IntraService API. Версия 4.47.1

Оглавление

Введение

IntraService API выполнен по стандартам REST и предназначен для решения следующих задач:

- API является серверной частью для мобильных приложений под iPhone, Android. Сами приложения для мобильных устройств разрабатываются IntraVision.
- Для реализации модулей «Личный кабинет» на корпоративных сайтах. Следующий функционал: создать новую заявку, посмотреть список заявок, добавить комментарий по имеющейся заявке.
- Для интеграции с другими информационными системами.

  API не предполагает покрывать весь функционал IntraService.
  Более того, на данный момент не все планируемые функции в API реализованы. Список нереализованных, но запланированных в API функций доступен в разделе ниже.

Общая спецификация

Авторизация

В IntraService API используется **базовая** авторизация.
В заголовке каждого запроса передается логин и пароль пользователя IntraService.
Запросы выполняются в соответствии с полномочиями указанного пользователя.

Подробнее о базовой авторизации

Базовая авторизация представляет собой отправку заголовка **«Authorization»** со значением вида «Basic *<base64UserLoginPass>*», где *<base64UserLoginPass>* - закодированная в base64 строка вида «<login>:<password>».

Пример использования

Есть пользователь с логином *ivanov* и паролем *myivanov23*. Для авторизации нам нужна строка вида «*ivanov:myivanov23*».Используя функцию по переводу строки в Base64 получаем, что:
ivanov:myivanov23 = aXZhbm92Om15aXZhbm92MjM=
Посылаем заголовок в следующем виде:
Authorization: Basic aXZhbm92Om15aXZhbm92MjM=

Общий вид и типы запроса.

Общий вид запроса: http(s)://{site}/api/{resource}/{id}

Операция, выполняемая в запросе, определяется **типом запроса**:

- GET – получение
- POST – создание
- PUT – изменение
- DELETE – удаление (в настоящее время не реализовано)

Для запросов типа GET и POST часть **{id} (идентификатор)** является необязательной.
Для GET-запросов, если параметр {id}:

- отсутствует, то возвращается коллекция ресурсов;
- присутствует, то в ответе возвращается конкретный ресурс с указанным id.

Для PUT запросов достаточно отсылать только те поля, которые изменяются.

Поддерживаемые форматы

Для получения данных (GET) можно указать формат: **json** или **xml**. Используемый формат определяется по стандартным параметрам в заголовке запроса.
При отправке запросов, *для передачи данных* (POST, PUT) используется формат ***json***.

Ответ на запрос и возврат ошибки

В ответе на запрос возвращается стандартный код состояния HTTP.
Коды 2XX соответствуют успешному выполнению запроса, коды 4XX-5XX – ошибка.

При выполнении POST и PUT запросов в результате успешного выполнения в большинстве случаев возвращается также созданный\измененный объект, по аналогии с вызовом получения определенного ресурса (метод GET с параметром id).

Пример **успешно выполненного** запроса:

Получим сервис с идентификатором 1:

GET /api/service/1

Ответ в json:

Accept: application/json
Status: 200 OK
{
"Id": 1,
"Code": "SRVC",
"Name": "Service",
"Description": "Описание сервиса",
"IsArchive": false,
"IsPublic": false,
"ParentId": 185,
"Path": "185|1|"
}

Все запросы, завершившиеся **неудачей**, возвращают помимо ответа Bad Request также и json-объект с указанием ошибки.
Пример такого ответа для PUT-запроса:

{"Message":"The request is invalid.","
MessageDetail":"The parameters dictionary contains a null entry for parameter 'id' of non-nullable type 'System.Int32' for method 'IntraService.API.Domain.Models.Views.UserView Get(IntraService.Security.IIntraServicePrincipal, Int32)' in 'IntraService.Web.API.Controllers.UserController'. An optional parameter must be a reference type, a nullable type, or be declared as an optional parameter."}

Ограничение числа получаемых полей

Для запросов конкретных ресурсов и их коллекций может быть ограничено число получаемых полей ресурса. Для этого используется параметр **fields**, в котором через запятую перечисляются

названия требуемых полей. В разделе «Детализация по ресурсам» для каждого ресурса определены возможные поля.
Пример вызова:
GET /api/service?fields=Id,Name,Code

Поиск и фильтрация коллекции ресурсов

Если у ресурса есть возможность использования поиска или фильтра, то это указано в ее детальном описании в разделе «Детализация по ресурсам» - по каким полям осуществляется поиск и фильтрация.

Особенности вывода коллекции ресурсов

Коллекция ресурсов возвращается **постранично**.
В выводе коллекции ресурсов присутствует блок **Paginator** со следующими полями

Название	Описание	Использование в запросе
Count	Общее число ресурсов в коллекции	
Page	Номер отображаемой страницы (по умолчанию 1)	+
PageCount	Количество страниц	
PageSize	Количество ресурсов на странице (по умолчанию 25, не больше 2000)	+
CountOnPage	Возвращенное количество ресурсов на странице	

Вы можете задать размер страницы параметром **pagesize** и номер – параметром **page**.

*Пример запроса и ответа в **json**:*
Вывести список заявок постранично, на странице по 10 записей. Взять 2ую страницу, т.е. записи с 11 по 20:
GET /api/task?page=2&pagesize=10

```
{
  "Tasks": [{
    "Assets": "", …}]
  "Paginator": {
    "Count": 111786,
    "Page": 2,
    "PageCount": 11179,
    "PageSize": 10,
    "CountOnPage": 10
  }
}
```

Мультиязычность и даты.

Мультиязычные строковые данные возвращаются в соответствии с настройкой языка у пользователя.
Даты возвращаются в соответствии с часовым поясом пользователя. Соответственно при передаче дат в API, они должны быть указаны в часовом поясе пользователя.

> *Часовой пояс может быть определен для Пользователя, Подразделения и в Общих настройках. Приоритет имеет значение, выставленное для Пользователя. Если для Пользователя не определено, то наследуется от Подразделения (в том числе и вверх по иерархии подразделений). Если и для Компании не определено, то используется значение, заданное в Общих настройках.*

Получить информацию о часовом поясе и языке текущего (авторизованного) пользователя можно, используя специальный запрос - Получение языка и смещение времени для текущего пользователя

Данные в заголовке HTTP-запросов

Передача имени устройства и версии системы в заголовках

При разработке мобильных приложений в заголовках всех запросов к API можно передавать следующую информацию:
Device-Name - имя устройства (например, "iOS iPhone5")
Device-Version - версия клиентского ПО (для мобильных приложений - версия приложения, например, "1.04").
Если в заголовках присутствует эта информация, то в системе IntraService можно будет отследить, с какого устройства производились изменения в заявке.

Версия API

В заголовок каждого ответа вставляется заголовок (header) **X-API-Version**, значение которого показывает текущую версию API.

Например: *X-API-Version: 4.31 beta*

Заявка

/api/task
/api/newtask

Метод GET. Получение списка заявок, определенной заявки и шаблона новой заявки

Поля заявки для получения

Название	Тип данных	Используется в search	Описание
Список заявок, конкретная заявка			
Id	Int	+	Идентификатор заявки. Присваивается автоматически при создании

PriorityId	Int		Идентификатор приоритета
StatusId	Int		Идентификатор статуса
ServiceId	Int		Идентификатор сервиса
ServiceStageId	Int		Идентификатор этапа
ServiceStage	String		Название этапа
ParentId	Int		Идентификатор родительской заявки
Name	String	+	Название
Description	String	+	Описание
Deadline	DateTime		Срок
Created	DateTime		Дата создания
Changed	DateTime		Дата изменения
CreatorId	Int		Идентификатор заявителя
Creator	String		Заявитель
EditorId	Int		Идентификатор пользователя, последним изменившего заявку
ExecutorIds	String		Идентификаторы исполнителей через запятую
Executors	String	+	Список исполнителей
Files	String	+	Список имен файлов. Внимание! Устарело. Будет удалено. Используйте поле FileNames
FileNames	String		Список имен файлов. Разделитель – ||
FileIds	String		Идентификаторы файлов
CategoryIds	String		Идентификаторы категорий через запятую. В списке только при указании в параметре **include**.
Categories	String	+	Список категорий (через || на карточной форме)
Hours	Decimal		Трудозатраты в часах
Price	Decimal		Трудозатраты в деньгах
Assets	String	+	Список наименований активов (через || на карточной форме)
AssetIds	String		Идентификаторы активов через запятую. В списке только при указании в параметре **include.**

