания о наступлении контрольной точки электронной почтой; может принимать два значения: 1 – отправлять; 0 – не отправлять;

sms – отправка напоминания о наступлении контрольной точки через SMS; может принимать два значения: 1 – отправлять; 0 – не отправлять.

2.7.3.11. checkPointGetInfo – получение списка контрольных точек задач

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="checkPointGetInfo">
    <param name="login">ivanov</param>
    <param name="pass">b3776e576ba4a84aee9b974e03741f59</param>
    <param name="aTaskIDs">1464</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

login – логин пользователя (строка);

pass – пароль пользователя (строка);

aTaskIDs – массив ID задач, список контрольных точек которых требуется получить (целое положительное число, через запятую).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>17:28:52</time>
    <execution_time>0.204065084457</execution_time>
  </info>
  <result value="0">
    <checkpoints>
      <p1>
        <id>624</id>
        <enddate><![CDATA[26.04.2019 14:39:00]]</enddate>
```

```

    <description><![CDATA[описание КТ1]]></description>
    <state>0</state>
    <id_task>1464</id_task>
    <name><![CDATA[название КТ1]]></name>
    <id_cp_tpl>115</id_cp_tpl>
</p1>
...
<p4>
    <id>627</id>
    <enddate><![CDATA[05.05.2019 14:39:00]]></enddate>
    <description><![CDATA[описание КТ4]]></description>
    <state>0</state>
    <id_task>1464</id_task>
    <name><![CDATA[название КТ4]]></name>
    <id_cp_tpl>120</id_cp_tpl>
</p4>
</checkpoints>
</result>
</document>

```

2.7.3.12. getIsolatedGroups – получение списка изолированных групп задачи и их состава

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
    <function name="getIsolatedGroups">
        <param name="login">ivanov</param>
        <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
        <param name="idTask">2000</param>
    </function>
    <client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи, список изолированных групп которой требуется получить (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <info>
```

```
    <api_version>1.0</api_version>
```

```
    <time>11:18:58</time>
```

```
    <execution_time>0.0264701843262</execution_time>
```

```
  </info>
```

```
  <result value="0">
```

```
    <groups>
```

```
      <isolatedworkgroup id="156">
```

```
        <state>0</state>
```

```
        <isreported>0</isreported>
```

```
        <isuseparentdocs>1</isuseparentdocs>
```

```
        <is_post_sysactions_to_parent>1</is_post_sysactions_to_parent>
```

```
      <resppersons>
```

```
        <employer>
```

```
          <id>21609</id>
```

```
          <name>Иванов Иван Иванович</name>
```

```
          <dep_name>Главное подразделение</dep_name>
```

```
          <hashid>ad3e44031c3c17ff94c2cf2eeb760f42</hashid>
```

```
        </employer>
```

```
      </resppersons>
```

```
    <employers>
```

```
      <employer>
```

```
        <id>21005</id>
```

```
        <name>Петров Петр Петрович</name>
```

```

    <dep_name>Главное подразделение</dep_name>
    <hashid>ad3e44031c3c17ff94c2cf2eeb760f42</hashid>
  </employer>
</employers>
</isolatedworkgroup>
</groups>
</result>
</document>

```

Описание выходных параметров:

**state** – признак состояния задачи; может принимать два значения: 1 – задача закрыта; 0 – задача открыта;

**isreported** – признак *Доклад отправлен*; может принимать два значения: 1 – доклад отправлен; 0 – доклад не отправлен;

**isuseparentdocs** – признак доступности документа из родительской задачи; может принимать два значения: 1 – документы доступны; 0 – документы не доступны;

**is\_post\_sysactions\_to\_parent** – может принимать два значения: 1 – отображать системные действия в родительской задаче; 0 – не отображать системные действия в родительской задаче.

2.7.3.13. setTaskMarker – назначение маркера для задачи

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="setTaskMarker">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">50</param>
    <param name="idEmployer">106195</param>
    <param name="color">4</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи (целое положительное число);

**idEmployer** – ID пользователя, который назначает маркер для задачи (целое положительное число);

**color** – цвет маркера (строка).

Сброс маркера для задачи осуществляется, если параметр **color=0**.

#### 2.7.3.14. taskHide – скрытие задач

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="taskHide">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```
    <param name="idTASK">106195</param>
```

```
    <param name="IdEmployer">1337</param>
```

```
    <param name="date">20.09.2018 10:00:00</param>
```

```
    <param name="isWithRemind">1</param>
```

```
    <param name="createdate">10.10.2018 22:42:43</param>
```

```
    <param name="showdate">10.10.2018 22:42:43</param>
```

```
    <param name="isshowtasklink">1</param>
```

```
    <param name="to_all_users">1</param>
```

```
    <param name="place"><![CDATA[]]></param>
```

```
    <param name="description"><![CDATA[]]></param>
```

```
    <param name="noticetype">1</param>
```

```
    <param name="sendbymail">1</param>
```

```
    <param name="sendbysms">0</param>
```

```
    <param name="sendbymotiw">1</param>
```

```
    <param name="meetingtime">10.10.2018 11:22:00</param>
```

```
    <param name="empids">20345</param>
```

```
    <param name="name"><![CDATA[]]></param>
```

```
<param name="flag">1</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idtask** – ID задачи (целое положительное число);

**idemployer** – ID пользователя, скрывающего задачу (целое положительное число);

**date** – дата показа задачи (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**isWithRemind** – может принимать два значения: 1 – скрыть задачу и создать напоминание о появлении задачи; 0 – скрыть задачу без напоминания;

**createdate** – дата создания напоминания (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**showdate** – дата показа напоминания (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**isshowtasklink** – может принимать два значения: 1 – отображать ссылку на задачу в напоминании; 0 – не отображать ссылку на задачу в напоминании;

**to\_all\_users** – может принимать два значения: 1 – скрыть задачу для всех участников (может только автор); 0 – скрыть задачу только в своём списке задач;

**place** – место проведения совещания, указывается при **noticetype** = 3 (строка);

**description** – описание напоминания (строка);

**noticetype** – тип напоминания; может принимать следующие значения: 0 – глобальное сообщение; 1 – пользовательское сообщение; 2 – напоминание; 3 – приглашение на совещание; 4 – автосообщение при приближении даты окончания согласования; 5 – напоминание приближения наступления контрольной точки;

**sendbyemail** – может принимать два значения: 1 – отправить напоминание по электронной почте; 0 – не отправлять напоминание по электронной почте;

**sendbysms** – может принимать два значения: 1 – отправить напоминание через SMS; 0 – не отправлять напоминание через SMS;

**sendbymotiw** – может принимать два значения: 1 – отправить напоминание системой «Мотив»; 0 – не отправлять напоминание системой «Мотив»;

**meetingtime** – дата проведения совещания, указывается при **noticetype** = 3 (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**empids** – массив ID пользователей, которым отправляется напоминание (целое положительное число; через запятую);

**name** – название напоминания (строка);

**flag** – может принимать два значения: 1 – скрыть задачу; 0 – показать задачу.

2.7.3.15. **taskExamine** – ознакомление пользователей с задачей

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskExamine">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">106195</param>
    <param name="idEmployer">1000</param>
    <param name="message"><![CDATA[Ознакомлен]]></param>
    <param name="alarm">0</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи по ознакомлению (целое положительное число);

**idEmployer** – ID пользователя, для которого выполняется функция (целое положительное число);

**message** – текст действия при ознакомлении с задачей (строка);

**alarm** – может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение.

2.7.3.16. **closetask** – закрытие задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="closeTask">
```

```

<param name="login">ivanov</param>
<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="idTask">0</param>
<param name="idEmployer">1337</param>
<param name="message"><![CDATA[Закрываю]]></param>
<param name="alarm">0</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID закрываемой задачи (целое положительное число);

**idEmployer** – ID пользователя, который закрывает задачу (целое положительное число);

**message** – текст действия при закрытии задачи (строка);

**alarm** – тип действия при закрытии задачи; может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение.

2.7.3.17. **rejectTask** – отправка задачи на доработку

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="rejectTask">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">1000</param>
    <param name="idEmployer">409</param>
    <param name="message"><![CDATA[Доработать]]></param>
    <param name="alarm">1</param>
  </function>
<client name="SomeProgram" version="1.0"/>

```

**</document>**

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи, которая отправляется на доработку (целое положительное число);

**idEmployer** – ID пользователя, который отправляет задачу на доработку (целое положительное число);

**message** – текст действия при отправке задачи на доработку (строка);

**alarm** – тип действия; может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение.

2.7.3.18. **taskReOpen** – восстановление задачи из архива

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="taskReOpen">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```
    <param name="idemployer">243</param>
```

```
    <param name="idtask">3042</param>
```

```
  </function>
```

```
  <client name="SomeProgram" version="1.0"/>
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который восстанавливает задачу (целое положительное число);

**idtask** – ID восстанавливаемой задачи (целое положительное число).

2.7.3.19. **getTasksSystemInfo** - получение задач для iOS-приложения

Вызов: **/rest/getTasksSystemInfo**

```
/**
```

```

* getTasksSystemInfo
*   Получает данные полей задач
* @param string $mode    Режим вызова:
*   -"all": все задачи пользователя;
*   -"allInFolders": все задачи в папках пользователя
* @param array $folderIds  Массив целочисленных идентификаторов папок
* @param array $taskIds    Массив целочисленных идентификаторов задач
* @param bool|string $lastchange  Временная метка для получения только задач,
изменивших свои выходные данные после этой отсечки
* @return array
*/

```

2.7.3.20. getTasksWorkGroupInfo - получение рабочих групп задач для iOS-приложения

Вызов: **/rest/getTasksWorkGroupInfo**

```

/**
* getTasksWorkGroupInfo
*   Получает данные рабочих групп задач
*   Параметры используются для определения выборки задач
* @param string $mode    Режим вызова:
*   -"all": все задачи пользователя;
*   -"allInFolders": все задачи в папках пользователя
* @param array $folderIds  Массив целочисленных идентификаторов папок
* @param array $taskIds    Массив целочисленных идентификаторов задач
* @param bool|string $lastchange  Временная метка для получения только задач,
изменивших свои выходные данные после этой отсечки
* @return array
*/

```

2.7.3.21. getTasksFiles - получение файлов задач для iOS-приложения

Вызов: **/rest/getTasksFiles**

```

/**

```

- \* getTasksFiles
- \* Получить список файлов задач
- \* Параметры используются для определения выборки задач
- \* @param string \$mode      Режим вызова:
  - \*    -"all": все задачи пользователя;
  - \*    -"allInFolders": все задачи в папках пользователя
- \* @param array \$folderIds   Массив целочисленных идентификаторов папок
- \* @param array \$taskIds     Массив целочисленных идентификаторов задач
- \* @param bool|string \$lastchange   Временная метка для получения только задач, изменивших свои выходные данные после этой отсечки
- \* @return mixed
- \*/

#### 2.7.4. Функции для работы со списком задач

##### 2.7.4.1. setNewInAction – установка/сброс признака «Новые действия»

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="setNewInAction">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">887294</param>
    <param name="idEmployer">1000</param>
    <param name="currentActionID">826973</param>
    <param name="flag">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи, для которой устанавливается признак (целое положительное число);

**idEmployer** – ID пользователя, который устанавливает признак *Новые действия* (целое положительное число);

**currentActionID** – ID последнего действия по задаче на момент вызова функции (целое положительное число);

**flag** – может принимать два значения: 1 – установить признак *Новые действия*; 0 – сбросить признак *Новые действия*.

При передаче значения 0 для параметра **flag** проверяется наличие действия более позднего по времени, чем действие, ID которого передан в параметре **currentActionID**. Если более позднего действия нет, то признак сбрасывается. Если действие есть, то признак не сбрасывается. Состояние признака при этом определяется по значению тега `<issetflag>`, которое возвращается в выходном XML. Если значение тега `<issetflag>` равно 1, то признак сбрасывается, если значение равно 0, то признак устанавливается.

Если для параметра «**flag**» передаётся значение 1, то признак устанавливается без предварительной проверки.

#### 2.7.4.2. setNewInTask – установка/сброс признака «Новая задача»

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="setNewInTask">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">887294</param>
    <param name="idEmployer">1000</param>
    <param name="flag">1</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи, для которой устанавливается признак (целое положительное число);

**idEmployer** – ID пользователя, который устанавливает/сбрасывает признак *Новая задача* (целое положительное число);

**flag** – может принимать два значения: 1 – установить признак *Новая задача*; 0 – сбросить признак *Новая задача*.

2.7.4.3. **resetNewInTask** – сброс признаков обновления при открытии задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="editAction">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idTask">887294</param>
    <param name="idEmployer">1000</param>
    <param name="currentActionID">826973</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idTask** – ID задачи (целое положительное число);

**idEmployer** – ID пользователя (целое положительное число);

**currentActionID** – ID последнего действия по задаче на момент вызова функции (целое положительное число).

**Замечание** – Вызов этой функции равнозначен вызовам двух функций: «setNewInTask» и «setNewInAction» с параметром **flag**, равным нулю.

2.7.4.4. **setNewActions** – установка/сброс признака «Новые действия» в REST

Функция для работы с REST API (подробнее см. п. [Работа с Rest API](#)).

Пример:

GET <https://localhost/rest/setNewActions> {"idTask": 37820, "flag": 1}

```

/**
 * setNewActions
 *
 * Управление флагом "Новые действия"
 * @param int $idTask   id задачи
 * @param int $flag     значение флага:
 * - 1 - установить флаг есть новое действие
 * - 0 - сбросить флаг есть новое действие
 * @return object
 */

```

Описание входных параметров:

**idTask** – ID задачи, для которой устанавливается признак (целое положительное число);

**flag** – может принимать два значения: 1 – установить признак *Новые действия*; 0 – сбросить признак *Новые действия*.

Ответ:

```

{
  "success": true,
  "result": true,
  "request": "setNewActions"
}

```

### 2.7.5. Функции для работы с проектами

#### 2.7.5.1. copyProject – копирование проекта со всеми его задачами

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="copyproject">
    <param name="login">ivanov</param>
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
    <param name="idemployer">2000</param>
    <param name="idsourceproject">155</param>
    <param name="projectname"><![CDATA[Проект2]]></param>
    <param name="projectdescription"><![CDATA[Проект2]]></param>
    <param name="startdate"><![CDATA[20.07.2018]]></param>
  </function>
</document>