ReactionDate	DateTime		Время реакции
ReactionDateFact	DateTime		Фактическое время реакции
ReactionOverdue	Bool		Срок реакции истек?
Closed	Date		Дата закрытия
TypeId	Int		Идентификатор типа
Type	String		Тип
EvaluationId	Int		Идентификатор оценки заявки
Evaluation	String		Оценка заявки (название)
IsMassIncident	Bool		Массовый инцидент?
CreatorIP	String		Ip заявителя
ExecutorGroupId	Int		Идентификатор группы исполнителей
ExecutorGroup	String		Название группы исполнителей
TaskRepeatRuleId	Int		Идентификатор правила повторения
ResolutionDateFact	DateTime		Фактическое время выполнения
ResolutionOverdue	Bool		Срок выполнения истек?
Coordinators	String	+	Список согласующих
IsCoordinatedForCoordinators	String	+	Перечисление признаков согласования (True, False) для списка согласующих. Порядок соответствует порядку в Coordinators и CoordinatorIds.
CoordinatorIds	String		Идентификаторы согласующих
Observers	String	+	Список наблюдателей
ObserverIds	String		Идентификаторы наблюдателей
Data	xml	+	Значение дополнительных полей заявки в виде xml
Конкретная заявка			
Field{id}	String		Значение дополнительного поля с идентификатором id. Например, Field8.
CompletionStatus	Int		Выполнено, %
WorkflowId	Int		Идентификатор бизнес процесса
IsClient	Boolean		Тип роли пользователя на сервисе заявки: true – клиент, false - исполнитель

FieryFields	String		Список атрибутов, при изменении которых возможно изменение других атрибутов заявки. При изменении этих атрибутов необходимо получать измененные атрибуты с помощью GET метода /api/task?field=…
IsConcurrence	Bool		Если true, то заявка находится в статусе согласование.
IsCoordinatedByCurrentUser	Bool		Если значение True, то текущий пользователь согласовал заявку
CreatorADGuid	Guid		Идентификатор Заявителя в Active Directory
CreatorComments	string		Поле «Описание» для Заявителя
CreatorCompanyId	int		Идентификатор компании Заявителя
CreatorCompanyName	string		Название компании заявителя
CreatorCompanyPath	string		Иерархия для компании заявителя
CreatorCurrentLanguage	string		Язык Заявителя
CreatorEmail	string		E-mail Заявителя
CreatorIsArchive	bool		Является ли Заявитель архивным пользователем
CreatorLogin	string		Логин Заявителя
CreatorMobilePhone	string		Мобильный телефон Заявителя
CreatorPhone	string		Телефон Заявителя
CreatorPosition	string		Должность заявителя
CreatorRoleId	int		Идентификатор роли Заявителя
CreatorTimeZone	string		Часовой пояс Заявителя
PriorityDescription	string		Описание текущего приоритета заявки
PriorityImage16Url	string		Ссылка на иконку текущего приоритета заявки
PriorityName	string		Название текущего приоритета заявки
ServiceCode	string		Код сервиса
ServiceDescription	string		Описание сервиса
ServiceIsArchive	bool		Сервис архивый?
ServiceIsPublic	bool		Сервис публичный?

ServiceName	string		Название сервиса
ServiceParentId	int		Идентификатор родительского сервиса для сервиса заявки
ServicePath	string		Иерархия сервиса
StatusName	string		Название текущего статуса
StatusDescription	string		Описание текущего статуса
StatusImage16Url	string		Ссылка на иконку текущего статуса малого размера
StatusImage24Url	string		Ссылка на иконку текущего статуса большого размера
StatusIsCommentRequired	bool		Для текущего статуса комментарий обязателен?
StatusIsFixed	bool		Текущий статус с признаком «Заявка выполнена»?
StatusIsFinal	bool		Текущий статус с признаком «Конечный»?

Получение списка заявок

Вызов

GET
/api/task?serviceid={serviceid}&fields={fieldList}&search={searchString}&archive={value}&inactive={value}&sort={sortFields}&filterid={filter_id}&{filterFields}&include={includeList}&pagesize={value}&page={value}

При выводе списка заявок учитываются настройки роли на сервисе для текущего пользователя по видимости заявок.

Возможные параметры

Все параметры являются необязательными.

serviceid *– идентификатор сервиса для фильтрации списка заявок*

Если не задан – то выведет заявки всех доступных сервисов
Пример: Выведем все доступные пользователю заявки на сервисе 3
/api/task?serviceid=3

fields *- перечень полей ресурса «Заявка» для вывода (из таблицы, поля для списка)*

Например:
Выведем заявки в виде 3х полей: Номер, Название, идентификатор статуса
/api/task?fields=Id,Name,StatusId

search *– строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке, а также по всем комментариям заявки.*

Поиск действует по тем же правилам, как и в web-части.

archive *– позволяет вернуть заявки архивных сервисов*

Возможные значения:

- **False** – возвращает заявки только не архивных сервисов (значение по умолчанию, если параметр не задан)

- **True** – возвращает заявки как архивный, так и неархивных сервисов

Например:

Получим заявки из архивных и неархивных сервисов в виде списка из номера заявки и идентификатора пользователя заявителя и его имени:
/api/task?fields=Id,CreatorId,Creator&archive=true

inactive *– позволяет вернуть заявки неактуальных сервисов*

Возможные значения:

- **False** – возвращает заявки только актуальных сервисов (значение по умолчанию, если параметр не задан)
- **True** – возвращает заявки как актуальных, так и неактуальных сервисов

Например:

Получим заявки всех актуальных и неактуальных сервисов, доступных пользователю
/api/task?inactive=true

filterid *- список заявок и полей заявки возвращаются в соответствии с фильтром с идентификатором* ***{filter_id}***

Например:

Получим список заявок с использование фильтра «Мои Заявки», который имеет идентификатор 45
/api/task?filterid=45

sort *– позволяет получить отсортированный список заявок*

Сортировка, заданная этим параметром, имеет более высокий приоритет, чем сортировка, заданная фильтром.

Сортировка по полю задается в виде: «***FieldName {asc|desc}***», где FieldName – название поля заявки, а **asc** или **desc** указывает на то, что сортируем соответственно либо по возрастанию, либо по убыванию.

Пример:

Выведем списком заявок для сервиса 31 и отсортируем его сначала по статусу по убыванию, а затем по приоритету
/api/task?serviceid=31&sort=StatusId desc, PriorityId asc

{filterFields} *– поля заявки и значения для фильтрации по ним. Подробнее в разделе о фильтрации заявок*

pagesize, page *– параметры отображения списка*

include *- позволяет получить связанные с заявками ресурсы в одном запросе.*

Возможные значения:

- **ASSETIDS** – позволяет вывести поле AssetIds (идентификаторы активов, прикрепленных к заявке) в списке для каждой заявки.
- **CATEGORYIDS** - позволяет вывести поле CategoryIds (идентификаторы категорий, прикрепленных к заявке) в списке для каждой заявки.
- **PRIORITY** – включает в ответ помимо списка заявок <Tasks> блок <Priorities>, который содержит более подробную информацию о всех приоритетах заявок, полученных в списке.
- **SERVICE** – включает в ответ помимо списка заявок <Tasks> блок <Services>, который содержит более подробную информацию о всех сервисах заявок, полученных в списке.

- **STATUS** – включает в ответ помимо списка заявок <Tasks> блок <Statuses>, который содержит более подробную информацию о всех статусах заявок, полученных в списке.
- **USER** – включает в ответ помимо списка заявок <Tasks> блок <Users>, который содержит более подробную информацию о Заявителях, Исполнителях, Наблюдателях и Согласующих на заявках, полученных в списке.

Подробнее использование данного параметра на примере будет рассмотрено ниже.

Пример вывода

1) Пример запроса с параметром ***include***:
   GET /api/task?include=status,priority
   Ответ:

```
{
 "Tasks": [
        {
         "StatusId": 7,
         "PriorityId": 12,
         …
        },
        {
         "StatusId": 8,
         "PriorityId": 12,
         …
        }
 ],
 "Statuses": [
        {
         "Id": 7,
         "Name": "Открыта",
         …
        },
        {
         "Id": 8,
         "Name": "В работе",
         …
        }
 ],
 "Priorities": [
        {
         "Id": 12,
         "Name": "Средний",
         …
        }
 ]
 "Paginator": [
        {
         "Count": 111786,
         "Page": 1,
         …
        }
 ]
```

```
}
```

2) Пример вывода в xml

GET: /api/task?fields=Id,Name,ServiceId&include=service&pagesize=1&page=2

```
<TaskList>
<Tasks>
        <Task>
                <Id>158</Id>
                <Name>test category</Name>
                <ServiceId>16</ServiceId>
        </Task>
</Tasks>
<Services>
        <Service>
                <Id>16</Id>
                <Code>c2</Code>
                <Name>сервис 2</Name>
                <Description/>
                <IsArchive>False</IsArchive>
                <IsPublic>False</IsPublic>
                <ParentId/>
                <Path>16|</Path>
        </Service>
</Services>
<Paginator>
        <Count>11</Count>
        <Page>2</Page>
        <PageCount>11</PageCount>
        <PageSize>1</PageSize>
        <CountOnPage>1</CountOnPage>
</Paginator>
</TaskList>
```

Фильтрация списка заявок по полям

Поля для фильтрации

Поле фильтрации	Тип передаваемого значения	Описание фильтрации
ChangedLessThan	DateTime	Дата последнего изменения заявок меньше или равна указанной
ChangedMoreThan	DateTime	Дата последнего изменения заявок больше или равна указанной
Changed	DateTime	Дата последнего изменения заявок равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
CreatedLessThan	DateTime	Дата создания заявок меньше или равна указанной
CreatedMoreThan	DateTime	Дата создания заявок больше или равна указанной

Created	DateTime	Дата создания заявок равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
ClosedLessThan	DateTime	Дата закрытия заявок меньше или равна указанной
ClosedMoreThan	DateTime	Дата закрытия заявок больше или равна указанной
Closed	DateTime	Дата закрытия заявок равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
DeadlineLessThan	DateTime	Плановый срок выполнения заявок меньше или равен указанному
DeadlineMoreThan	DateTime	Плановый срок выполнения заявок больше или равен указанному
Deadline	DateTime	Плановый срок выполнения заявок равен указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
ReactionDateLessThan	DateTime	Плановый срок реакции заявок меньше или равен указанному
ReactionDateMoreThan	DateTime	Плановый срок реакции заявок больше или равен указанному
ReactionDate	DateTime	Плановый срок реакции заявок равен указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
ReactionDateFactLessThan	DateTime	Фактический срок реакции заявок меньше или равен указанному
ReactionDateFactMoreThan	DateTime	Фактический срок реакции заявок больше или равен указанному
ReactionDateFact	DateTime	Фактический срок реакции заявок равен указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
ResolutionDateFactLessThan	DateTime	Фактический срок выполнения заявок меньше или равен указанному
ResolutionDateFactMoreThan	DateTime	Фактический срок выполнения заявок больше или равен указанному
ResolutionDateFact	DateTime	Фактический срок выполнения заявок равен указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
ResolutionOverdue	Bool	Заявки, срок выполнения которых истек
ReactionOverdue	Bool	Заявки, срок реакции которых истек
TypeIds	String	Заявки в указанных типах. Перечисление идентификаторов типов заявок (Id) через запятую.

StatusIds	String	Заявки в указанных статусах. Перечисление идентификаторов статусов (Id) через запятую.
PriorityIds	String	Заявки с указанными приоритетами. Перечисление идентификаторов приоритетов (Id) через запятую.
ServiceIds	String	Заявки с указанными сервисами. Перечисление идентификаторов сервисов (Id) через запятую.
EditorIds	String	Заявки, у которых последний изменивший пользователь входит в указанный набор пользователей. Перечисление идентификаторов пользователей (Id) через запятую.
CreatorIds	String	Заявки, у которых заявитель входит в указанный набор пользователей. Перечисление идентификаторов пользователей (Id) через запятую.
ExecutorIds	String	Заявки, на которые назначен хотя бы один исполнитель из заданного перечня пользователей. Перечисление идентификаторов пользователей (Id) через запятую.
ExecutorGroupIds	String	Заявки, на которые назначена одна из групп исполнителей из заданного перечня групп исполнителей. Перечисление идентификаторов групп исполнителей (Id) через запятую.
ObserverIds	String	Заявки, на которые назначен хотя бы один наблюдатель из заданного перечня пользователей. Перечисление идентификаторов пользователей (Id) через запятую.
CategoryIds	String	Заявки, на которые назначена хотя бы одна категория из заданного перечня категорий. Перечисление идентификаторов категорий (Id) через запятую.
AssetIds	String	Заявки, на которые назначен хотя бы один актив из заданного перечня активов. Перечисление идентификаторов активов (Id) через запятую.

Примеры запросов.

1) Выведем все заявки, которые изменялись после 13 ноября 2015 года 10:00 и умеют статус с Id 1 или 7:
   GET /api/task?ChangedMoreThan=2015-11-13 10:00&StatusIds=1,2

2) Выведем заявки, у которых есть хотя бы одна категория с Id из следующих: 18, 22, 10 и на заявке пользователь с Id 215 является одним из исполнителей:
GET /api/task?CategoryIds=18,22,10&ExecutorIds=215

Получение конкретной заявки

Вызов

/api/task/taskid?include={includeList}

Где ***taskid*** **–** идентификатор (номер) заявки

Возможные параметры

include *- позволяет получить связанные с заявкой ресурсы в одном запросе.*

Возможные значения:

- **PRIORITY** – включает в ответ помимо заявки <Task> блок <Priorities>, который содержит более подробную информацию о приоритете заявки
- **SERVICE** – включает в ответ помимо заявки <Task> блок <Services>, который содержит более подробную информацию о сервисе заявки.
- **STATUS** – включает в ответ помимо заявки <Task> блок <Statuses>, который содержит более подробную информацию о статусе заявки.
- **USER** – включает в ответ помимо заявки <Task> блок <Users>, который содержит более подробную информацию о Заявителе, Исполнителях, Наблюдателях и Согласующих на заявке.
- **TASKTYPE** - включает в ответ помимо заявки <Task> блок <TaskType>, который содержит более подробную информацию о типе заявки (включая информацию о дополнительных полях).
- **TASKTYPESETTINGS** - включает в ответ помимо заявки <Task> блок <TaskTypeSettings>, который содержит значения для бизнес-настроек, задающихся в типе заявки
- **USERTASKRIGHTS** - включает в ответ помимо заявки <Task> блок <UserTaskRights>, который описывает полномочия пользователя по работе с заявкой. Подробнее в разделе.

Пример вывода в xml

GET /api/task/159?include=status,priority

```
<TaskForm>
    <Task>
        <AssetIds>42</AssetIds>
        <Assets>Office 12</Assets>
        <Categories>Аппаратная ошибка</Categories>
        <CategoryIds>22</CategoryIds>
        <Changed>27.11.2015 12:54:05</Changed>
        <Closed/>
        <CompletionStatus/>
        <CoordinatorIds/>
        <Coordinators/>
        <Created>26.11.2015 16:18:06</Created>
        <Creator>Администратор</Creator>
        <CreatorADGuid/>
        <CreatorComments/>
```

```
                .....
                .....
        </Task>
        <Statuses>
                <Status>
                        <Id>31</Id>
                        <Name>Открыта</Name>
                        .....
                </Status>
        </Statuses>
        <Priorities>
                <Priority>
                        <Id>9</Id>
                        <Name>Средний</Name>
                        .....
                </Priority>
        </Priorities>
</TaskForm>
```

Получение шаблона заявки для создания (значения по умолчанию)

Вызов

GET /api/newtask?serviceid={serviced}&tasktypeid={tasktypeid}&include={includeList}

Возможные параметры

serviceid *– обязательный параметр. Идентификатор сервиса, в котором создается заявка.*

tasktypeid *– обязательный параметр. Идентификатор типа заявки*

include *– необязательный, позволяет получить связанные с заявкой ресурсы в одном запросе.*

Возможные значения:

Аналогично, как и для получения конкретной заявки

Полномочия пользователя по работе с заявкой

Поля ресурса UserTaskRights

Название	Тип	Описание
Name	Int	Название
Description	Int	Описание
Priority	Int	Приоритет
Deadline	Int	Срок исполнения
Categories	Int	Категории
CompletionStatus	Int	Выполнено
ServiceStage	Int	Этап
Files	Int	Файлы
Assets	Int	Активы
ExecutorGroup	bool	Группа исполнителей. Видимость блока
Observers	bool	Наблюдатели. Видимость блока
Executors	bool	Исполнители. Видимость блока
Creator	bool	Заявитель. Видимость блока

Lifetime	bool	Жизненный цикл. Видимость блока
Comment	Int	Комментарий
ChangeService	Boolean	Можно менять сервис и тип заявки?
ChangeCreator	Boolean	Можно изменять заявителя заявки?
Status	int	Статус
SystemFields	Int	Системные поля
ViewHistory	Boolean	Можно просматривать историю изменений?
CanBeExecutor	Boolean	Может быть исполнителем заявки?
AssignItselfExecutor	Boolean	Можно назначить себя исполнителем?
AssignExecutors	Boolean	Можно назначить исполнителей?
RemoveExecutors	IDictionary<int,bool>	Можно удалять исполнителей? Словарь <Идентификатор пользователя, можно удалять?>
AssignExecutorGroup	Boolean	Можно назначать группу исполнителей?
RemoveExecutorGroup	Boolean	Можно удалять группу исполнителей?
HasExecutorGroup	Boolean	На сервисе есть группы исполнителей. Если true то показываем блок группы исполнителей, в противном случае не показываем.
AssignItselfObserver	Boolean	Можно назначать себя наблюдателем?
AssignObservers	Boolean	Можно назначать наблюдателей?
RemoveObservers	Boolean	Можно удалять наблюдателей?
AssignCoordinators	Boolean	Можно назначать согласующих?
RemoveCoordinators	Boolean	Можно удалять согласующих?
TaskTypeFieldsData	Object	Полномочия для дополнительных полей
MassIncident	TaskFieldRights	Массовый инцидент
Hours	Int	Трудозатраты
Rate	Int	Стоимость трудозатрат
EditOthersExpenses	Boolean	Можно редактировать чужие трудозатраты (часы и стоимость)?
RemoveExpenses	IDictionary<int,bool>	Можно удалить трудозатраты? Словарь<идентификатор трудозатрат, можно удалять?>
ToStatuses	List<TaskStatusSimpleView>	Список объектов TaskStatusSimpleView. Содержит статусы, в которые можно перевести заявку
ToPriorities	List<TaskPrioritySimpleView>	Список объектов TaskPrioritySimpleView. Содержит

		приоритеты, которые можно установить для заявки
ToServiceStages	List<ServiceStageSimpleView>	Список объектов ServiceStageSimpleView. Содержит этапы, которые можно установить для заявки
CategoriesWithAutoAssignUsers	List<AutoAssignsTaskCreateView>	Список объектов AutoAssignsTaskCreateView. Содержит категории, по которым возможны автоназначения при создании заявки, и правила автоназначения. Поле не возвращается при получении полномочий по существующей заявке
StatusesWithAutoAssignUsers	List<AutoAssignsStatusChangeView>	Список объектов AutoAssignsStatusChangeView. Содержит статусы, по которым возможны авто назначения при изменении статуса заявки, и правила авто назначения.
ViewExpenses	Boolean	Если true, то показываем блок трудозатрат.

Полномочия по работе с полями заявки рассчитываются как сумма следующих чисел:
0 – нет прав
1 – разрешено редактировать
2 – разрешено только чтение
4 – поле является обязательным

Поля объекта AutoAssignsStatusChangeView ресурса UserTaskRights.

Объект содержит возможные авто назначения пользователей на заявку при переводе в данный статус

Название	Тип	Описание
Id	Int	Id статуса
Name	String	Название статуса
DeleteExecutors	Boolean	Снимать текущих исполнителей по заявке
DeleteObservers	Boolean	Снимать текущих наблюдателей на заявке
ExecutorIds	String	Идентификаторы исполнителей через запятую
Executors	String	Имена исполнителей через запятую
ExecutorGroupId	Int	Id группы исполнителей
ExecutorGroup	String	Название группы исполнителей
ObserverIds	String	Идентификаторы наблюдателей через запятую
Observers	String	Имена наблюдателей через запятую

Поля объекта TaskStatusSimpleView ресурса UserTaskRights.

Название	Тип	Описание
Id	Int	Id статуса
Name	String	Название статуса
IsExternal	Boolean	Статус является внешним.
IsConcurrence	Boolean	Согласование

Поля объектов TaskPrioritySimpleView ресурса UserTaskRights.

Название	Тип	Описание
Id	Int	Id приоритета
Name	String	Название приоритета

Поля объекта ServiceStageSimpleView ресурса UserTaskRights.

Название	Тип	Описание
Id	Int	Id этапа
Name	String	Название этапа
EndDate	Date	Дата окончания этапа

Поля объекта AutoAssignsTaskCreateView ресурса UserTaskRights.

Объект содержит возможные автоназначения пользователей на заявку при выборе данной категории. Правила действуют только при создании заявки.

Название	Тип	Описание
Id	Int	Id категории
Name	String	Название категории
ExecutorIds	String	Идентификаторы исполнителей через запятую
Executors	String	Имена исполнителей через запятую
ExecutorGroupId	Int	Id группы исполнителей
ExecutorGroup	String	Название группы исполнителей
ObserverIds	String	Идентификаторы наблюдателей через запятую
Observers	String	Имена наблюдателей через запятую

Получение атрибутов заявки при изменении других атрибутов.

Вызов

GET
/api/task/field={field}&oldworkflowid={oldworkflowid}&serviced={serviceid}&priorityid={priorityid}&statusid={statusid}&taskid={taskid}&creatorid={creatorid}&tasktypeid={tasktypeid}&assetids={assetids}&categoryids={categoryids}

Примеры использования:

1) При изменении атрибутов заявки возможно изменение бизнес процесса и/или класса обслуживания, что в свою очередь влияет на изменение атрибутов заявки.

2) При изменении приоритета заявки возможно изменение срока выполнения заявки (в соответствии с классом обслуживания).

Возможные параметры

field, oldworkflowid, serviced, priorityid, statusid *– обязательные;*

taskid, creatorid, tasktypeid, assetids, categoryids *– необязательные.*

Описание параметров запроса

Название	Тип	Описание
field	Int	Какое поле изменилось? 1 – приоритет 2 – заявитель 3 – актив 4 – категория
oldworkflowid	Int	Идентификатор текущего бизнес процесса
serviceid	Int	Новый идентификатор сервиса
priorityid	Int	Новый идентификатор приоритета
statusid	Int	Новый идентификатор статуса
taskid	Int	Идентификатор заявки (если идет редактирование заявки)
creatorid	Int	Новый заявитель
tasktypeid	Int	Новый тип заявки
assetids	Int	Новые идентификаторы активов через запятую
categoryids	Int	Новые идентификаторы категорий через запятую

Результат запроса

Возвращается следующая структура пересчитанных полей

Название	Тип	Описание
Deadline	DateTime?	Срок
ReactionDate	DateTime?	Планируемый срок реакции
StatusId	Int	Идентификатор статуса
Statuses	IdName[]	Идентификаторы и имена статусов, в которые можно перевести заявку
WorkflowId	Int	Идентификатор бизнес процесса

Метод POST. Создание заявки

Поля для создания заявки

Название	Тип данных	Обязательность	Описание
Name	String	+	Название
ServiceId	int	+	Идентификатор сервиса
StatusId	Int	+	Идентификатор статуса
Comment	String		Комментарий
Deadline			Срок выполнения

Description	String		Описания
ParentId	Int		Идентификатор родительской заявки
PriorityId	Int	+	Идентификатор приоритета
TypeId	Int	+	Идентификатор типа заявки
CreatorId	Int		Идентификатор заявителя (если есть права на создание заявки от другого лица)
ServiceStageId	Int		Идентификатор этапа сервиса
IsPrivateComment	bool		Скрыть комментарий от клиента?
IsMassIncident	bool		Массовый инцидент?
CompletionStatus	Int		Выполнено, %
AssetIds	String		Идентификаторы активов через запятую
CategoryIds	String		Идентификаторы категорий через запятую
ExecutorIds	String		Идентификаторы исполнителей через запятую
CoordinatorIds	String		Идентификаторы согласующих через запятую. Описание перевода в согласование.
ExecutorGroupId	int		Идентификатор группы исполнителей
ObserverIds	String		Идентификаторы наблюдателей через запятую
CoordinatorIds	String		Идентификаторы согласующих через запятую
FileTokens	String		Идентификаторы временных файлов через “,”
DeletedFiles	String		Идентификаторы фалов для удаления через “|” (например, 1|2|3…)
Field{id}	String	+ (только обязательные)	Значение дополнительного поля с идентификатором id. Например, Field8.