```

```

    <param name="enddate"><![CDATA[20.09.2018]]></param>
    <param name="with_files">1</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который копирует проект (целое положительное число);

**idSourceProject** – ID исходного проекта (целое положительное число);

**projectname** – название нового проекта (строка);

**projectdescription** – описание нового проекта (строка);

**startdate** – дата начала нового проекта (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**enddate** – дата окончания нового проекта (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**with\_files** – может принимать два значения: 1 – копировать проект с файлами, прикрепленными к задачам проекта; 0 – копировать проект без файлов, прикрепленных к задачам проекта.

2.7.5.2. **getProjectsInfo** – получение информации о проекте

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="getProjectsInfo">
    <param name="login">ivanov</param>
    <param name="pass">21232f297a57a5a743894a0e4a801fc3</param>
    <param name="aProjectIDs">1,2,3</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**aProjectIDs** – массив ID проектов, информацию о которых требуется получить (целое положительное число; через запятую).

#### 2.7.6. Функции для работы с документами

##### 2.7.6.1. documentCreate – создание документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="documentCreate">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="document">
      <id_doctype>12</id_doctype>
      <id_author>212</id_author>
      <id_initiator>212</id_initiator>
      <id_realauthor>212</id_realauthor>
      <id_task>21234</id_task>
      <registration_date>01.12.2018 23:59:00</registration_date>
      <extended_number>5</extended_number>
      <fields>
        <field>ZXC1</field>
        <field>ZXC2</field>
        <field>
          <numfield>A</numfield>
          <numfield>B</numfield>
          <numfield>C</numfield>
          <numfield>D</numfield>
          <numfield>E</numfield>
        </field>
      </fields>
    </param>
  </function>
</document>
<routetemplate id="1087">
```

```

<routeremplatenode id="65396">
  <signhours>72</signhours>
  <remindbeforehours/>
</routeremplatenode>
<routeremplatenode id="65398">
  <signhours>72</signhours>
  <remindbeforehours>2</remindbeforehours>
</routeremplatenode>
<routeremplatenode id="65399">
  <signhours>72</signhours>
  <remindbeforehours/>
</routeremplatenode>
<routeremplatenode id="65397">
  <employers>
    <id>206</id>
  </employers>
  <signhours>48</signhours>
  <remindbeforehours/>
</routeremplatenode>
</routeremplate>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**id\_doctype** – ID типа документа (целое положительное число);

**id\_author** – ID автора документа (целое положительное число);

**id\_realauthor** – ID постановщика документа (целое положительное число);

**id\_task** – ID задачи, если документ создаётся по задаче (целое положительное число);

**registration\_date** – дата регистрации документа (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**extended\_number** – дополнительный номер (число).

#### **Замечания:**

1. У тега **<field>** вложения могут быть только для поля типа нумератор, если в нём есть поля для заполнения.

2. Содержание тега **<routetemplatenode>** можно получить с помощью функции **getRoutetemplateNodes**.

#### 2.7.6.2. **getDocumentInfo** – получение информации о карточке документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="getDocumentInfo">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="iddocument">1000</param>
    <param name="idemployer">2000</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**iddocument** – ID документа, информацию о карточке которого требуется получить (целое положительное число);

**idemployer** – ID пользователя (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <result value="0">
    <createdate><![CDATA[07.09.2018 11:15:22]]></createdate>
```

```

<regdate><![CDATA[03.09.2018 00:00:00]]></regdate>
<extendednumber><![CDATA[269560]]></extendednumber>
<doctype><![CDATA[Приказ]]></doctype>
<authorid>3308</authorid>
<authorname><![CDATA[Иванов Иван Иванович]]></authorname>
<fields>
  <field>
    <id>71</id>
    <name><![CDATA[Суть обращения]]></name>
    <fieldtype>3</fieldtype>
    <value><![CDATA[Ущерб имуществу.]]>
  </field>
  <field>
    <id>72</id>
    <name><![CDATA[Тип обращения]]></name>
    <fieldtype>4</fieldtype>
    <value><![CDATA[Жалоба]]></value>
  </field>
</fields>
</result>
</document>

```

Описание выходных параметров:

**createdate** – дата создания документа (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**regdate** – дата регистрации документа (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**extendednumber** – дополнительный номер (строка);

**authorid** – ID автора документа (целое положительное число);

**authorname** – ФИО автора документа (строка);

**name** – название поля (строка);

**fieldtype** – тип поля (целое положительное число);

**value** – значение поля в соответствии с форматом поля (например, для поля формата «файл» указывается полный путь к файлу, для поля формата «справочник» – ID полей, на которые ссылается данное поле и т.д.).

### 2.7.6.3. getTasksAndDocData – получение информации о полях задачи и документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getTasksAndDocData">
    <param name="login">ivanov</param>
    <param name="pass">60fd0775d7d425219b12d87b979f3ed8</param>
    <param name="idemployer">1000</param>
    <param name="doc_field_ids">
      <![CDATA[KIND,EXECORGANIZATION,SIGNPERSON]]>
    </param>
    <param name="task_ids">
      <![CDATA[2835165]]>
    </param>
    <param name="task_type_id">
      <![CDATA[560]]>
    </param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**iddocument** – ID документа, информацию о карточке которого требуется получить (целое положительное число);

**idemployer** – ID пользователя (целое положительное число);

**doc\_field\_ids** – Идентификаторы полей документа, информацию из которых требуется получить (заглавные латинские буквы и символы);

**task\_ids** – ID полей задачи, информацию о которых требуется получить (целое положительное число);

**task\_type\_id** - ID типа задачи, информацию о полях которой требуется получить (целое положительное число).

Пример выходного XML:

```
<!--?xml version="1.0" encoding="utf-8"?-->
<document>
  <info>
    <api_version>1.0</api_version>
    <time>11:12:53</time>
    <execution_time>2.5590298175812</execution_time>
  </info>
  <result value="0">
    <p1>
      <task_id>726</task_id>
      <document_id>244</document_id>
      <signperson>
        <value>
          <p1>
            <data>
              <!--[CDATA[Петров Дмитрий Артемович]]-->
            </data>
            <empid>
              <!--[CDATA[2000]]-->
            </empid>
            <dep>
              <!--[CDATA[Иванов]]-->
            </dep>
          </p1>
        </value>
      </signperson>
    </p1>
  </result>
</document>
```

```
<deptid>
  <!--[CDATA[2]]-->
</deptid>
</p1>
<p2>
  <data>
    <!--[CDATA[Генеральный директор]]-->
  </data>
  <empid>
    <!--[CDATA[20012]]-->
  </empid>
  <dep>
    <!--[CDATA[Главное подразделение]]-->
  </dep>
  <deptid>
    <!--[CDATA[1]]-->
  </deptid>
  <post>
    <!--[CDATA[Администрация]]-->
  </post>
</p2>
</value>
</signperson>
<execorganization>
  <value>
    <p1>
      <data>
        <!--[CDATA[Финансовый]]-->
      </data>
      <id>
        <!--[CDATA[535]]-->
```

```

    </id>
</p1>
<p2>
  <data>
    <!--[CDATA[Аналитика, ООО]]-->
  </data>
  <id>
    <!--[CDATA[458]]-->
  </id>
</p2>
</value>
</execorganization>
</p1>
</result>
</document>

```

Описание выходных параметров:

**value** – значение поля в соответствии с форматом поля (например, для поля типа «файл» указывается полный путь к файлу, для поля формата «справочник» – ID полей, на которые ссылается данное поле и т.д.). Пример: value/p1/data - ФИО, value/p1/post - должность.

**isSigned** – признак подписания файла ЭП для поля типа «Файл».

2.7.6.4. getRoutetemplateNodes – получение информации о маршруте рассмотрения документа

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getRoutetemplateNodes">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idroutetemplate">1141</param>
  </function>

```

```
<client name="SomeProgram" version="1.0"/>
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idroutetemplate** – ID маршрута (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
<info>
```

```
<version>3.7.5.4</version>
```

```
<api_version>1.0</api_version>
```

```
<time>14:28:44</time>
```

```
<execution_time>4.03932499886</execution_time>
```

```
</info>
```

```
<result value="0">
```

```
<routetemplatenodes>
```

```
<routetemplatenode id="65429">
```

```
<info_node>
```

```
<name>Произвольная рабочая группа</name>
```

```
</info_node>
```

```
<signhours>24</signhours>
```

```
<remindbeforehours>0</remindbeforehours>
```

```
<employers>
```

```
<id/>
```

```
<id/>
```

```
</employers>
```

```
</routetemplatenode>
```

```
<routetemplatenode id="65430">
```

```
<info_node>
```

```
<parent_id>65429</parent_id>
```

```

    <name>Произвольные начальники</name>
  </info_node>
  <signhours>24</signhours>
  <remindbeforehours>0</remindbeforehours>
  <employers>
    <id/>
    <id/>
  </employers>
</routetemplatenode>
<routetemplatenode id="65431">
  <info_node>
    <parent_id>65430</parent_id>
    <name>Исполнение</name>
  </info_node>
  <signhours>0</signhours>
  <remindbeforehours>0</remindbeforehours>
  <employers>
    <id/>
    <id/>
  </employers>
</routetemplatenode>
</routetemplatenodes>
</result>
</document>

```

Описание выходных параметров:

тег <routetemplatenode> :

- **id** – ID блока (целое положительное число);

тег <info\_node>:

- **name** – описание узла (строка);
- **parent\_id** – ID родительского блока (целое положительное число);
- **signhours** – длительность рассмотрения в часах (целое положительное число);

- **remindbeforehours** – количество часов, за которые необходимо послать напоминание (целое положительное число);

тег **<employers>**:

- **id** – ID пользователя, участвующего в рассмотрении (целое положительное число).

Тег **<routetemplatenodes>** выходного XML может использоваться в функции «**documentCreate**».

#### 2.7.6.5. documentConsider – рассмотрение документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="documentConsider">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idDocument">106195</param>
    <param name="idTask">106195</param>
    <param name="idEmployer">1337</param>
    <param name="idsubstitutor">1338</param>
    <param name="consider_type">0</param>
    <param name="message"><![CDATA[Рассмотрен]]></param>
    <param name="alarm">0</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idDocument** – ID рассматриваемого документа (целое положительное число);

**idTask** – ID задачи по рассмотрению документа (целое положительное число);

**idEmployer** – ID пользователя, который рассматривает или от чьего имени будет рассмотрен документ (целое положительное число);

**idsubstitutor** – ID заместителя рассматривающего пользователя;

**consider\_type** – тип рассмотрения/согласования. По умолчанию - 0. Типы рассмотрения: 0 - Обычное согласование, 1 - Согласование с замечаниями, 2 - Отказ от согласования.

**message** – текст действия при согласовании документа (строка);

**alarm** – может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение.

#### 2.7.6.6. coordDocument – принудительная смена статуса документа на «Рассмотрен»

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="coordDocument">
    <param name="login">ivanov</param>
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
    <param name="idemployer">2951</param>
    <param name="iddocument">1098</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который меняет статус (целое положительное число);

**idDocument** – ID документа, у которого меняется статус (целое положительное число).

#### 2.7.6.7. sendDocumentOnReject – отправка документа на доработку

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="sendDocumentOnReject">
    <param name="login">ivanov</param>
```

```
<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="idDocument">887294</param>
<param name="idTask">887294</param>
<param name="idEmployer">1337</param>
<param name="message"><![CDATA[Сделаю завтра.]]></param>
<param name="alarm">0</param>
<param name="flag">1</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idDocument** – ID документа (целое положительное число);

**idTask** – ID задачи по рассмотрению документа (целое положительное число);

**idEmployer** – ID пользователя, который отправляет документ на доработку (целое положительное число);

**message** – текст действия при отправке документа на доработку (строка);

**alarm** – тип действия; может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение;

**flag** – может принимать два значения: 1 – после доработки открывается задача по рассмотрению, которая была текущей в момент отправки документа на доработку; 0 – после доработки создаётся новый этап рассмотрения (после доработки начнётся рассмотрение с начала схемы).

#### 2.7.6.8. sendDocumentToArchive – отправка документа в архив

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="sendDocumentToArchive">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```

<param name="idDocument">106195</param>
<param name="idTask">106195</param>
<param name="idEmployer">1337</param>
<param name="message"><![CDATA[Рассмотрен.]]></param>
<param name="alarm">0</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idDocument** – ID документа (целое положительное число);

**idTask** – ID задачи по рассмотрению документа (целое положительное число);

**idEmployer** – ID пользователя, который отправляет документ в архив (целое положительное число);

**message** – текст действия при отправке документа в архив (строка);

**alarm** – тип действия; может принимать следующие значения: 0 – обычное действие; 1 – предупреждение; 2 – поощрение.

2.7.6.9. `sendDocumentToArchiveManual` – принудительная отправка документа в архив

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="sendDocumentToArchiveManual">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idDocument">1000</param>
    <param name="idEmployer">1000</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idDocument** – ID документа (целое положительное число);

**idEmployer** – ID пользователя, который отправляет документ в архив (целое положительное число).

#### 2.7.6.10. createNewStageOfDocumentConsideration – создание нового этапа рассмотрения документа

Возврат документа на рассмотрение

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="createNewStageOfDocumentConsideration">
    <param name="login">ivanov</param>
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
    <param name="idemployer">2951</param>
    <param name="iddocument">109438</param>
    <param name="reassignmentmanual">1</param>
    <param name="type">3</param>
    <param name="id_routetemplate">1</param>
    <files>
      ...
    </files>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
</code>
```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, который создаёт новый этап (целое положительное число);

**iddocument** – ID документа, по рассмотрению которого создаётся новый этап (целое положительное число);

**reassignmentmanual** – принудительно создается новый этап, может принимать значения:

1 – да, создание нового этапа с прикреплением новых версий файлов к документу (требуется, чтобы в параметрах передавалась вся маршрутная схема);

2 - да, дополнительно автоматически получить детали маршрута из **id\_routetemplate** (позволяет передать только **id\_routetemplate**, а все настройки маршрута будут взяты из БД);

0 – нет, создание нового этапа без прикрепления новых версий файлов к документу.

**type** – тип возврата на доработку:

0 - будет использована рекомендация выбранная при возврате на доработку;

1 – новая стадия на основе текущей схемы, параметр **reassignmentmanual** необходимо отправлять значение 1;

2 – возврат с доработки в ту же точку, параметр **reassignmentmanual** необходимо отправлять значение 1;

3 – новая стадия с использованием новой схемы, параметр **reassignmentmanual** необходимо обязательно отправлять значение 2;

**id\_routetemplate** – равен ID маршрутной схемы, маршрутная схема должна быть полностью заполнена (пользователи, длительность рассмотрения, уведомления)

#### 2.7.6.11. deleteDocument – удаление документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="deleteDocument">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idEmployer">1337</param>
    <param name="idDocument">106195</param>
  </function>
```

```
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который удаляет документ (целое положительное число);

**idDocument** – ID удаляемого документа (целое положительное число).

#### 2.7.6.12. approveResolution - утверждение проектов резолюций

```
/**
 * Утверждение проектов резолюций
 * @param int $idDocument
 * @param array $resolutionIDs
 * @param bool $hasClientAnnotations
 * @return bool
 */
```

Описание параметров:

**idDocument** – id документа;

**resolutionIDs** – массив id резолюций;

**hasClientAnnotations** - проверяет добавление новых комментариев на клиенте

Для утверждения резолюций:

Вызывать через /rest/approveResolution

idDocument = 0 (Если указать документ, то утверждаются все резолюции в документе, ID резолюций игнорируются)

resolutionIDs = массив id резолюций

hasClientAnnotations = 0 (для мобильного интерфейса, автоматическое утверждение проекта резолюции при написании комментария без перевода на исполнение)

Для работы с пунктами документов в резолюциях используются параметры:

resolution\_document\_paragraphs - значение настройки «Работа с пунктами документов», в типе документов на закладке Резолюции (0 - нет / 1 - Да (по-умолчанию 0))

resolution\_not\_add\_controler - значение настройки Не добавлять контролеров в текст резолюции, в типе документов на закладке Резолюции (0 - нет / 1 - Да (по-умолчанию 0))

resolution\_task\_name\_template - значение настройки Шаблон названия задачи по исполнению документа, в типе документов на закладке Резолюции (по-умолчанию null)

При установке значения resolution\_not\_add\_controler = 1, в текст резолюции не добавляются контролёры.

Эти настройки передаются в API вместе с документами в блоке "resolutionSettings" в getMobileData, getObjectData и getSeveralObjectsData.

#### 2.7.6.13. getDocumentParents - получение связанных документов

Возвращает все связанные документы

Пример входного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <function name="getDocumentParents">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="documentid">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

login – логин пользователя (строка);

pass – пароль пользователя (строка);

documentid - id документа.

## 2.7.7. Функции для работы с библиотекой

### 2.7.7.1. addLibraryFolder – создание папки библиотеки

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="addLibraryFolder">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="currentid">106195</param>
    <param name="id_obj">1000</param>
    <param name="obj_type">2</param>
    <param name="foldername"><![CDATA[Папка для файлов]]></param>
    <param name="idemployer">10</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**currentid** – ID родительской папки в библиотеке (целое положительное число);

**id\_obj** – ID объекта-родителя; тип объекта определяется в соответствии со значением параметра **obj\_type** (целое положительное число);

**obj\_type** – тип папки; может принимать следующие значения: 1 – системная папка подразделения, 2 – системная папка пользователя, 3 – пользовательская папка;

**foldername** – название папки (строка);

**idemployer** – ID пользователя (целое положительное число).

### 2.7.7.2. APIfolderSetRightsOnEmployer – назначение пользователю прав на папку библиотеки

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="APIfolderSetRightsOnEmployer">
```

```

<param name="login">ivanov</param>
<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="idFolder">1095</param>
<param name="idEmployer">164</param>
<param name="right">3</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idFolder** – ID папки, на которую даются права (целое положительное число);

**idEmployer** – ID пользователя, которому даются права (целое положительное число);

**right** – право доступа; может принимать значения: 0 – «Нет прав»; 1 – «Только чтение»; 2 – «Добавление»; 3 – «Изменение».

2.7.7.3. `renameLibraryFolder` – переименование папки библиотеки

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="renameLibraryFolder">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="currentid">106195</param>
    <param name="id_obj">1000</param>
    <param name="obj_type">2</param>
    <param name="foldername"><![CDATA[Папка для файлов]]></param>
    <param name="idemployer">10</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**currentid** – ID родительской папки в библиотеке (целое положительное число);

**id\_obj** – ID объекта-родителя; тип объекта определяется по значению параметра **obj\_type** (целое положительное число);

**obj\_type** – тип объекта-родителя; может принимать следующие значения: 1 – при создании нового подразделения; 2 – при создании нового пользователя; 3 – при создании пользователем новой папки в библиотеке;

**foldername** – новое название папки (строка);

**idemployer** – ID пользователя (целое положительное число).

#### 2.7.7.4. moveLibraryFolder – перемещение папки в библиотеке

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="moveLibraryFolder">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="id">1095</param>
    <param name="newparentid">1095</param>
    <param name="idemployer">1000</param>
    <param name="newdepid">1040</param>
    <param name="isdepartment">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**id** – ID перемещаемой папки (целое положительное число);

**newparentid** – ID папки, в которую перемещают исходную (целое положительное число);

**idemployer** – ID пользователя (целое положительное число);

**newdepid** – ID подразделения (в случае, если функцию вызвали при перемещении подразделения) (целое положительное число);

**isdepartment** – может принимать два значения: 1 – перемещение папки вызвано в связи с перемещением подразделения; 0 – ручное перемещение.

2.7.7.5. deleteLibraryFolder – удаление папки библиотеки

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="deleteLibraryFolder">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="currentid">106195</param>
    <param name="obj_type">2</param>
    <param name="deleteflag">0</param>
    <param name="idemployer">10</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**currentid** – ID удаляемой папки (целое положительное число);

**obj\_type** – тип объекта-родителя; может принимать следующие значения: 1 – при удалении подразделения; 2 – при удалении пользователя; 3 – при удалении папки пользователем;

**deleteflag** – может принимать два значения: 0 – выполняется проверка на наличие вложенных папок, файлов (в случае их наличия, возвращается ошибка); 1 – удаление папки без проверки на наличие вложенных папок и файлов;

**idemployer** – ID пользователя (целое положительное число).

2.7.7.6. addLibraryFile – добавление файла в библиотеку

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
```

```

<function name="moveLibraryFile">
  <param name="login">ivanov</param>
  <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
  <param name="idemployer">1095</param>
  <param name="idfolder">1103</param>
  <param name="files">
    <id>123</id>
    <tmp_name><![CDATA[C:/azxc.xml]]></tmp_name>
    <name><![CDATA[azxc.xml]]></name>
    <description><![CDATA[Приложение]]></description>
    <idtask>314</idtask>
    <filesize>1852</filesize>
  </param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя (целое положительное число);

**idfolder** – ID папки, в которую добавляется файл (целое положительное число);

**2.7.7.7. copyLibraryFile** – копирование файла из одной папки библиотеки в другую

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="copyLibraryFile">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idfolder">1095</param>
    <param name="files">
      <id>123</id>

```

```
<id>314</id>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idfolder** – ID папки-преемника (целое положительное число);

**id** – ID копируемых файлов (целое положительное число).

2.7.7.8. `moveLibraryFile` – перемещение файлов библиотеки из одной папки в другую

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="moveLibraryFile">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idfolder">1095</param> <!--ID папки преемника-->
    <param name="files">
      <id>123</id>
      <id>314</id>
    </param>
  </function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idfolder** – ID папки, в которую перемещается файл (целое положительное число);

**files** – массив ID перемещаемых файлов (целое положительное число).

### 2.7.7.9. deleteLibraryFile – удаление файла из библиотеки

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="deleteLibraryFile">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idemployer">1095</param>
    <param name="files">
      <id>123</id>
      <id>314</id>
    </param>
  </function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, который удаляет файл (целое положительное число);

**id** – ID удаляемых файлов (целое положительное число).

### 2.7.8. Функции для работы со справочниками, типами задач и типами таблиц

#### 2.7.8.1. catalogCreate – создание справочника/типа задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogCreate">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1</param>
    <param name="name"><![CDATA[Новый справочник]]></param>
```

```
<param name="iddepartment">1000</param>
<param name="idsystem">0</param>
<param name="recordtype">0</param>
<param name="quotient">0</param>
<param name="visible">0</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя (целое положительное число);

**iddepartment** – ID подразделения пользователя (целое положительное число);

**name** – название создаваемого справочника (строка);

**idsystem** – вид справочника; может принимать два значения: 1 – системный справочник; 0 – пользовательский справочник; Для типа задачи параметр должен быть равен 0. Для типа таблицы параметр должен быть равен 6.

**quotient** – признак общедоступности записей справочника; может принимать два значения: 1 – частные записи; 0 – общедоступные записи. Для типа таблиц параметр должен быть равен 0.

**visible** – возможность выбора признака общедоступности записи (параметр quotient) при её добавлении в справочник; может принимать два значения: 1 – выбор доступен; 0 – выбор недоступен.

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>11:18:58</time>
    <execution_time>0.0264701843262</execution_time>
  </info>
  <result value="0">
```

```
<id>139</id>
<message>139</message>
</result>
</document>
```

Описание выходных параметров:

**id** – ID созданного справочника, типа задачи или типа таблицы (целое положительное число).

2.7.8.2. **catalogRename** – переименование справочника/типа задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogRename">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1</param>
    <param name="name"><![CDATA[New dictionary]]></param>
    <param name="iddepartment">1000</param>
    <param name="idcatalog">139</param>
    <param name="quotient">0</param>
    <param name="visible">0</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя (целое положительное число);

**iddepartment** – ID подразделения пользователя (целое положительное число);

**idcatalog** – ID справочника/типа задачи/типа таблицы (целое положительное число);

**quotient** – признак общедоступности записей справочника; может принимать два значения: 1 – частные записи; 0 – общедоступные записи. Для типа таблиц параметр должен быть равен 0.

**visible** – возможность выбора признака общедоступности записи (параметр quotient) при её добавлении в справочник; может принимать два значения: 1 – выбор доступен; 0 – выбор не доступен. Для типа таблиц параметр должен быть равен 0.

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>11:32:00</time>
    <execution_time>0.038204908371</execution_time>
  </info>
  <result value="0">
    <name>New dictionary</name>
  </result>
</document>
```

2.7.8.3. catalogDelete – удаление справочника/типа задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogDelete">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1000</param>
    <param name="iddepartment">1000</param>
    <param name="idcatalog">100</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

---

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, удаляющего справочник/тип задачи/тип таблицы (целое положительное число);

**iddepartment** – ID подразделения пользователя, удаляющего справочник/тип задачи (целое положительное число);

**idcatalog** – ID удаляемого справочника/типа задачи или типа таблицы (целое положительное число).

2.7.8.4. taskTypeGetList – получение списка доступных пользователю типов задач

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskTypeGetList">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idemployer">341</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, список типов задач которого требуется получить (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>15:07:30</time>
    <execution_time>0.0423209667206</execution_time>
```

```

</info>
<result value="0">
  <tasktypes>
    <tasktype>
      <id>93</id>
      <name><![CDATA[type1]]></name>
    </tasktype>
    <tasktype>
      <id>94</id>
      <name><![CDATA[type2]]></name>
    </tasktype>
  </tasktypes>
</result>
</document>

```

Описание выходных параметров:

**id** – ID типа задачи (целое положительное число);

**name** – название типа задачи (строка).

2.7.8.5. catalogGetList – получение списка доступных пользователю справочников

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogGetList">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>14:00:20</time>
    <execution_time>0.0144839286804</execution_time>
  </info>
  <result value="0">
    <catalogs>
      <catalog>
        <id>9</id>
        <name><![CDATA[Название справочника1]]></name>
      </catalog>
      <catalog>
        <id>82</id>
        <name><![CDATA[Название справочника2]]></name>
      </catalog>
    </catalogs>
  </result>
</document>
```

Описание выходных параметров:

**id** – ID справочника, доступного пользователю (целое положительное число);

**name** – название справочника, доступного пользователю (строка).

2.7.8.6. catalogAddField – добавление поля в справочник/тип задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
```

```

<function name="catalogAddField">
  <param name="login">ivanov</param>
  <param name="pass">202cb962ac59075b964b07152d234b70</param>
  <param name="idemployer">1</param>
  <param name="name"><![CDATA[Города]]></param>
  <param name="idcatalog">5</param>
  <param name="type">8</param>
  <param name="not_null">0</param>
  <param name="unique">0</param>
  <param name="connect">15</param>
  <param name="issystem">0</param>
  <param name="uselist">0</param>
  <param name="listvalues"><![CDATA[Москва, Белгород]]></param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, добавляющего поле в справочник, тип задачи или тип таблицы (целое положительное число);

**name** – название поля (строка);

**idcatalog** – ID справочника, типа задачи или типа таблицы (целое положительное число);

**type** – тип поля (значение ID из таблицы `type_field`) (целое положительное число);

**not\_null** – признак обязательности заполнения поля; может принимать для справочников два значения: 1 – обязательное для заполнения поле; 0 – необязательное для заполнения поле; Для типов задач:

**unique** – признак уникальности значения поля; может принимать два значения: 1 – значение поля должно быть уникальным; 0 – значение поля может быть не уникальным;

**connect** – ID поля, на которое ссылается поле типа «ссылка» и «множественная ссылка» (целое положительное число);

**issystem** – вид поля справочника; может принимать два значения: 1 – системное поле; 0 – пользовательское поле;

**uselist** – использовать список значений при создании записи для полей типа «строка» (список значений указывается в параметре **listvalues**); может принимать два значения: 1 – добавление значений только из списка; 0 – добавление значений как из списка, так и вручную;

**listvalues** – список значений, перечисленных через запятую, для поля типа «строка» справочника/типа задачи/типа таблицы.