Создание заявки

Вызов

POST /api/task

В теле запроса передаются поля объекта Task (сам объект Task передавать не надо). Допускается передавать только те поля, которые содержат значения.
Запрос возвращает все поля созданной заявки (аналогично запросу «Получение заявки»).

Для создания заявки рекомендуется получить объект Task со значениями полей по умолчанию с помощью операции получения шаблона заявки, изменить значения требуемых полей и выполнить операцию create.

Создание заявки от лица, которое идентифицируется электронным адресом

Вызов

POST /api/task

Для того, чтобы создать заявку от лица пользователя, который идентифицируется электронным адресом необходимо в теле запроса передать дополнительные поля, а именно:

Название	Тип данных	Обязательность	Описание
UserEmail	string	+	Электронный адрес пользователя
UserPassword	string	+	Пароль пользователя (если создаем от пользователя, которого нет в системе)
UserConfirmPassword	string	+	Подтверждение пароля
UserComments	String		Комментарий в карточке пользователя
UserCompanyId	int	+	Компания нового пользователя
UserCurrentLanguage	String		Язык нового пользователя
UserLogin	String		Логин нового пользователя (если пусто, возьмется UserEmail)
UserMobilePhone	String		Мобильный телефон нового пользователя
UserName	String		Имя нового пользователя (если пусто, возьмется UserEmail)
UserPhone	String		Телефон нового пользователя
UserPosition	String		Должность нового пользователя
UserRoleId	int	+	Системная роль нового пользователя
UserTimeZone	string		Часовой пояс нового пользователя

Поле **UserEmail** обязательно в любом случае.
Алгоритм создания заявки следующий:
если в POST запросе передается электронный адрес **UserEmail** пользователя, то проверяем наличие пользователя в системе с таким электронным адресом.
Если такой пользователь:

1) *есть* – заявка будет создана от лица этого пользователя
2) *нет* – система будет пытаться сначала создать самого пользователя, а потом уже заявку от его имени. В этом случае обязательно нужно передать также поля *UserPassword* и *UserConfirmPassword*, остальные поля – необязательно.

Примечание: данный способ хорош для одновременного создания нового пользователя в системе и заведения от его имени заявки.

Если вам нужна более упрощенная схема создания заявки от другого лица – используйте передачу поля CreatorId. В этом случае также учитываются права текущего пользователя на изменение заявителя: если таких нет, то заявителем останется текущий пользователь.

Метод PUT. Изменение заявки.

Поля заявки для изменения

Поля для изменения аналогичны, как и для создания. Также есть несколько полей, которые нельзя задать при создании, но можно изменить:

Название	Тип данных	Описание
EvaluationId	int	Идентификатор оценки
ReactionDate	DateTime	Плановый срок реакции
ReactionDateFact	DateTime	Фактический срок реакции
ResolutionDateFact	DateTime	Фактический срок выполнения
Coordinate	bool	Заявка согласована данным пользователем? (Только для заявок в статусе согласование)

Изменение заявки

Вызов

PUT /api/task/taskid
где taskid – идентификатор изменяемой заявки

Отслеживание изменений другим пользователем

Для отслеживания изменений другим пользователем в момент сохранения нужно передавать поле **Changed**, которое мы получаем в момент получения заявки.
Возможны 2 случае:

1) Передаем поле Changed. Тогда, если другой пользователь изменит заявку, то возникнет следующая ошибка:
   "NetworkError: 409 Conflict - /api/task/140134"
   140134
   {"Message":"Пользователь admin изменил заявку 26.12.2013 8:18, пока вы ее редактировали."}
2) Не передаем поле Changed. Тогда изменения предыдущего пользователя будут перезаписаны.

Прикрепление файла при создании или изменении заявки

Прикрепление файлов происходит в 2 этапа:

Сохранение файлов и получение уникальных идентификаторов

Для сохранения файлов отправляем запрос
POST /api/TaskFile
Параметрами запроса служат:

1) файлы (количество файлов не ограничено);
2) требуемое имя (Необязательный параметр).

Имя формируется следующим образом:

1) Файлы заявки – **«*{filekey}*Name»**:
2) Файлы в дополнительных полях – **«*{filekey}*TaskTypeFieldId»**

Имя файла должно быть без расширения.

Например:

```
<input name="file0" type="file"/>
<input name="file0Name" type="text"/>
<input name="file1" type="file"/>
<input name="file2" type="file"/>
<input name="file2Name" type="text"/>
.....
<input name="file2TaskTypeFieldId" type="text" value="15"/>
```

В ответе получаем ***{"FileTokens":"1,23"}***

Примечание:

В заголовках запроса необходимо указывать *content-type: multipart/form-data* и отсылать файлы соответственно этому заголовку (более подробно можно ознакомиться в документациях к языку программирования, на котором вы пишете).

Прикрепление сохраненных файлов к заявке

Полученные ране идентификаторы файлов ***FileTokens*** – передаем в запросе для создания или изменения заявки в поле **FileTokens.**

Согласование

Перевод или создание в статусе с признаком «Согласование»

При переводе заявки в статус с признаком «Согласование» или создании заявки в нем возможны 2 варианта:

1) Указаны **CoordinatorIds**, а также хотя бы один из указанных пользователей принадлежит сервису, текущий пользователь может изменять список согласующих и согласование является простым (не несколько согласований друг за другом). Если эти условия выполняются, то заявка перейдет в согласование с указанными согласующими CoordinatorIds. Иначе как в пункте 2.
Если у текущего пользователя есть права на изменение списка согласующих, то вы можете запросить список согласующих по умолчанию для данного статуса, чтобы он смог отредактировать данный список. Или же назначить любых других согласующих на заявку из списка пользователей, привязанных к сервису - для этого можно воспользоваться запросом списка наблюдателей, т.к. если пользователь может быть наблюдателем, он может быть и согласующим.
2) Не указаны **CoordinatorIds**, или они не привязаны к сервису, или у пользователя нет прав назначать согласующих, или согласований несколько друг за другом - тогда для каждого

согласования будет определяться список согласующих по умолчанию (по настройкам перехода в Бизнес-процессе).

Обращаем внимание:

1) В данной реализации если согласований несколько подряд – переданный список *CoordinatorIds* будет игнорироваться, для всех согласований в цепочке будет определяться список согласующих по умолчанию.
2) Если хотя бы для одного согласования не будет найден список согласующих по умолчанию - будет ошибка: «для согласования обязательно определение согласующих». Поэтому правильно настраивайте свой бизнес-процесс.

Согласование заявки

Вызов

PUT /api/task/taskid

Где **taskid** - идентификатор заявки в статусе с признаком «Согласование», для которой текущий пользователь хочет проставить признак, что заявка им согласована или несогласована.

Пример использования

1) Согласуем заявку № 143. Заявка находится в статусе «Согласование»:
   PUT /api/task/143
   {"Coordinate":true}
2) Заявку № 145 в статусе «Согласование» не будем согласовывать и оставим комментарий, почему мы отказали:
   PUT /api/task/145
   {"Coordinate": false, "Comment": "Причина отказа от согласования: документы составлены неверно"}

Прогресс согласования заявки

При вызове информации по конкретной заявке, если она находится в статусе согласования, вы можете посмотреть информацию о том, кто согласовал заявку на данный момент (актуально для настройки согласования «все», а не «хотя бы один из»).
В выводе присутствует:

1) Указание имен согласующих через запятую - **Coordinators**.
2) Указание идентификаторов согласующих через запятую - **CoordinatorIds**.
3) Указание признака согласовал ли согласующий заявку также через запятую - **IsCoordinatedForCoordinators**.
4) Согласована ли заявка текущим пользователем - **IsCoordinatedByCurrentUser**.