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>12:04:22</time>
    <execution_time>0.0662388801575</execution_time>
  </info>
  <result value="0">
    <id>756</id>
  </result>
</document>
```

2.7.8.7. catalogEditField – редактирование поля справочника/типа задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogEditField">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1</param>
    <param name="idfield">756</param>
    <param name="name"><![CDATA[новое название поля]]></param>
```

```
<param name="not_null">1</param>
<param name="unique">1</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, редактирующего поле (целое положительное число);

**name** – название редактируемого поля (строка);

**idcatalog** – ID справочника, типа задачи или типа таблицы, поле которого редактируется (целое положительное число);

**not\_null** – признак обязательности заполнения поля справочника/типа задачи; может принимать два значения: 1 – обязательное для заполнения поле; 0 – необязательное для заполнения поле;

**unique** – признак уникальности значения поля; может принимать два значения: 1 – уникальное значения; 0 – неуникальное значение;

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>12:48:45</time>
    <execution_time>0.0567698478699</execution_time>
  </info>
  <result value="0">
    <name>[CDATA[новое название поля]]</name>
  </result>
</document>
```

2.7.8.8. catalogDeleteField – удаление поля справочника/типа задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
  <function name="catalogDeleteField">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1000</param>
    <param name="iddepartment">1000</param>
    <param name="idfield">100</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, удаляющего поле из справочника/типа задачи/типа таблицы (целое положительное число);

**iddepartment** – ID подразделения пользователя, удаляющего поле из справочника/типа задачи/типа таблицы (целое положительное число);

**idfield** – ID удаляемого поля (целое положительное число).

2.7.8.9. **taskTypeGetFields** – получение списка полей типа задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskTypeGetFields">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idemployer">243</param>
    <param name="typetaskid">129</param>
    <param name="taskid">3042</param> <!-- необязательный параметр -->
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, для получения списка полей его типа задачи (целое положительное число);

**typeTaskId** – ID типа задачи, для которого возвращается список полей (целое положительное число);

**taskId** – ID задачи; если задано значение параметра, то возвращается список полей с их значениями для этой задачи (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <info>
```

```
    <api_version>1.0</api_version>
```

```
    <time>15:07:30</time>
```

```
    <execution_time>0.0423209667206</execution_time>
```

```
  </info>
```

```
  <result value="0">
```

```
    <fields>
```

```
      <field>
```

```
        <id>93</id>
```

```
        <name><![CDATA[name1]]></name>
```

```
        <idtype>1</idtype>
```

```
        <idconnect></idconnect>
```

```
        <isnessesary>1</isnessesary>
```

```
        <isunique>1</isunique>
```

```
        <usevalues>1</usevalues>
```

```
        <listvalues>
```

```
          <listvalue><![CDATA[value1]]></listvalue>
```

```
          <listvalue><![CDATA[value2]]></listvalue>
```

```
          <listvalue><![CDATA[value3]]></listvalue>
```

```
        </listvalues>
```

```

    <values>
      <value><![CDATA[value1]]></value>
    </values>
  </field>
  <field>
    <id>94</id>
    <name><![CDATA[name2]]></name>
    <idtype>2</idtype>
    <idconnect></idconnect>
    <isnessesary>0</isnessesary>
    <isunique>0</isunique>
    <values>
      <value><![CDATA[text]]></value>
    </values>
  </field>
</fields>
</result>
</document>

```

2.7.8.10. catalogAddRecord – добавление записи в справочник

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogAddRecord">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1</param>
    <param name="iddepartment">1</param>
    <param name="iddictionary">19</param>
    <param name="dictrecords">
      <onedictvalue>
        <fieldid>112</fieldid>

```

```
<values>
  <value><![CDATA[значение поля]]></value>
</values>
</onedictvalue>
<onedictvalue>
  <fieldid>113</fieldid>
  <values>
    <value>12</value>
  </values>
</onedictvalue>
<onedictvalue>
  <fieldid>18</fieldid>
  <values>
    <value>45</value>
    <value>46</value>
    <value>47</value>
  </values>
</onedictvalue>
<onedictvalue>
  <fieldid>16</fieldid>
  <values>
    <value><![CDATA[полный путь к файлу]]></value>
    <value><![CDATA[полный путь к файлу]]></value>
    <value><![CDATA[полный путь к файлу]]></value>
  </values>
</onedictvalue>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

---

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя (целое положительное число);

**iddepartment** – ID подразделения (целое положительное число);

**iddictionary** – ID справочника (целое положительное число);

**fieldid** – ID поля (целое положительное число);

**value** – значение поля в соответствии с форматом поля (например, для поля формата «файл» указывается полный путь к файлу, для поля формата «множественная ссылка» – ID полей, на которые ссылается данное поле и т.д.).

2.7.8.11. catalogEditRecord – редактирование записи справочника

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="catalogEditRecord">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
```

```
    <param name="idemployer">1</param>
```

```
    <param name="iddepartment">1</param>
```

```
    <param name="iddictionary">9</param>
```

```
    <param name="idrecord">2107</param>
```

```
    <param name="dictrecords">
```

```
      <onedictvalue>
```

```
        <fieldid>112</fieldid>
```

```
        <values>
```

```
          <value><![CDATA[значение поля]]></value>
```

```
        </values>
```

```
      </onedictvalue>
```

```
      <onedictvalue>
```

```
        <fieldid>113</fieldid>
```

```
        <values>
```

```
          <value>12</value>
```

```
        </values>
```

```

</onedictvalue>
<onedictvalue>
  <fieldid>18</fieldid>
  <values>
    <value>45</value>
    <value>46</value>
    <value>47</value>
  </values>
</onedictvalue>
<onedictvalue>
  <fieldid>16</fieldid>
  <values>
    <value><![CDATA[полный путь к файлу]]></value>
    <value><![CDATA[полный путь к файлу]]></value>
    <value><![CDATA[полный путь к файлу]]></value>
  </values>
</onedictvalue>
</param>
</function>
<client name="SomeProgram" version="1.0" />
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя (целое положительное число);

**iddepartment** – ID подразделения пользователя (целое положительное число);

**iddictionary** – ID справочника, в котором находится запись (целое положительное число);

**idrecord** – ID редактируемой записи (целое положительное число);

**fieldid** – ID поля; поля, для которых не будет ничего передано, считаются удалёнными (целое положительное число);

**value** – значение поля в соответствии с форматом поля (например, для поля формата «файл» указывается полный путь к файлу, для поля формата «множественная ссылка» – ID полей, на которые ссылается данное поле и т.д.).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>12:37:02</time>
    <execution_time>0.268166065216</execution_time>
  </info>
  <result value="0" />
</document>
```

2.7.8.12. catalogDeleteRecord – удаление записи справочника

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="catalogDeleteRecord">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1000</param>
    <param name="iddepartment">1000</param>
    <param name="idrecord">100</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, удаляющего запись (целое положительное число);

**iddepartment** – ID подразделения пользователя, удаляющего запись (целое положительное число);

**idrecord** – ID удаляемой записи (целое положительное число).

2.7.8.13. catalogGetRecord – получение записи справочника

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <function name="catalogGetRecord">
    <param name="login">ivanov</param>
    <param name="pass">202cb962ac59075b964b07152d234b70</param>
    <param name="idemployer">1</param>
    <param name="idrecord">14</param>
    <param name="iddictionary">6</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя (целое положительное число);

**idrecord** – ID записи, которую необходимо получить (целое положительное число);

**iddictionary** – ID справочника, запись которого требуется получить (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>11:08:18</time>
    <execution_time>0.063570022583</execution_time>
  </info>
  <result value="0">
    <field>
```

```

    <fieldid>12</fieldid>
    <fieldname><![CDATA[Название]]></fieldname>
    <fieldvalue><![CDATA[Москва]]></fieldvalue>
  </field>
  <field>
    <fieldid>13</fieldid>
    <fieldname><![CDATA[Страна]]></fieldname>
    <fieldvalue><![CDATA[Россия]]></fieldvalue>
  </field>
</result>
</document>

```

Описание выходных параметров:

**fieldid** – ID поля (целое положительное число);

**fieldname** – название поля (строка);

**fieldvalue** – значение поля в соответствии с форматом поля (например, для поля формата «файл» указывается полный путь к файлу, для поля формата «множественная ссылка» – ID полей, на которые ссылается данное поле и т.д.).

2.7.8.14. `getCatalogRecordsExt` – получение списка (расширенного) записей справочника

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getCatalogRecordsExt">
    <param name="login">ivanov</param>
    <param name="pass">21232f297a57a5a743894a0e4a801fc3</param>
    <param name="aCatRecFlag">
      <rec>
        <IDCatalog>1</IDCatalog>
        <LastChange>13:22:24 06.06.2018</LastChange>
        <aRecIDs>1,2,3,4,5</aRecIDs>
        <Flag>INARRAY</Flag>
      </rec>
    </param>

```

```
</function>
<client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**IDCatalog** – ID справочника (целое положительное число);

**LastChange** – дата, позже которой будут выбраны записи (ГГГГ:ММ:ЧЧ ЧЧ:ММ:СС);

**aRecIDs** – ID записей справочника (целое положительное число);

**flag** – определяет по каким записям из заданного массива получать информацию; может принимать следующие значения:

**ALL** – все записи (**aRecIDs** не учитывается), в данном случае должно быть задано значение для **idCatalog**;

**INARRAY** – записи, ID которых указаны в **aRecIDs**;

**NOTINARRAY** – все записи, кроме тех, ID которых указаны в **aRecIDs**.

**Замечание** – При выборе значений для нескольких справочников следует создать тег **<rec>** для каждого справочника.

#### 2.7.9. Функции для работы со словарями

##### 2.7.9.1. editDictionary – создание, редактирование и удаление словаря

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="editDictionary">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="sign">renamecatalog</param>
    <param name="catalog_id">280</param> <!-- ID словаря-->
    <param name="catalogname"><![CDATA[Колеса]]></param>
    <param name="readonly">1</param>
    <param name="id_owner">1564</param>
  </function>
</client name="SomeProgram" version="1.0" />
```

**</document>**

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**sign** – флаг, указывающий какое действие выполняется; может принимать следующие значения:

**createcatalog** – добавление нового словаря;

**renamecatalog** – переименование словаря;

**saveascatalog** – копирование словаря;

**delcatalog** – удалить словарь.

**catalog\_id** – ID словаря (целое положительное число);

**catalogname** – название словаря (строка);

**readonly** – право доступа к словарю «Только чтение»; может принимать два значения: 1 – право доступа «Только чтение»; 0 – нет;

**id\_owner** – ID пользователя, который является владельцем словаря (целое положительное число).

2.7.9.2. editDictionaryWord – редактирование элемента словаря

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="editDictionaryWord">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
```

```
    <param name="sign">addword</param>
```

```
    <param name="catalog_id">280</param>
```

```
    <param name="wordid">20080</param>
```

```
    <param name="wordname"><![CDATA[Колесо.]]></param>
```

```
    <param name="wordvalue"><![CDATA[R14 Guardian]]></param>
```

```
  </function>
```

```
  <client name="SomeProgram" version="1.0"/>
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**sign** – флаг, указывающий какое действие выполняется; может принимать следующие значения:

**addword** – добавление элемента в словарь;

**delword** – удаление элемента из словаря;

**changeword** – редактирование элемента словаря (строка);

**catalog\_id** – ID словаря (целое положительное число);

**wordid** – ID элемента в словаре (целое положительное число);

**wordname** – название элемента словаря (строка);

**wordvalue** – значение элемента словаря (строка).

2.7.10. Функции для работы со счётчиками документов/ задач

2.7.10.1. editDocCounter – редактирование счётчика документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="editDocCounter">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
```

```
    <param name="id_department">157</param>
```

```
    <param name="id_doctype">280</param>
```

```
    <param name="id_employer">20080</param>
```

```
    <param name="is_del_counter">0</param>
```

```
    <param name="reg_number">0</param>
```

```
    <param name="resetdate">15.10.2018</param>
```

```
    <param name="type"><![CDATA[doctype]]</param>
```

```
  </function>
```

```
  <client name="SomeProgram" version="1.0"/>
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**id\_department** – ID подразделения, к которому привязан счётчик (целое положительное число);

**id\_doctype** – ID типа документа/ типа задачи, к которой привязан счётчик (строка);

**id\_employer** – ID пользователя, который редактирует счётчик (целое положительное число);

**is\_del\_counter** – может принимать два значения: 1 – удалить счётчик; 0 – не удалять счётчик;

**reg\_number** – значение счётчика (целое положительное число);

**resetdate** – дата обнуления счётчика (ДД.ММ.ГГГГ ЧЧ:ММ:СС).

**type** – doctype – счётчик документа, tasktype – счётчик задачи.

2.7.10.2. getDocCounter – получение информации о счётчике документа

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<document>
```

```
  <function name="getDocCounter">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
```

```
    <param name="id_department">157</param>
```

```
    <param name="id_doctype">280</param>
```

```
    <param name="id_employer">20080</param>
```

```
    <param name="type"><![CDATA[doctype]]</param>
```

```
  </function>
```

```
  <client name="SomeProgram" version="1.0" />
```

```
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**id\_department** – ID подразделения, к которому привязан счётчик (целое положительное число);

**id\_doctype** – ID типа документа/ типа задачи, к которой привязан счётчик (целое положительное число);

**id\_employer** – ID пользователя (целое положительное число).

**type** – doctype – счетчик документа, **tasktype** – счетчик задачи.

2.7.11. Функции для работы с внешними пользователями/доменами

2.7.11.1. **addOuterUser** – добавление внешнего пользователя/домена в опрашиваемый ящик пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="addOuterUser">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idmailbox">23</param>
    <param name="idemployer">20025</param>
    <param name="outerusername"><![CDATA[Имя внешнего пользователя]]</param>
    <param name="outeruseremail"><![CDATA[mail.ru]]</param>
    <param name="isdomain">1</param>
    <param name="isinblacklist">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idmailbox** – ID опрашиваемого ящика; если **idmailbox=0**, то функция добавляет внешнего пользователя или домен в первый активный ящик пользователя (целое положительное число);

**idemployer** – ID пользователя, являющегося владельцем опрашиваемого ящика (целое положительное число);

**outerusername** – имя внешнего пользователя или домена (в зависимости от значения параметра **isdomain**);

**outeruseremail** – email внешнего пользователя или домена (в зависимости от значения параметра **isdomain**);

**isdomain** – может принимать два значения: 1 – добавление домена; 0 – добавление внешнего пользователя;

**isinblacklist** – может принимать два значения: 1 – внести пользователя/домен в запрещённый список; 0 – не вносить пользователя/домен в запрещённый список.

2.7.11.2. **editOuterUser** – редактирование внешнего пользователя/домена

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="editOuterUser">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idouteruser">255</param>
    <param name="idemployer">20025</param>
    <param name="outerusername"><![CDATA[Новое имя]]</param>
    <param name="isinblacklist">1</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idouteruser** – ID редактируемого внешнего пользователя/домена (целое положительное число);

**idmailbox** – ID опрашиваемого ящика; если установлено, то внешний пользователь перемещается в этот ящик (целое положительное число);

**idemployer** – ID пользователя, являющегося владельцем опрашиваемого ящика (целое положительное число);

**outerusername** – новое имя внешнего пользователя/домена (строка);

**outeruseremail** – новый email внешнего пользователя/домена (строка);

**isinblacklist** – может принимать следующие значения: 1 – добавить в список запрещённых пользователей/доменов; 0 – удалить из списка запрещённых пользователей/доменов.

2.7.11.3. **deleteOuterUser** – удаление внешнего пользователя/домена

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
  <function name="deleteOuterUser">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idemployer">20025</param>
    <param name="idouteruser">255</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idouteruser** – ID внешнего пользователя/домена, который удаляется из Системы (целое положительное число);

**idemployer** – ID пользователя, который является владельцем удаляемого ящика (целое положительное число).

2.7.11.4. deleteMailbox – удаление почтового ящика пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="deleteMailbox">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idmailbox">25</param>
    <param name="idemployer">20025</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idmailbox** – ID удаляемого ящика (целое положительное число);

**idemployer** – ID пользователя, который является владельцем ящика (целое положительное число).

**Замечание** – При удалении почтового ящика также удаляются внешние пользователи и домены, привязанные к удаляемому почтовому ящику.

#### 2.7.12. Функции для работы с папками задач

##### 2.7.12.1. taskFolderCreate – создание папки задач

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskFolderCreate">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idEmployer">106195</param>
    <param name="folderName"><![CDATA[Срочные задачи]]></param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который создаёт папку (целое положительное число);

**folderName** – название папки (строка).

##### 2.7.12.2. taskFolderRename – переименование папки задач

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskFolderRename">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idfolder">50</param>
```

```
<param name="folderName"><![CDATA[Срочные задачи]]></param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**IdFolder** – ID папки, название которой редактируется (целое положительное число);

**folderName** – новое название папки (строка).

2.7.12.3. **taskFolderDelete** – удаление папки задач

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskFolderDelete">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idEmployer">106195</param>
    <param name="idFolder">50</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, который удаляет папку (целое положительное число);

**idFolder** – ID удаляемой папки (целое положительное число).

2.7.12.4. **getUserTaskFolders** – получение списка папок задач пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getUserTaskFolders">
```

```

<param name="login">ivanov</param>
<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="idEmployer">1337</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, список папок задач которого требуется получить (целое положительное число).

### 2.7.13. Функции для работы с напоминаниями

#### 2.7.13.1. taskNoticeCreate – создание напоминания

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskNoticeCreate">
    <param name="login">ivanov</param>
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
    <param name="idemployer">2951</param>
    <param name="idtask">109438</param>
    <param name="name"><![CDATA[День рождения!]]></param>
    <param name="description"><![CDATA[Поздравить сотрудника]]></param>
    <param name="showdate">00:10 10.2018</param>
    <param name="noticetype">1</param>
    <param name="place"><![CDATA[]]></param>
    <param name="sendbymail">1</param>
    <param name="sendbysms">0</param>
    <param name="sendbymotiw">1</param>
    <param name="meetingtime">14:00:00 10.10.2018</param>
  </function>
</document>

```

```
<param name="empids">20345</param>
<param name="isrepeated">1</param>
<param name="repeat_type">4</param>
<param name="repeat_opt_1">1</param>
<param name="daily_val">1</param>
<param name="weekly_val">1</param>
<param name="week_days">3</param>
<param name="repeat_opt_3">1</param>
<param name="month_day">10</param>
<param name="monthly_val">1</param>
<param name="monthly_val2">1</param>
<param name="month_sp_num">1</param>
<param name="month_sp_week">0</param>
<param name="repeat_opt_4">1</param>
<param name="year_month1">12</param>
<param name="year_month2">10</param>
<param name="year_sp_week">3</param>
<param name="year_sp_num">1</param>
<param name="year_day">17</param>
<param name="repeat_limit_type">1</param>
<param name="limit_count">10</param>
<param name="limit_date">17.10.2018</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, который создаёт напоминание (целое положительное число);

**idtask** – ID задачи, если создаётся напоминание по задаче (целое положительное число);

**name** – название напоминания (строка);

**description** – описание напоминания (строка);

**showdate** – дата показа напоминания (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**noticetype** – тип напоминания; может принимать следующие значения:

- 0 – глобальное сообщение;
- 1 – пользовательское сообщение;
- 2 – напоминание;
- 3 – приглашение на совещание;
- 4 – автосообщение при приближении даты окончания согласования;
- 5 – напоминание приближения наступления контрольной точки;

**place** – место проведения совещания, указывается при **noticetype** = 3 (строка);

**sendbymail** – может принимать два значения: 1 – отправить напоминание по электронной почте; 0 – не отправлять напоминание по электронной почте;

**sendbysms** – может принимать два значения: 1 – отправить напоминание через SMS; 0 – не отправлять напоминание через SMS;

**sendbymotiw** – может принимать два значения: 1 – отправить напоминание системой «Мотив»; 0 – не отправлять напоминание системой «Мотив»;

**meetingtime** – дата проведения совещания, указывается при **noticetype** = 3 (ДД.ММ.ГГГГ ЧЧ:ММ:СС);

**empids** – массив ID пользователей, которым отправляется напоминание (целое положительное число; через запятую);

**isrepeated** – периодически повторяемое напоминание; может принимать два значения: 1 – периодически повторяемое напоминание; 0 – нет;

**repeat\_type** – тип повторения; может принимать следующие значения: 1 – каждый день; 2 – каждую неделю; 3 – каждый месяц; 4 – каждый год;

**repeat\_opt\_1** – настройки для типа повторения «Каждый день»; может принимать следующие значения: 1 – каждый кратный **daily\_val** день; 2 – каждый рабочий день;

**daily\_val** – кратность ежедневного повторения (например, **daily\_val**=2 каждый 2 день при типе повторения **repeat\_type**=1) (целое положительное число);

**weekly\_val** – кратность еженедельного повторения (например, **weekly\_val**=2 каждую 2 неделю при типе повторения **repeat\_type**=2) (целое положительное число);

**week\_days** – дни, в которые отправляется напоминание; может принимать следующие значения: 1 – понедельник; 2 – вторник; 3 – среда; 4 – четверг; 5 – пятница; 6 – суббота; 7 – воскресенье (дни указывать через запятую);

**repeat\_opt\_3** – настройки для типа повторения «Каждый месяц»; может принимать следующие значения: 1 – повторять каждый **month\_day** каждого **monthly\_val** месяца; 2 – повторять каждый **month\_sp\_num month\_sp\_week** каждого **monthly\_val2** месяца;

**month\_day** – день повторения при **repeat\_opt\_3=1** (число от 1 до 31);

**monthly\_val** – кратность ежемесячного повторения при **repeat\_opt\_3=1** (например, повторять каждый **month\_day** числа каждого **monthly\_val2** месяца);

**monthly\_val2** – кратность ежемесячного повторения при **repeat\_opt\_3=2** (например, повторять каждый **month\_sp\_num month\_sp\_week** каждого **monthly\_val2** месяца) (целое положительное число);

**month\_sp\_num** – номер недели при **repeat\_opt\_3=2** (например, повторять каждый второй (**month\_sp\_num=2**) вторник (**month\_sp\_week=2**) каждого 3 месяца (**monthly\_val2=3**)) (1, 2, 3, 4, последний);

**month\_sp\_week** – день повторения при **repeat\_opt\_3=2**; может принимать следующие значения: -1 – день; -2 – рабочий день; -3 – выходной день; 1 – понедельник; 2 – вторник; 3 – среда; 4 – четверг; 5 – пятница; 6 – суббота; 7 – воскресенье;

**repeat\_opt\_4** – настройки для типа повторения «Каждый год»; может принимать следующие значения: 1 – повторять каждый **year\_day** каждого **year\_month1** месяца; 2 – повторять каждый **year\_sp\_num year\_sp\_week** каждого **year\_month2** месяца;

**year\_day** – день месяца при **repeat\_opt\_4=1** (число от 1 до 31);

**year\_month1** – номер месяца при **repeat\_opt\_4=1** (число от 1 до 12);

**year\_month2** – номер месяца при **repeat\_opt\_4=2** (число от 1 до 12);

**year\_sp\_week** – день повторения при **repeat\_opt\_4=2**; может принимать следующие значения: -1 – день; -2 – рабочий день; -3 – выходной день; 1 – понедельник; 2 – вторник; 3 – среда; 4 – четверг; 5 – пятница; 6 – суббота; 7 – воскресенье;

**year\_sp\_num** – номер недели при **repeat\_opt\_4=2** (например, повторять каждый третий (**month\_sp\_num=3**) четверг (**month\_sp\_week=4**) каждого января (**monthly\_val2=1**)) (1, 2, 3, 4, последний);

**repeat\_limit\_type** – предел повторения; может принимать следующие значения: 1 – нет даты окончания; 2 – завершить после **limit\_count** повторений; 3 – дата окончания **limit\_date**;

**limit\_count** – количество повторений при **repeat\_limit\_type=2** (целое положительное число);

**limit\_date** – дата окончания повторений включительно при **repeat\_limit\_type=3** (ДД.ММ.ГГГГ).

Выходной XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>11:21:43</time>
    <execution_time>0.85905694962</execution_time>
  </info>
  <result value="0">
    <notice_id>1515</notice_id>
  </result>
</document>
```

2.7.13.2. taskNoticeView – ознакомление с напоминанием

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskNoticeView">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idnotice">106195</param>
    <param name="idemployer">1337</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idnotice** – ID напоминания (целое положительное число);

**idemployer** – ID пользователя, для которого выполняется функция (целое положительное число).

2.7.13.3. **taskNoticeDelete** – удаление напоминания по его ID

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="taskNoticeDelete">
    <param name="login">ivanov</param>
    <param name="pass">e37f0136aa3ffaf149b351f6a4c948e9</param>
    <param name="idemployer">1315</param>
    <param name="idnotice">48657</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, удаляющего напоминание (целое положительное число);

**idnotice** – ID удаляемого напоминания (целое положительное число).

Пример выходного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <info>
    <api_version>1.0</api_version>
    <time>12:44:37</time>
    <execution_time>0.244467973709</execution_time>
  </info>
  <result value="0" />
</document>
```

2.7.14. Функции для работы с шаблонами задач

2.7.14.1. createTaskTemplate – создание шаблона задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="createtasktemplate">
    <param name="id_employer">2000</param>
    <param name="name"><![CDATA[Шаблон1]]></param>
    <param name="id_initiator">500</param>
    <param name="authors"><![CDATA[123,566]]></param>
    <param name="resppersons"><![CDATA[123,566]]></param>
    <param name="controllers"><![CDATA[123,566]]></param>
    <param name="workers"><![CDATA[123,566]]></param>
    <param name="id_project">144</param>
    <param name="day_duration">15</param>
    <param name="period_id">5</param>
    <param name="priority">1</param>
    <param name="id_department">654</param>
    <param name="id_folder">333</param>
    <param name="id_tasktype">589</param>
    <param name="taskTypeFields" type="array_assoc">
      <3045>"test"</3045>
    </param>
  </function>
</document>
```

```

<3046>"Описание createTaskTemplate!"</3046>
<3047>"1992"</3047>
<3048>"12.12"</3048>
<3049>"64625"</3049>
<3050>"64626,64627,67965"</3050>
<3051>"1"</3051>
<3052>"24498"</3052>
<3054>"123"</3054>
<3057>"23.06.2018"</3057>
<3058>"10"</3058>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**idEmployer** – ID пользователя, который создаёт шаблон (целое положительное число);

**name** – название шаблона задачи (строка);

**authors** – массив ID авторов задачи (целое положительное число; через запятую);

**resppersons** – массив ID ответственных руководителей задачи (целое положительное число; через запятую);

**controllers** – массив ID контролёров задачи (целое положительное число; через запятую);

**workers** – массив ID исполнителей задачи (целое положительное число; через запятую);

**id\_project** – ID проекта, к которому прикрепляется задача (целое положительное число);

**day\_duration** – длительность задачи (целое положительное число);

**period\_id** – id периода, в чем измеряется длительность задачи;

**priority** – приоритет задачи; может принимать два значения: 1 – важная задача; 0 – обычная задача;

**id\_department** – ID подразделения проекта (целое положительное число);

**id\_folder** – ID папки, в которую будет помещена задача (целое положительное число);

**id\_tasktype** – ID типа задачи (целое положительное число).

**taskTypeFields** – ryfxtybz gjkmpjdfntkmcrb[ gjktq из типа задачи, все значения в taskTypeFields приходят в виде строк.