Порядок согласующих в полях **Coordinators**, **CoordinatorIds** и **IsCoordinatedForCoordinators** одинаков, поэтому вы можете определить кто согласовал заявку, а кто нет.

Например, вывод:
<CoordinatorIds>5304, 5094, 5224</CoordinatorIds>

<Coordinators>Иван Никифоров, Петр Иванов, Вячеслав Смирнов</Coordinators>
<IsCoordinatedForCoordinators>False, True, False</IsCoordinatedForCoordinators>

означает, что на данный момент:

1) согласующий «Иван Никифоров» имеет идентификатор в системе 5304 и он не согласовал заявку (значение False);
2) согласующий «Петр Иванов» имеет идентификатор в системе 5094 и он согласовал заявку (значение True);
3) согласующий «Вячеслав Смирнов» имеет идентификатор в системе 5224 и он не согласовал заявку (значение False).

Получение связанных ресурсов по заявке

Получение списка заявителей

Вызов

GET
/api/taskcreator?serviceid={serviceid}&fields={fieldList}&seach={searchString}&pagesize={value}&page={value}
Получить пользователей сервиса, которые могут быть заявителями на заявках этого сервиса.
Возвращает коллекцию объектов ресурса Пользователь (User).

Возможные параметры

serviceid *– обязательный. Идентификатор сервиса.*

fieldList *– список полей для получения (поля ресурса Пользователь)*

search *– строка поиска. Ищет в полях «Имя пользователя» и «Логин».*

pagesize, page *- параметры отображения списка*

Получение списка исполнителей

Вызов

GET
/api/taskexecutor?serviceid={serviceid}&fields={fieldList}&seach={searchString}&pagesize={value}&page={value}
Получить пользователей сервиса, которые могут быть исполнителями на заявках этого сервиса.
Возвращает коллекцию объектов ресурса Пользователь (User).

Возможные параметры

Аналогично, как и для получения списка заявителей

Получение списка групп исполнителей

GET
/api/taskexecutorgroup?serviceid={serviceid}&fields={fieldList}&seach={searchString}&pagesize={value}&page={value}
Получить все группы исполнителей, доступные для указанного сервиса.
Возвращает коллекцию объектов ресурса Группа исполнителей (ExecutorGroup)

Возможные параметры

serviceid *– обязательный. Идентификатор сервиса*

fieldList *– список полей для получения (поля ресурса Группа исполнителей)*

search *– строка поиска. Ищет в поле «Название группы исполнителей»*

pagesize, ***page*** *- параметры отображения списка*

Получение списка наблюдателей

Вызов

GET
/api/taskobserver?serviceid={serviceid}&fields={fieldList}&seach={searchString}&pagesize={value}&page={value}
Получить пользователей сервиса, которые могут быть наблюдателями на заявках этого сервиса.
Возвращает коллекцию объектов ресурса Пользователь (User).

Возможные параметры

Аналогично, как и для получения списка заявителей

Получение списка согласующих для перевода в статус согласование

Вызов

GET
/api/taskcoordinator?workflowid={wfid}&statusid={sid}&serviceid={servid}&creatorid={crid}
Получить список согласующих для перевода заявки в статус с признаком «Согласование» в соответствии с настроенными правилами перевода в Бизнес-процессе.
Возвращает коллекцию объектов ресурса Пользователь (User).

Возможные параметры

workflowid *– обязательный. Идентификатор Бизнес-процесса.*

statusid *– обязательный. Статус с признаком «согласование», в который переводим (если не согласование - будет пустая выборка в ответе)*

serviceid *– обязательный. Сервис. Нужен для определения пользователей сервиса, подходящих под правила.*

creatorid *– обязательный. Заявитель на заявке. Заявитель нужен при формировании списка согласующих, так как в БП есть настройка по функциональным ролям, которые почти все заточены на заявителя.*

Получение файла, привязанного к заявке

GET /api/taskfile/fileid
где ***fileid*** – идентификатор файла.
Идентификаторы файлов заявки указаны в поле ***FileIds*** при получении конкретной заявки.

Детализация по остальным ресурсам

Сервис

/api/service

Метод GET. Получение списка сервисов и конкретного сервиса

Поля сервиса для получения

Название	Тип	Описание
Id	Int	Идентификатор сервиса
Code	String	Код сервиса
Name	String	Название
Description	String	Описание
IsArchive	Bool	Сервис архивный?
IsPublic	Bool	Сервис публичный?
ParentId	Int	Идентификатор родительского сервиса
Path	String	Путь. Идентификаторы сервисов с корневого по текущий через разделитель “|”
CanCreateTask	Bool	Можно ли создать заявку на этом сервисе? Выводится только на списке сервисов при значении параметра FOR = CREATETASK
HasTaskTypes	Bool	Сервис содержит типы заявок? Выводится только на списке сервисов при значении параметра FOR = CREATETASK и только для тех сервисов, у которых CanCreateTask=true

Получение списка доступных сервисов

Вызов

GET:
/api/service?fields={fieldList}&for={value}&archive={value}&inactive={value}&pagesize={value}&page={value}
Список возвращается в отсортированном виде (по порядку сервисов и уровню иерархии).

Возможные параметры

Все параметры являются **необязательными**.

fields *- перечень полей ресурса «Сервис» для вывода (из таблицы)*

Например: Выведем все коды и называния сервисов
/api/service?fields=Code,Name

archive *- позволяет фильтровать список сервисов по признаку «Архивный»*

Возможные значения

- **false** – возвратит только не архивные сервисы (значение **по умолчанию**, если параметр не задан);
- **true** - возвратит как архивные, так и не архивные сервисы.

inactive *– позволяет отфильтровать список сервисов по признаку «Актуальный» (задается на вкладке «Главная»)*

Возможные значения

- **false** – получить только актуальные сервисы (значение **по умолчанию**, если параметр не задан);
- **true** - получить как актуальные сервисы, так и неактуальные.

for *–параметр для фильтрации списка сервисов*

Возможные значения:

- **Параметр не задан** – вернет список сервисов, если у текущего пользователя есть права на просмотр списка сервисов;
- **filtertasks** – возвращает список сервисов, на которые назначен пользователь (вместе с родительскими)
- **createtask** - возвращает список сервисов, на которые назначен пользователь (вместе с родительскими). Для каждого сервиса в выводе присутствуют поля *CanCreateTask* и *HasTaskTypes* (описание можно посмотреть в таблице)

page, pagesize *– параметры отображения таблицы*

Пример вывода в xml

1) GET: /api/service?fields=Id,Name,Code

```
<ServiceList>
    <Services>
        <Service>
            <Id>16</Id>
            <Code>c2</Code>
            <Name>сервис 2</Name>
        </Service>
        <Service>
            <Id>15</Id>
            <Code>c1</Code>
            <Name>Сервис1</Name>
        </Service>
    </Services>
    <Paginator>
        <Count>2</Count>
        <Page>1</Page>
        <PageCount>1</PageCount>
        <PageSize>25</PageSize>
        <CountOnPage>2</CountOnPage>
    </Paginator>
</ServiceList>
```

2) GET: /api/service?for=createtask&fields=Id,Name,Code

```
<ServiceList>
    <Services>
        <Service>
            <Id>16</Id>
            <Code>c2</Code>
            <Name>сервис 2</Name>
            <CanCreateTask>True</CanCreateTask>
            <HasTaskTypes>True</HasTaskTypes>
```

```
                </Service>
                <Service>
                        <Id>15</Id>
                        <Code>c1</Code>
                        <Name>Сервис1</Name>
                        <CanCreateTask>True</CanCreateTask>
                        <HasTaskTypes>True</HasTaskTypes>
                </Service>
        </Services>
        <Paginator>
                <Count>2</Count>
                <Page>1</Page>
                <PageCount>1</PageCount>
                <PageSize>25</PageSize>
                <CountOnPage>2</CountOnPage>
        </Paginator>
</ServiceList>
```

Получение определенного сервиса

Вызов

GET: /api/service/serviced
Где **serviceid** – идентификатор сервиса

Пример вывода в xml

GET: /api/service/15

```
<ServiceView>
        <Code>c1</Code>
        <Description>cc</Description>
        <Id>15</Id>
        <IsArchive>false</IsArchive>
        <IsPublic>false</IsPublic>
        <Name>Сервис1</Name>
        <ParentId i:nil="true"/>
        <Path>15|</Path>
</ServiceView>
```