Формат описания каждого поля : <id поля>значение поля</id поля>

Доступные типы полей:

**Строка, текст, нумератор** – значение: строка.

**Целое** – значение: целое число.

**Вещественное** – значение: вещественное число.

**Ссылка на справочник** – значение: ID записи справочника (meta\_record.id).

**Множественная ссылка на справочник** – значение: строка с id записей справочника (meta\_record.id) через запятую.

**Логическое** – значение: 1 | 0 | "" (Да | Нет | Значение не выбрано).

**Ссылка на библиотеку** – значение: id записи библиотеки (dlfile.id).

**Дата** – значение: формат даты.

**Подразделение** – значение: ID подразделения.

2.7.14.2. editTaskTemplate – редактирование шаблона задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<document>
```

```
  <function name="edittasktemplate">
```

```
    <param name="login">ivanov</param>
```

```
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
```

```
    <param name="id_employer">2000</param>
```

```
    <param name="id_template">155</param>
```

```
    <param name="name"><![CDATA[Шаблон1]]></param>
```

```
    <param name="id_initiator">500</param>
```

```
    <param name="authors"><![CDATA[123,566]]></param>
```

```
    <param name="resppersons"><![CDATA[123,566]]></param>
```

```
    <param name="controllers"><![CDATA[123,566]]></param>
```

```
    <param name="workers"><![CDATA[123,566]]></param>
```

```
    <param name="id_project">144</param>
```

```
    <param name="day_duration">15</param>
```

```

<param name="period_id">5</param>
<param name="priority">1</param>
<param name="id_department">654</param>
<param name="id_folder">333</param>
<param name="id_tasktype">589</param>
<param name="id_cp_template">5</param>
<param name="taskTypeFields" type="array_assoc">
  <3045>"test"</3045>
  <3046>"Описание createTaskTemplate!"</3046>
  <3047>"1992"</3047>
  <3048>"12.12"</3048>
  <3049>"64625"</3049>
  <3050>"64626,64627,67965"</3050>
  <3051>"1"</3051>
  <3052>"24498"</3052>
  <3054>"123"</3054>
  <3057>"23.06.2018"</3057>
  <3058>"10"</3058>
</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**id\_employer** – ID пользователя, который создаёт шаблон (целое положительное число);

**id\_template** – ID редактируемого шаблона (целое положительное число);

**name** – название шаблона (строка);

**authors** – массив ID авторов задачи (целое положительное число; через запятую);

**resppersons** – массив ID ответственных руководителей задачи (целое положительное число; через запятую);

**controllers** – массив ID контролёров задачи (целое положительное число; через запятую);

**workers** – массив ID исполнителей задачи (целое положительное число; через запятую);

**id\_project** – ID проекта, к которому прикрепляется задача (целое положительное число);

**day\_duration** – длительность задачи (целое число);

**period\_id** – id периода, в чем измеряется длительность задачи;

**priority** – приоритет задачи; может принимать два значения: 1 – важная задача; 0 – обычная задача;

**id\_department** – ID подразделения проекта (целое положительное число);

**id\_folder** – ID папки, в которую будет помещена задача (целое положительное число);

**id\_tasktype** – ID типа задачи (целое положительное число);

**id\_cp\_template** – ID шаблона контрольной точки (целое положительное число).

**taskTypeFields** – значения пользовательских полей из типа задачи, все значения в **taskTypeFields** приходят в виде строк.

Формат описания каждого поля : <id поля>значение поля</id поля>

Доступные типы полей:

**Строка, текст, нумератор** – значение: строка.

**Целое** – значение: целое число.

**Вещественное** – значение: вещественное число.

**Ссылка на справочник** – значение: ID записи справочника (**meta\_record.id**).

**Множественная ссылка на справочник** – значение: строка с id записей справочника (**meta\_record.id**) через запятую.

**Логическое** – значение: 1 | 0 | "" (Да | Нет | Значение не выбрано).

**Ссылка на библиотеку** – значение: id записи библиотеки (**dlfile.id**).

**Дата** – значение: формат даты.

**Подразделение** – значение: ID подразделения.

2.7.14.3. **deleteTaskTemplate** – удаление шаблона задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
```

```
<function name="deletetasktemplate">
  <param name="login">ivanov</param>
  <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
  <param name="id_employer">2000</param>
  <param name="id_template">155</param>
</function>
<client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**id\_employer** – ID пользователя, который удаляет шаблон (целое положительное число);

**id\_template** – ID удаляемого шаблона (целое положительное число).

2.7.14.4. getTaskTemplate – получение информации о шаблоне задачи

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="gettasktemplate">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="id_employer">2000</param>
    <param name="id_template">155</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idEmployer** – ID пользователя, от имени которого выполняется функция (целое положительное число);

**id\_template** – ID шаблона (целое положительное число).

---

## 2.7.15. Функции для работы с шаблонами изолированных рабочих групп

### 2.7.15.1. saveIWGTemplate – создание шаблона изолированной рабочей группы

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="saveIWGTemplate">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idemployer">2000</param>
    <param name="templates">
      <template id='153'>
        <name><![CDATA[Шаблон1]]></name>
        <workgroups>
          <group id='25' is_post_sysacts='1'>
            <resppersons><![CDATA[45,236,1]]></resppersons>
            <employers><![CDATA[45,236,1]]></employers>
          </group>
        </workgroups>
        <emprights><![CDATA[45,236,1]]></emprights>
      </template>
    </param>
  </function>
</client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя (целое положительное число);

**name** – название шаблона (строка);

**is\_post\_sysacts** – может принимать два значения: 1 – писать системные действия из подзадачи в главную задачу; 0 – не писать системные действия из подзадачи в главную задачу;

**resppersons** – ID ответственных исполнителей (целое положительное число);

**employers** – ID исполнителей (целое положительное число);

**emprights** – ID пользователей, которым назначают права на создаваемый шаблон (целое положительное число).

2.7.15.2. **getIWGForTemplate** – получение списка изолированных групп в шаблоне

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getIWGForTemplate">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="idemployer">2000</param>
    <param name="idtemplate">544</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя, который делает запрос (целое положительное число);

**idtemplate** – ID шаблона, список изолированных точек по которому требуется получить (целое положительное число).

2.7.16. Функции для работы с шаблонами карточек документов

2.7.16.1. **getDocumentTemplateList** – получение списка шаблонов карточек документов

Пример входного XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="getDocumentTemplateList">
```

```
<param name="login">ivanov</param>
<param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
<param name="idemployer">2000</param>
</function>
<client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**idemployer** – ID пользователя (целое положительное число).

#### 2.7.17. Функции для работы с полномочиями пользователей

##### 2.7.17.1. setEmpPermission - назначение полномочий на пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<document>
  <function name="setEmpPermission">
    <param name="login">admin</param>
    <param name="pass">21232f297a57a5a743894a0e4a801fc3</param>
    <param name="employer">1</param>
    <param name="id_boss">20054</param>
    <param name="id_employer">20050</param>
    <param name="enabled">1</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**id\_boss** - владелец полномочий, id;

**id\_employer** - пользователь, на которого назначены полномочия, id (список пользователей указывать нельзя);

**enabled** - значение полномочия (0 - нет прав, 1 - полные права, 2 - только свои задачи и null для удаления), если не указать значение enabled - удаление полномочия;

**is\_manual** - системное или вручную назначенное полномочие, необязательный параметр, по умолчанию 1.

#### 2.7.17.2. getEmpPermission - возврат полномочий на пользователя

Пример входного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <function name="getEmpPermission">
    <param name="login">admin</param>
    <param name="pass">21232f297a57a5a743894a0e4a801fc3</param>
    <param name="employer">1</param>
    <param name="id_boss">20054</param>
    <param name="id_employer">20055</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>
```

Описание входных параметров:

**id\_boss** - владелец полномочий, id;

**id\_employer** - пользователь, на которого назначены полномочия, id (список пользователей указывать нельзя).

Пример выходного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <info>
    <api_version>1.0</api_version><time>11:41:59</time>
    <execution_time>0.13336896896362</execution_time>
  </info>
```

```
<result value="0">
  <enabled>2</enabled>
  <is_manual>1</is_manual>
  <postauthor>1</postauthor>
</result>
</document>
```

Описание выходных параметров:

**enabled** - значение права;

**is\_manual** - признак назначения права пользователем;

**postauthor** - пользователь, который право назначил.

.

2.7.17.3. getEmpPermissionMultiData - возврат полномочий на нескольких пользователей

Пример входного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <function name="getEmpPermissionMultiData">
    <param name="login">admin</param>
    <param name="pass">21232f297a57a5a743894a0e4a801fc3</param>
    <param name="employer">1</param>
    <param name="id_boss">20054,20051</param>
    <param name="id_employer">20055,20050,20047</param>
  </function>
  <client name="SomeProgram" version="1.0" />
</document>
```

Описание входных параметров:

**id\_boss** - владелец полномочий, id;

**id\_employer** - пользователь, на которого назначены полномочия, id (можно указывать id одного или нескольких пользователей, через запятую).

## Пример выходного XML:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <info>
    <api_version>1.0</api_version><time>11:35:23</time>
    <execution_time>0.13065791130066</execution_time>
  </info>
  <result value="0">
    <p1>
      <id_boss>20054</id_boss>
      <emps>
        <p1>
          <id_employer>20055</id_employer>
          <enabled>2</enabled>
        </p1>
        <p2>
          <id_employer>20050</id_employer>
          <enabled>1</enabled>
        </p2>
        <p3>
          <id_employer>20047</id_employer>
          <enabled>0</enabled>
        </p3>
      </emps>
    </p1>
    <p2>
      <id_boss>20051</id_boss>
      <emps>
        <p1>
          <id_employer>20055</id_employer>
```

```
<enabled>0</enabled>
</p1>
<p2>
  <id_employer>20050</id_employer>
  <enabled>0</enabled>
</p2>
<p3>
  <id_employer>20047</id_employer>
  <enabled>0</enabled>
</p3>
</emps>
</p2>
</result>
</document>
```

#### 2.7.18. API для делегирования полномочий заместителю

Использование REST API на примере функционала делегирования полномочий (см. п. [Работа с Rest API](#)).

Наибольшую сложность представляют фильтры замещения, имеющие иерархическую структуру. Представлены следующие функции:

- getListSubstitutors** - получить список заместителей пользователя с фильтрами;
- deleteSubstitutors** - удалить переданных заместителей у пользователя;
- updateSubstitutors** - добавить / изменить заместителей и фильтры.

#### getListSubstituto

Получение заместителей у пользователя. Например, с id 37820:

**GET** <https://localhost/rest/getListSubstitutors?employerId=37820>

Параметр **employerId** передан в строке запроса.

Ответ:

```
{
  "success": true,
```

```

"result": {
  "data": [
    {
      "id": 22587,
      "name": "Маслов Игорь Николаевич",
      "dep_name": "Организация арбитражных управляющих",
      "suid_id_original": 22587,
      "descr": "добавил 2",
      "enabled": 1,
      "id_type": 4,
      "active_in_absence": 0,
      "type": "folder",
      "children": [
        {
          "id": 1393,
          "object_id": 1393,
          "filterid": 12164,
          "object_type": 100,
          "name": "Попов",
          "tree_filter": [
            {
              "expressionid": null,
              "logicaloperation": 5,
              "children": [
                {
                  "fieldid": 85,
                  "fieldname": "Адресат",
                  "realfield": "Адресат",
                  "operation": 2,
                  "value": "Попов Иван Сергеевич",
                  "logicaloperation": 0,
                  "children": null,
                  "expanded": true,
                  "editable": false,
                  "leaf": true,
                  "not_supported": 0
                },
                {
                  "fieldid": "",
                  "fieldname": "",
                  "realfield": 6,
                  "operation": "",
                  "value": "",
                  "logicaloperation": 6,
                  "children": [
                    {
                      "fieldid": 53,
                      "fieldname": "Название задачи",
                      "realfield": "",
                      "operation": 2,
                      "value": "test1",
                      "logicaloperation": 0,
                      "children": null,

```



```

        "not_supported": 0
      }
    ],
    "expanded": true,
    "leaf": false
  }
],
  "id_parent": 1194
}
],
},
{
  "id": 22584,
  "name": "Селюков Евгений Олегович",
  "dep_name": "Управление приставов",
  "suid_id_original": 22584,
  "descr": "Пристав",
  "enabled": 1,
  "id_type": 4,
  "active_in_absence": 0,
  "type": "folder",
  "children": []
}
],
},
"request": "getListSubstitutors"
}

```

Общая структура ответа:

**"success": true** - запрос обработан успешно,

**"result". "data"** - данные ответа,

**"request": "getListSubstitutors"** - вызываемая функция.

В случае списка заместителей будет получен список заместителей, где помимо атрибутов, указан перечень фильтров, фильтры замещения для каждого из них. Все в соответствии с web-интерфейсом настроек замещения. Аналогичная структура данных используется и при создании / изменении настроек в **updateSubstitutors**.

## deleteSubstitutors

Пример. Удаление заместителя с id 22584 у пользователя 37820:

**POST https://localhost/rest/deleteSubstitutors {"employerId": 37820, "substIds": [22584]}**

Параметры:

**employerId** - пользователь,

**substIds** - array, список id удаляемых заместителей.

Ответ:

```
{
  "success": true,
  "result": {
    "data": [
      22584
    ]
  },
  "request": "deleteSubstitutors"
}
```

В ответе возвращен перечень удаленных заместителей.

### **updateSubstitutors**

Для изменения нужно подготовить структуру данных.

Информация о заместителе задается атрибутами:

**id** - id заместителя,

**id\_type** - тип делегирования полномочий (Администрирование / Типы делегирования полномочий (id)),

**enabled** - активность ( 0 — отключен, 1 — активный ),

**active\_in\_absence** - действует в мое отсутствие (0 - нет (по умолчанию), 1 - да),

**descr** - описание (необязательно),

**children** - список фильтров, определяющих объекты замещения.

Структура фильтра:

**object\_id** - id объекта-сущности, для которого задается фильтр

**filter\_action** - действие с фильтром (0 - добавить, 1 - изменить, 2 - удалить)

**object\_action** - действие с объектом (0 - добавить, 1 - изменить, 2 - удалить)

**object\_type** - int <тип объекта-сущности, для которого задается фильтр> (100 - заместители), необязательно

**filterid** - id фильтра (для существующих записей)

**name** - наименование

**tree\_filter** - непосредственно иерархическая структура фильтра

Иерархия фильтра состоит из условий и групп, в которые объединяются условия.  
Атрибуты:

- logicaloperation** - логическая операция группы (5 - И, 6 - ИЛИ, 7 - НЕ),
- fieldid** - id поля (Определяется по метаданным задачи или документа),
- fieldname** - отображаемое в фильтре наименование поля,
- operation** - операция:
  - 2- равно,
  - 3- не равно,
  - 4- больше,
  - 5- меньше
  - 6- больше или равно,
  - 7- меньше или равно,
  - 8- содержит,
  - 9- не содержит,
  - 10- начинается с
  - 11- заканчивается на,
- value** - значение для условия.