Метод POST. Назначение пользователей на сервис

api/serviceuser

Поля для назначения пользователей на сервис

Поле	Тип значения	Обязательность	Комментарий
ServiceId	Целое число (int)	+	Идентификатор сервиса
AddUsers	ServiceUserView Array	+	Массив значений Пользователь-Роль (ServiceUserView), которых назначаем на сервис

Поля ServiceUserView

Поле	Тип значения	Обязательность	Комментарий
UserId	Целое число (int)	+	Идентификатор пользователя
RoleId	Целое число (int)		Идентификатор роли. Если значения нет, то пользователь добавится на сервис со своей системной ролью

Вызов и пример использования

POST: /api/serviceuser

Данный запрос доступен только пользователям, у системной роли которых есть полномочие *«Редактировать сервисы» -> «Все сервисы»*
Допускается передавать только те поля, которые содержат значения. Обязательно должны быть заполнены поля, отмеченные как обязательные.
Запрос возвращает код 201 в случае успеха.
Пример:

Добавим на сервис с идентификатором 35 трех пользователей: пользователя 21 с ролью, идентификатор которой равен 8, а также 2х пользователей – 25 и 81 – с их системной ролью.

{“ServiceId”: 35, “AddUsers”: [{“UserId”: 21, “RoleId”: 8}, {“UserId”: 25}, {“UserId”: 81}]}

Метод POST. Назначение компаний на сервис

api/servicecompany

Поля для назначения компаний на сервис

Поле	Тип значения	Обязательность	Комментарий
ServiceId	Целое число (int)	+	Идентификатор сервиса
CompanyIds	Строка (String)	+	Идентификаторы компаний через запятую

Вызов и пример использования

POST: /api/servicecompany

Данный запрос доступен только пользователям, у системной роли которых есть полномочие *«Редактировать сервисы» -> «Все сервисы»*
Допускается передавать только те поля, которые содержат значения. Обязательно должны быть заполнены поля, отмеченные как обязательные.
Запрос возвращает код 201 в случае успеха.
Пример:

Добавим на сервис с идентификатором 35 две компании с идентификаторами 29 и 341:

{“ServiceId”: 35, “CompanyIds”: “29,341”}

Фильтр

/api/filter

Метод GET. Получение списка фильтров

Поля фильтра для получения

Название	Тип	Описание
Id	Int	Идентификатор фильтра
Name	String	Название
IsCommon	Bool	Общий фильтр (да\нет)
IsDefault	Bool	Фильтр по-умолчанию (да\нет)

Получение списка фильтров

Вызов

GET:
/api/filter?resource={ resource _name}

Возможные параметры

resource - обязательный параметр

Содержит название ресурса, по которому возвращается список фильтров (task, user, service и т.п.).
Пример: Получим список фильтров для заявок
GET: /api/filter?resource=task

Пример вывода в xml

GET: /api/filter?resource=task

```
<ArrayOfFilterView>
    <FilterView>
        <Id>106</Id>
        <IsCommon>false</IsCommon>
        <IsDefault>true</IsDefault>
        <Name>1. Инциденты</Name>
    </FilterView>
    <FilterView>
        <Id>107</Id>
        <IsCommon>false</IsCommon>
        <IsDefault>false</IsDefault>
        <Name>2. Проблемы</Name>
    </FilterView>
<ArrayOfFilterView>
```

Статус

/api/taskstatus

Метод GET. Получение списка статусов

Поля статуса для получения

Название	Тип	Описание
Id	Int	Идентификатор статуса
Name	String	Название
Description	String	Описание
Image16Url	String	Url маленькой иконки
Image24Url	String	Url большой иконки
IsInitial	Bool	Признак «Начальный»
IsCommentRequired	Bool	Признак «Комментарии обязательны»
IsFinal	Bool	Признак «Конечный»
IsFixed	Bool	Признак «Заявка выполнена»
IsDeadlineAccountingPaused	Bool	Признак «Учет времени выполнения приостановлен»
IsConcurrence	Bool	Признак «Согласование»
IsExternal	Bool	Признак «Во внешней организации»
Changed	DateTime	Дата последнего изменения статуса

Получение списка статусов

Вызов:

GET:
/api/taskstatus?{filterFields}

Список возвращается в отсортированном виде.

Возможные параметры:

{filterFields} - поля для фильтрации

Все условия фильтрации склеиваются по условию И. Подробнее в разделе о фильтрации статусов

Пример вывода в xml
GET: /api/taskstatus

```
<ArrayOfTaskStatusView>
    <TaskStatusView>
        <Changed>2015-10-29T13:51:14.023</Changed>
        <Description/>
        <Id>31</Id>
        <Image16Url>http://localhost/IntraService4.40p2/img/statuses/actual16.png </Image16Url>
        <Image24Url>http://localhost/IntraService4.40p2/img/statuses/actual24.png </Image24Url>
        <IsCommentRequired>false</IsCommentRequired>
        <IsConcurrence>false</IsConcurrence>
        <IsDeadlineAccountingPaused>false</IsDeadlineAccountingPaused>
        <IsExternal>false</IsExternal>
        <IsFinal>false</IsFinal>
        <IsFixed>false</IsFixed>
        <IsInitial>true</IsInitial>
        <Name>Открыта</Name>
```

```
        </TaskStatusView>
        <TaskStatusView>
              .....
        </TaskStatusView>
</ArrayOfTaskStatusView>
```

Фильтрация списка статусов

Поля фильтрации

Поле фильтрации	Тип передаваемого значения	Описание фильтрации
ChangedLessThan	DateTime	Дата последнего изменения статусов меньше или равна указанной дате
ChangedMoreThan	DateTime	Дата последнего изменения статусов больше или равна указанной дате
Changed	DateTime	Дата последнего изменения статусов равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
IsInitial	bool	Статусы, у которых проставлен признак «Начальный»
IsFixed	bool	Статусы, у которых проставлен признак «Заявка выполнена»
IsFinal	bool	Статусы, у которых проставлен признак «Конечный»
IsCommentRequired	bool	Статусы, у которых проставлен признак «Комментарии обязательны»
IsConcurrence	bool	Статусы, у которых проставлен признак «Согласование»
IsExternal	bool	Статусы, у которых проставлен признак «Во внешней организации»
IsDeadlineAccountingPaused	bool	Статусы, у которых проставлен признак «Учет времени выполнения приостановлен»

Примеры запросов

1) Получим все статусы для согласования:
   GET /api/taskstatus?IsConcurrence=true
2) Получим все статусы с приостановленным временем выполнения и измененные 25 октября 2015 года.
   GET /api/taskstatus?IsDeadlineAccountingPaused=true&Changed=2015-10-25

Приоритет

/api/taskpriority

Метод GET. Получение списка приоритетов

Поля приоритета для получения

Название	Тип	Описание
Id	Int	Идентификатор приоритета
Name	String	Название
Description	String	Описание
Image16Url	String	Url иконки

Получение списка приоритетов

Вызов:

GET:
/api/taskpriority
Список возвращается в отсортированном виде.

Пример вывода в xml
GET /api/taskpriority

```
<ArrayOfTaskPriorityView>
    <TaskPriorityView>
        <Description/>
        <Id>11</Id>
        <Image16Url>http://localhost/IntraService4.40p2/img/priorities/priority2.gif </Image16Url>
        <Name>Низкий</Name>
    </TaskPriorityView>
    <TaskPriorityView>
        .....
    <TaskPriorityView>
</ArrayOfTaskPriorityView>
```

Компания

/api/company

Метод GET. Получение списка компаний и определенной компании.