Пример:

Создание заместителя с id 37767 у пользователя 37820, фильтр по полю задачи с id 85 типа "Пользователь". Условие "Исполнитель" = "Михаил Иванович Ломов"

**POST** <https://localhost/rest/updateSubstitutors>

Параметры:

```
{
  "employerId": 37820,
  "data": [
    {
      "id": 37767,
      "descr": "Исполнитель Ломов",
      "enabled": 1,
      "id_type": 4,
      "active_in_absence": 0,
      "type": "folder",
      "children": [
        {
```



```

        "value": "Михаил Иванович Ломов"
    }
}
}
}
}
}
}
}
}
},
"request": "updateSubstitutors"
}

```

В интерфейсе настройки заместителей добавленная запись будет выглядеть как фильтр по полю **Исполнитель** равно *"Михаил Иванович Ломов"*. Другие фильтры задаются в окне параметров замещения интерфейса Системы. Подробное описание процедуры назначения заместителей см. в п. «Делегирование полномочий» «Руководства пользователя».

#### 2.7.19. saveColorSettings – сохранение цветовых настроек пользователя

Пример входного XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<document>
  <function name="saveColorSettings">
    <param name="login">ivanov</param>
    <param name="pass">c4ca4238a0b923820dcc509a6f75849b</param>
    <param name="settings_value"><![CDATA[Проект2]]></param>
    <param name="idemployer">155</param> <!--ID пользователя-->
    <param name="empset_ID">200</param>
  </function>
  <client name="SomeProgram" version="1.0"/>
</document>

```

Описание входных параметров:

**login** – логин пользователя (строка);

**pass** – пароль пользователя (строка);

**settings\_value** – название цветового профиля пользователя (строка);

**idEmployer** – ID пользователя, который сохраняет цветовые настройки (целое положительное число);

**empset\_ID** – ID в **EmployerSettings** (целое положительное число).

#### 2.7.20. Создание объектов в системе «Мотив» на основе данных, вводимых на стороннем сайте

Использование API-функции позволяет создавать объекты в системе «Мотив» на основе данных, вводимых пользователем на стороннем сайте. Рассмотрим пример создания в системе «Мотив» задачи, с заданными в HTML-форме значениями полей «**Название**» и «**Описание**» (рис.1.).

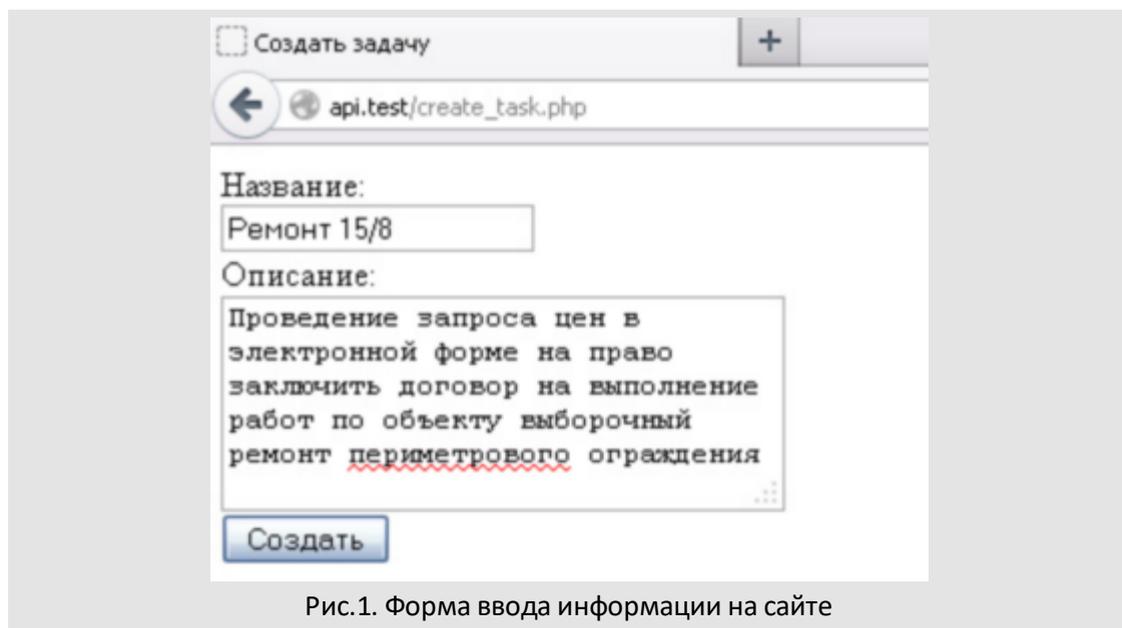


Рис.1. Форма ввода информации на сайте

Исходный код формы:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Создать задачу</title>
</head>
```

```
<body>
<!--Форма с входными параметрами-->
    <form name="F1" method=POST action="<? echo $_SERVER['PHP_SELF'] ?>"
ENCTYPE="multipart/form-data">
    Название:
    <br>
    <input type="text" name="task_name">
    <br>
    Описание:
    <br>
    <textarea rows='5' cols='30' name="task_descr"></textarea>
    <br>
    <input type=submit value="Создать">
</form><br>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    // название и
    $task_name = $_POST['task_name'];
    // описание задачи
    $task_descr = $_POST['task_descr'];

    // текст xml-запроса – (сразу строковое представление)
    // Также можно использовать класс DOMDocument или другие доступные
    // способы. Главное – правильно указать необходимые для вызова Api
    // параметры. Кодировка по умолчанию – utf-8
    $api_xml_text =
    '<?xml version="1.0" encoding="UTF-8"?>
<document>
<function name="addtask">
```

```
<param name="login">Kalina</param>
<param name="pass">202cb962ac59075b964b07152d234b70</param>
<param name="idTaskInitiator">20036</param>
<param name="idRealAuthor">20036</param>
<param name="iddepartment">14</param>
<param name="isOnlyForView">0</param>
<param name="parentId">0</param>
<param name="isresolution">0</param>
<param name="idemployer">20036</param>
<param name="idProject">73</param>
<param name="taskname"><![CDATA['.$task_name.']]></param>
<param name="description"><![CDATA['.$task_descr.']]></param>
<param name="priority">0</param>
<param name="authorsids"><![CDATA[20036]]></param>
<param name="groupIds"><![CDATA[20054]]></param>
<param name="idtasktype">17</param>
<param name="dictrecords">
<onedictvalue>
<fieldid>77</fieldid>
<values>
<value><![CDATA[1812]]></value>
<value><![CDATA[1813]]></value>
<value><![CDATA[1814]]></value>
</values>
</onedictvalue>
<onedictvalue>
<fieldid>78</fieldid>
<values>
<value><![CDATA[10000]]></value>
</values>
</onedictvalue>
```

```

<onedictvalue>
  <fieldid>79</fieldid>
  <values>
    <value><![CDATA[Рассмотрение предложений]]></value>
  </values>
</onedictvalue>
</param>
<param name="iswithreport">1</param>
<param name="issecret">0</param>
</function>
<client name="demo_app" version="1.1"/>
</document>';

```

// готовим параметры запроса для вызова функции

```
$postData = array();
```

```
$postData['input_xml'] = $api_xml_text;
```

```
/**
```

```
* @param $url – url точки входа на сервере Motiw, http://<host>/api/functions.php
```

```
* @param $post – параметры вызова функции
```

```
* @return mixed – результат выполнения запроса (xml – строка)
```

```
*/
```

```
function callApi($url, $post){
```

```
    // инициализируем объект для выполнение POST-запроса к API – Мотив
```

```
    $ch = curl_init();
```

```
    // настраиваем параметры
```

```
    curl_setopt($ch, CURLOPT_URL, $url);
```

```
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```

```
    curl_setopt($ch, CURLOPT_POST, 1);
```

```
    // данные запроса
```

```
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
```

```

// выполняем запрос, и получаем ответ сервера в xml-формате
$result = curl_exec($ch);
curl_close($ch);
return $result;
}

// вызываем API – адрес – <host> + url точки входа
$result = callApi('http://motiw/api/functions.php', $postData);

// разбираем результат выполнения
$dom = new DOMDocument();
$dom->loadXML($result);

// В ответе сервера ищем стандартный тег 'result'
$nodes = $dom->getElementsByTagName('result');
$res_node = $nodes->length == 1 ? $nodes->item(0) : null;

// если 'result' = 0 – выполнение прошло успешно
if ($res_node && $res_node->getAttribute('value') == 0){
    // анализируем другие теги ответа, в данном случае – id задачи
    $nodes = $dom->getElementsByTagName('task_id');
    if ($nodes && $nodes->length > 0)
        echo 'Создана задача N'. $nodes->item(0)->nodeValue;
} else {
    echo 'Ошибка при вызове Api!';
    if ($res_node) // выводим код ответа функции
        echo 'Сервер вернул значение '. $res_node->getAttribute('value');
}

unset($dom);

```

```
}  
?>  
</body>  
</html>
```

После заполнения полей формы и нажатия кнопки «**Создать**» (см. рис. 1), будет подготовлен XML для API-функции **addTask** (подробнее о вызове функции `addTask` см. п.2.4.3 «Функции для работы с задачами») и выполнен запрос на сервер Мотив. После того, как ответ сервера будет проанализирован, в HTML-форме появится сообщение с номером созданной задачи.

Результатом выполнения описанной функции будет создание задачи в системе «Мотив». Помимо полей заполненных в HTML-форме, в задаче отобразятся поля, заполнение которых, прописано непосредственно в XML-запросе, например, ссылки на справочник или другие объекты системы.

### 3. Создание обработчиков

**Обработчики (плагины)** – это независимые модули, которые могут использовать все возможности Системы и добавлять новые интерфейсы. В основном, они используются для создания специфических отчётов, однако этим их использование не ограничено. Например, экспорт данных реализован через плагины. Помимо экспорта данных возможен обратный процесс – загрузка данных в Систему из сторонних источников (импорт). Для загрузки справочников в используется обработчик importXLS (см. п. 15.5.2. Загрузка записей в справочник (Импорт XLS) Руководства администратора).

Рассмотрим пример создания и установки плагина в Системе.

Общая последовательность шагов для внесения плагина в Систему:

1. Создать плагин.
2. Войти в Систему с правами администратора системы.
3. Зарегистрировать плагин в Системе, для этого:
  - 1) Войти в раздел **Администрирование** → **Обработчики**.
  - 2) В открывшемся окне на панели инструментов нажать кнопку **Добавить**.
  - 3) Заполнить поля формы.
    - Заполнить поля **Название** и **Описание**.
    - Выбрать выполняемый файл – модуль плагина.
    - Указать тип обработчика и результат его выполнения.
    - Указать дополнительные параметры обработчика.
    - В поле **Дополнительные файлы** добавляются необходимые файлы для работы плагина (для плагина exportRTF.php это файл соответствия; для плагина отчёта это может быть html-файл с формой ввода параметров).
    - Нажать кнопку **Сохранить**.
4. После этих действий плагин будет доступен в Системе.

Если в поле **Тип обработчика** выбрано значение «Произвольный обработчик», то плагин-отчёт будет доступен через пункт меню **Отчёты** → **Пользовательские отчёты**.

В качестве примера рассмотрим реализацию обработчика (плагина) exportRTF, который предназначен для выгрузки в MS Word данных из задач и документов согласно файлу соответствия, прикреплённому к обработчику.

**Пример:**

```
<?
//-----вспомогательные функции-----
//проверка наличия файла по URL
```

```

function file_exist_URL($fileUrl)
{
    $AgetHeaders = @get_headers($fileUrl);
    return (preg_match("|200|", $AgetHeaders[0])) ? TRUE : FALSE;
}

//убираем теги и спецсимволы
function remove_html($value)
{
    $tmp_value = str_replace('<br />', '\\line ', $value);
    $tmp_value = iconv("UTF-8","UTF-8", $tmp_value);
    $tmp_value = strip_tags($tmp_value);
    $tmp_value = htmlspecialchars_decode($tmp_value);
    return $tmp_value;
}
//-----конец вспомогательных функций-----
//-----описание и инициализация переменных-----
$ServerURL = "http://".$_POST['server']; //URL сервера
$Login = $_POST['login']; //логин
$Pass = $_POST['password']; //пароль
$Type = $_POST['type']; //тип того, откуда вызвали (task, doc)
$Employer = $_POST['user_id']; //ID пользователя

/*идентификаторы стандартных полей для задачи
-Имя задачи – TASKNAME
-Дата начала – STARTDATE
-Дата окончания – ENDDATE
-Проект – PROJECT
-Авторы – AUTHORS
-Ответственные руководители – RESPPERSONS
-Исполнители – WORKGROUP
-Контролеры – CONTROLERS- Инициатор – INITIATOR
-Постановщик – CREATOR
-Описание – DESCRIPTION
-Важность – PRIORITY*/

//массив количества значений для задачи или документа
$num_ident_value_array = array();
//массив идентификаторов и значений
$ident_value_array = array("ident" => array (), "value" => array ());
//массив стандартных xml тегов и сопоставленных им идентификаторов для задач
$ident_array = array("tag" => array (), "ident" => array ());

//подключаем класс CURL
include($_SERVER["DOCUMENT_ROOT"]."/api2/curl.class.php");
//-----конец описанию и инициализации переменных-----
//проверка наличия файла шаблона
$fn = $ServerURL.$_POST['files'][0]['path']; //путь к файлу
if (!file_exist_URL($fn)) $Type = 'wrong'; //если файла шаблона не существует, указываем $Type = 'wrong'

//определяем тип того, с чем работаем
switch ($Type)
{
    case 'task': //работаем с задачей
    {
        $num_step = count($_POST['id'][0]); //количество задач
    }
}

```

```

for ($i=0; $i<$num_step; $i++)
{
    $TaskID[$i] = $_POST['id'][0][$i]; //ID задачи
}
$numstepag = 0; //количество тегов для задач
for ($taskstep=0; $taskstep<$num_step; $taskstep++)
{
//-----входная xml для задачи-----
$fiel = ""; //выходная xml
$result = ""; //результат CURL запроса
$input_xml = '<?xml version="1.0" encoding="UTF-8"?>
<document>
<function name="gettaskinfo">
<param name="login">'.$Login.'</param>
<param name="pass">'.$Pass.'</param>
<param name="idTask">'.$TaskID[$taskstep].</param>
<param name="idEmployer">'.$Employer.'</param>
</function>
<client name="MApiC" version="1.0"/>
</document>';
//-----конец входной xml для задачи-----
//-----вызов скрипта, используя CURL-----
$curl = new CURL; //создаём переменную
$data["input_xml"] = $input_xml; //входная xml
$link = $ServerURL."/api/functions.php"; //URL функции, которой отправляем запрос
$result = $curl->post($link,$data); //делаем запрос
$fiel = (substr($result,$curl->HeaderSize)); //выходная xml
//-----
//-----разбор xml документа-----
$domdoc = new DOMDocument(); //создаём DOMDocument
$domdoc->loadXML($fiel); //загружаем xml
$Task_Type_ID = ""; //тип задачи
$root = $domdoc->documentElement; //document
$node = $root->firstChild; //info
$node = $node->nextSibling; //result
$node = $node->firstChild; //taskname
while($node){
    $key = false;
    switch ($node->nodeName){
    case 'identifier':{ //получаем идентификаторы
        $tmp_ident_node = $node;
        $tmp_ident_node = $tmp_ident_node->firstChild;
        $k=0;
        while($tmp_ident_node){
            $ident_array['tag'][$k] = $tmp_ident_node->nodeName;
            $ident_array['ident'][$k++] = $tmp_ident_node->nodeValue;
            $tmp_ident_node = $tmp_ident_node->nextSibling;
        }
    }
    case 'authors':{ //авторы
        $key = array_search ("authors", $ident_array['tag']);
    }
    break;
    case 'resppersons':{ //ответственные руководители
        $key = array_search ("resppersons", $ident_array['tag']);
    }
}
}

```

```

break;
case 'workgroup':{      //исполнители
    $key = array_search ("workgroup", $ident_array['tag']);
}
break;
case 'controler':{     //контролёры
    $key = array_search ("controler", $ident_array['tag']);
}
break;
case 'id_meta_diction':{ //id типа задачи
    $Task_Type_ID = $node->nodeValue;
}
break;
}
if ($key!==false){     //если тег найден
    array_push ($ident_value_array['ident'], $ident_array['ident'][$key]); //записываем идентификатор
    $tmpnode = $node; //сохраняем
    $tmpnode = $tmpnode->firstChild; //переходим к следующему тегу
    $TmpStr = ""; //значение
    while($tmpnode){
        $tmpnode1 = $tmpnode;
        $tmpnode1 = $tmpnode1->firstChild;
        while($tmpnode1){
            if ($tmpnode1->nodeName == 'name') $TmpStr .= ", ".$tmpnode1->nodeValue; //записываем имена
            $tmpnode1 = $tmpnode1->nextSibling;
        }
        $tmpnode = $tmpnode->nextSibling;
    }
    $numstpehtag++;
    $TmpStr = substr($TmpStr, 2, strlen($TmpStr)-2); //убираем ", "
    array_push ($ident_value_array['value'], iconv("UTF-8","UTF-8", $TmpStr)); //созраняем значение
}else{
    $key = array_search ($node->nodeName, $ident_array['tag']);
    if ($key!==false){ //если тег найден
        $numstpehtag++;
        array_push ($ident_value_array['ident'], $ident_array['ident'][$key]);
        if ($node->nodeName=='taskdesc') //если описание
        {
            array_push ($ident_value_array['value'], remove_html($node->nodeValue)); //вырезаем теги
        }
        else
            array_push ($ident_value_array['value'], htmlspecialchars_decode(iconv("UTF-8","UTF-8", $node->nodeValue))); //сохраняем и убираем спец-символы
        }
    }
    $node = $node->nextSibling;
}

//-----конец разбора xml документа-----
if ($Task_Type_ID != ""){ //если тип задачи определён
//-----входная xml для получения информации о задаче-----
    $fiel = "";
    $result = "";
    $input_xml = '<?xml version="1.0" encoding="UTF-8"?>
<document>
<function name="taskTypeGetFields">

```

```

<param name="login">'. $Login.'</param>
<param name="pass">'. $Pass.'</param>
<param name="idemployer">'. $Employer.'</param>
<param name="taskid">'. $TaskID[$taskstep].</param>
<param name="tasktypeid">'. $Task_Type_ID.'</param>
</function>
<client name="MApiC" version="1.0"/>
</document>;

```

```

//-----
//-----вызов скрипта, используя CURL-----

```

```

$curl = new CURL;
$data['input_xml'] = $input_xml;
$link = $ServerURL."/api/functions.php";
$result = $curl->post($link,$data);
$fiel = (substr($result,$curl->HeaderSize));

```

```

//-----
//-----разбор xml документа-----

```

```

$domdoct = new DOMDocument();
$domdoct->loadXML($fiel);
$root = $domdoct->documentElement;
$node = $root->firstChild;
$node = $node->nextSibling;
$node = $node->firstChild;
if ($node){
    while ($node->nodeName != 'fields'){ //переход к тегу fields
        $node = $node->nextSibling;
    }
    $node = $node->firstChild;
    while ($node){
        $type_id = 0;
        $node = $node->firstChild;
        $tmpnode = $node;
        $IdentTmpT = "";
        $ValueTmpT = "";
        while ($tmpnode){
            switch ($tmpnode->nodeName){
                case 'idtype': //id типа
                    $type_id = $tmpnode->nodeValue;
                    break;
                case 'identifier': //идентификатор
                    $IdentTmpT = iconv("UTF-8","UTF-8", $tmpnode->nodeValue);
                    break;
                case 'values':{ //значение
                    $tmpnode2 = $tmpnode;
                    switch ($type_id){
                        case 8:{ //ссылка
                            $tmpnode2 = $tmpnode2->firstChild;
                            $tmpnode2 = $tmpnode2->firstChild;
                            $tmpnode2 = $tmpnode2->nextSibling;
                            $ValueTmpT = iconv("UTF-8","UTF-8", $tmpnode2->nodeValue);
                        }
                    }
                    break;
                case 9:{ //мн. ссылка
                    $tmp_str_vl = "";
                    $tmpnode2 = $tmpnode2->firstChild;
                    while ($tmpnode2){

```

```

        $tmpnode3 = $tmpnode2;
        $tmpnode3 = $tmpnode3->firstChild;
        $tmpnode3 = $tmpnode3->nextSibling;
        $tmp_str_vl .= iconv("UTF-8","UTF-8", $tmpnode3->nodeValue).';';
        $tmpnode2 = $tmpnode2->nextSibling;
    }
    $ValueTmpT = substr($tmp_str_vl, 0, strlen($tmp_str_vl)-2);
}
break;
default: $ValueTmpT = iconv("UTF-8","UTF-8", $tmpnode->nodeValue);
break;
}
}
break;
}
$tmpnode = $tmpnode->nextSibling;
}
if ($IdentTmpT != ""){ //если идентификатор есть
    $numsteptag++;
    array_push ($ident_value_array['ident'], $IdentTmpT); //идентификатор
    array_push ($ident_value_array['value'], htmlspecialchars_decode($ValueTmpT)); //значение
}
$node = $node->parentNode;
$node = $node->nextSibling;
}
}
}
//-----конец разбора xml документа-----
}
$num_ident_value_array[$taskstep] = $numsteptag; //сохраняем количество записей для данной задачи
}
}
break;
case 'doc':{ //работаем с документом
    $num_step = count($_POST['id'][0]); //количество документов
    for ($i=0; $i<$num_step; $i++){
        $DocID[$i] = $_POST['id'][0][$i]; //ID документа
    }
    $numsteptag = 0; //количество тегов для документов
    for ($docstep=0; $docstep<$num_step; $docstep++){
//-----входная xml для документа-----
        $input_xml = '<?xml version="1.0" encoding="UTF-8"?>
<document>
<function name="getDocumentInfo">
<param name="login">'.$Login.'</param>
<param name="pass">'.$Pass.'</param>
<param name="iddocument">'.$DocID[$docstep].</param>
<param name="idEmployer">'.$Employer.'</param>
</function>
<client name="MApiC" version="1.0"/>
</document>';
//-----
//-----вызов скрипта, используя CURL-----
        $curl = new CURL;
        $data['input_xml'] = $input_xml;
        $link = $ServerURL."/api/functions.php";
        $result = $curl->post($link,$data);

```

```

$fiel=(substr($result,$curl->HeaderSise));
//-----
//-----разбор xml документа-----
$domdocd=new DOMDocument();
$domdocd->loadXML($fiel);
$root=$domdocd->documentElement;
$node=$root->firstChild;
$node=$node->nextSibling;
$node=$node->firstChild;
while($node->nodeName!='fields'){
    $node=$node->nextSibling;
}
$node=$node->firstChild;
while($node){
    $node=$node->firstChild;
    $tmpnode=$node;
    $IdentTmpT="";
    $ValueTmpT="";
    $FieldTypeTmpT="";
    while($tmpnode){
        switch($tmpnode->nodeName){
            case 'fieldid': //идентификатор
                $IdentTmpT=iconv("UTF-8","UTF-8",$tmpnode->nodeValue);
                break;
            case 'fieldtype': //тип поля
                $FieldTypeTmpT=$tmpnode->nodeValue;
                break;
            case 'linked_documents'://документы
                $tmp_node=$tmpnode;
                $tmp_node=$tmp_node->firstChild;
                while($tmp_node){
                    $tmp_node2=$tmp_node->firstChild;
                    while($tmp_node2->nodeName!='name'){
                        $tmp_node2=$tmp_node2->nextSibling;
                    }
                    if($tmp_node2->nodeValue!="")
                        $ValueTmpT.=iconv("UTF-8","UTF-8",$tmp_node2->nodeValue).';';
                    $tmp_node=$tmp_node->nextSibling;
                }
                $ValueTmpT=substr($ValueTmpT,0,strlen($ValueTmpT)-2);
                break;
            case 'dictionary_values'://словарь
                $tmp_node=$tmpnode;
                $tmp_node=$tmp_node->firstChild;
                while($tmp_node){
                    if($tmp_node->nodeValue!="")
                        $ValueTmpT.=iconv("UTF-8","UTF-8",$tmp_node->nodeValue).';';
                    $tmp_node=$tmp_node->nextSibling;
                }
                $ValueTmpT=substr($ValueTmpT,0,strlen($ValueTmpT)-2);
                break;
            case 'value': //значение
                if($FieldTypeTmpT==3){
                    $ValueTmpT=remove_html($tmpnode->nodeValue);
                }else
                    $ValueTmpT=iconv("UTF-8","UTF-8",htmlspecialchars_decode($tmpnode->nodeValue));
        }
    }
}

```

```

        break;
    }
    $tmpnode = $tmpnode->nextSibling;
}
if ($IdentTmpT != ""){
    $numsteptag++;
    array_push ($ident_value_array['ident'], $IdentTmpT);
    array_push ($ident_value_array['value'], $ValueTmpT);
}
$node = $node->parentNode;
$node = $node->nextSibling;
}
$num_ident_value_array[$docstep] = $numsteptag;    //сохраняем количество записей для данной задачи
//-----конец разбора xml документа-----
}
}
break;
}
//-----заполнение шаблона-----
$fn1 = "OutFile.rtf";           //имя выходного файла
$content = "";                  //его содержимое
$fd = fopen ($fn1, "w");        //создаём файл
if (($Type != 'wrong') && ($_POST['files'][0]['path'] != "")) { //если файл шаблона найден
    $content = file_get_contents($fn); //читаем его содержимое
    //и начинаем заполнять шаблон путём замены идентификаторов на их значение
    $start_pos = strpos($content, "\pard"); //находим позицию первого появления/вхождения в строке
    $tmpstrlen = strlen($content)-1; //длина текста-1
    $text = substr($content, $start_pos, $tmpstrlen-$start_pos); //возвращает текст со $start_pos и до конца-1
    $before_text = substr($content, 0, $start_pos); //возвращает текст с начала и до $start_pos
    $tmp_before_text = $before_text; //сохраняем начальный текст
    $tmp_text = $text; //сохраняем конечный текст
    for ($k=0; $k<$num_ident_value_array[0]; $k++){
        $tmp_text = str_replace ($ident_value_array['ident'][$k], $ident_value_array['value'][$k], $tmp_text); //заменяем
    }
    $tmp_before_text = $tmp_before_text.$tmp_text; //добавляем текст к $before_text
    for ($j=0; $j<$num_step-1; $j++){
        $tmp_text = $text; //сохраняем
        for ($k=$num_ident_value_array[$j]; $k<$num_ident_value_array[$j+1]; $k++){
            $tmp_text = str_replace ($ident_value_array['ident'][$k], $ident_value_array['value'][$k], $tmp_text);
        }
        $tmp_before_text = $tmp_before_text."\page \line ".$tmp_text;//новый текст $text на новой странице
    }
    $content = $tmp_before_text.""; //закрываем
} else $content = "RTF template file not found!"; //иначе выдаём сообщение, что шаблон не найден
fwrite ($fd, $content); //записываем в файл
fclose ($fd); //закрываем файл
//заголовки
header ("Cache-Control: private");
header ("Cache-Control: must-revalidate, post-check=0, pre-check=0");
header ("Content-Type: application/rtf"); //тип возвращаемого файла, в модуле он проверяется
header ("Content-Length: " . filesize($fn1));
header ("Content-Disposition: attachment; filename=pluginanswer.rtf");

readfile($fn1); //возвращаем модулю содержимое файла
//-----
?>

```

### 3.1. Создание выполняемых файлов для обработчиков типа «Вычисление значений полей»

В создаваемом файле должен определяться один класс в пространстве имен **MotivPlugin\Calculation**, расширяющий **baseCalculation**. Класс должен содержать статический метод **getOperations**, возвращающий данные о реализованных вычислениях.

Если указанный в файле-сценарии класс определен в другом обработчике вычислений, на экран будет выведено сообщение об ошибке со ссылкой на этот обработчик.