Поля компании для получения

Название	Тип	Используется в search	Описание
Id	Int		Идентификатор компании
FullName	String		Полное наименование (с учетом иерархии)
Name	String	+	Наименование
Domain	String	+	Домен(ы)
ParentId	Int		Идентификатор родительского подразделения

Path	String		Путь. Идентификаторы подразделений с корневого по текущий через разделитель “|”
ServiceDeskPath	String		Путь к системе (в почтовых уведомлениях)
TypeId	int		Тип компании. 1 – Моя компания, 2 – Внешние подрядчики, 3 - Клиенты
IsArchive	bool		Признак архивный
ADGuid	String		Идентификатор компании в AD
CurrentLanguage	String		Язык
TimeZone	String		Часовой пояс
Address	String		Адрес
WEB	String	+	Адрес в интернете
Note	String		Примечание
Phone	String	+	Телефон
Email	String	+	Email
ManagerId	Int		Идентификатор руководителя
ContactPersonId	int		Идентификатор пользователя, который является контактным лицом
Changed	DateTime		Дата изменения компании
Data	xml		Значения дополнительных полей компании (поля Адрес, Примечание и Email выводятся также отдельными полями)

Получение списка компаний

Вызов:

GET:
/api/company?fields={fieldList}&search={searchString}&{filterFields}&pagesize={value}&page={value}

Возможные параметры:

fields - перечень полей ресурса «Компания» для вывода(из таблицы)

Например:

Выведем все названия и типы компаний списком
/api/company?fields=FullName, TypeId

search – *строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке.*

Например:

а) Выведем все компании, у которых в одном из полей для поиска содержится подстрока «Test»:
/api/company?search=Test

б) Выведем все компании, у которых в одном из полей для поиска содержится подстрока «Test», и вывод представим только 2мя полями для каждой компании – «Название» и «Тип»
/api/company?fields=FullName, TypeId&search=Test

page, pagesize *– параметры отображения таблицы*

{ filterFields } *поля для фильтрации*

Все условия фильтрации склеиваются по условию И. Подробнее в разделе о фильтрации компаний

Пример вывода в xml

GET: /api/company

```
<CompanyList>
        <Companies>
                <Company>
                        <Id>30</Id>
                        <Name>Центральный офис</Name>
                        .....
                        .....
                </Company>
                <Company>
                        <Id>91</Id>
                        <Name>Подразделение 1</Name>
                        ......
                        ......
                </Company>
        </Companies>
        <Paginator>
                <Count>2</Count>
                <Page>1</Page>
                <PageCount>1</PageCount>
                <PageSize>25</PageSize>
                <CountOnPage>3</CountOnPage>
        </Paginator>
</CompanyList>
```

Фильтрация списка компаний

Поля фильтрации

Поле фильтрации	Тип передаваемого значения	Описание фильтрации
ChangedLessThan	DateTime	Дата последнего изменения компании меньше или равна указанной дате
ChangedMoreThan	DateTime	Дата последнего изменения компании больше или равна указанной дате
Changed	DateTime	Дата последнего изменения компании равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
TypeIds	string	Компании, у которых тип равен одному из заданных: 1 – моя компания, 2 – внешний подрядчик, 3 – клиент. Перечисление типов через запятую.

ParentIds	string	Компании, у которых есть компания-родитель из указанного перечня компаний. Перечисление идентификаторов (Id) через запятую.
ContactPersonIds	string	Компании, у которых контактным лицом является один из указанных пользователей. Перечисление идентификаторов (Id) через запятую.
Name	string	Компании, в названии которых содержится указанная строка.
Path	string	Компании, путь которых начинается с указанного. Используется для поиска подразделения и всех его дочерних.
Email	string	Компании, email которых содержит указанную строку
Phone	string	Компании, телефон которых содержит указанную строку
Domain	string	Компании, в поле «Домен (ы)» которых содержится указанная строка
Note	string	Компании, для которых примечание содержит указанную строку.
ADGuid	guid	Компании, идентификатор в Active Directory которых равен указанному (должна быть одна компания в выводе)
IsArchive	bool	Компании, признак архивный у которых стоит в указанном положении

Примеры запросов:

1) Получим компании, домен которых сдержит «mycompany.ru» и которые не архивные:
   GET: /api/company?domain=mycompany.ru&isarchive=false
2) Получим компанию с идентификатором в AD «1617e62f-1c4e-496d-9819-31bc80487718»:
   GET: /api/company?adguid=1617e62f-1c4e-496d-9819-31bc80487718
3) Получим все компании типа «клиент», у которых контактным лицом является пользователь с идентификатором 50 или 43, а также информация о которых была изменена после 13 ноября 2015 года:
   GET: /api/company?typeids=3&contactpersonids=50,43&changedmorethan=2015-11-13

Получение определенной компании

Вызов:

GET: api/company/companyid

Где *companyid* – идентификатор компании

Пример вывода в xml:

```
<CompanyView>
        <Id>101</Id>
        <Name>Test company</Name>
        <FullName>Test company</FullName>
        <Domain/>
        <ServiceDeskPath>http://mycompany.intraservice.ru</ ServiceDeskPath >
        <IsArchive>False</IsArchive>
        <ParentId/>
        <Path>101|</Path>
        <ADGuid/>
        <CurrentLanguage/>
        <TimeZone/>
        <TypeId>3</TypeId>
        <Address/>
        <WEB>test web</WEB>
        <Note/>
        <Phone/>
        <Email/>
        <ContactPersonId/>
        <ManagerId>37</ManagerId>
        <Changed>29.09.2015 11:12:27</Changed>
</CompanyView>
```

Метод POST. Создание новой компании

Поля для создания компании

Поле	Тип значения	Обязательность	Комментарий
TypeId	Целое число (int)	+	Идентификатор типа компании
Name	string	+	Название компании
ServiceDeskPath	Строка (string)	+	Путь к системе (для почтовых уведомлений). Если не передается, по умолчанию возьмется путь из Общих настроек.
Domain	Строка (string)		Домены компании (через запятую)
ParentId	int		Идентификатор родительской компании
IsArchive	bool		Признак архивный
TimeZone	string		Часовой пояс
Address	string		Адрес
Email	string		Email
Note	string		Примечание

Phone	string		Телефон
WEB	string		Адрес сайта
CurrentLanguage	string		Язык
ContactPersonId	int		Контактное лицо (идентификатор пользователя)
Field{id}	string	Нет	Значение дополнительного поля с идентификатором, например, Field8

Вызов и пример использования

POST: /api/company

В теле запроса передается объект Company. Допускается передавать только те поля, которые содержат значения. Обязательно должны быть заполнены поля, отмеченные в таблице как обязательные.
Запрос возвращает все поля созданной компании (аналогично запросу Получение определенной компании

Пример:

Добавим компанию типа «Клиент» с именем «Тестовая компания»

{"TypeId": 3, "Name":"Тестовая компания"}

Метод PUT. Изменение существующей компании

Поля компании для изменения

Поля компании для изменения аналогичны, как и при создании (указаны в таблице).

Вызов и пример использования

PUT: /api/company/companyid
Где ***companyid*** – идентификатор компании

Пример:
Изменение компании с id = 123: изменение названия на «Тестовая компания изменение»,
PUT: /api/company/123

{"Name":"Тестовая компания изменение "}

Документ компании

/api/companydocument

Метод GET. Получение списка документов компаний и определенного документа.

Поля документа компании для получения

Название	Тип	Используется в search	Описание
Id	Int		Идентификатор документа
NameFromUser	String	+	Имя документа, заданное пользователем. Если имя не задавали, то совпадет с название файла
FileName	String	+	Название файла документа
CompanyFullName	String		Полное название компании, к которой относится документ
CompanyId	Int		Идентификатор компании, к которой относится документ
FileId	String		Идентификатор файла документа. Используется для получения данного файла.
TypeName	String		Название типа документа
TypeId	int		Тип документа: 1 – другое, 2 – договор.
IsActive	bool		Признак активности договора. Только для типа 2 – договор.
DateBegin	String		Начало действия договора. Только для типа 2 – договор.
DateEnd	String		Окончание действия договора. Только для типа 2 – договор.
EditorId	String		Идентификатор пользователя, последним изменившим документ.
EditorName	String		Имя пользователя, который последним изменял документ.
CreatorId	String		Идентификатор пользователя, создавшего документ.
CreatorName	String		Имя пользователя, создавшего документ.
Created	String		Дата создания документа.
Changed	String		Дата последнего изменения документа.

Получение списка документов

Вызов:

GET:
/api/companydocument?fields={fieldList}&search={searchString}&{filterFields}&pagesize={value}&page={value}

Возможные параметры:

fields - перечень полей ресурса «Документ компании» для вывода(из таблицы)

Например:

Выведем все названия файлов докментов и их типы списком
/api/companydocument?fields=FileName,TypeId

search – *строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке.*

Например:

а) Выведем все документы, у которых в одном из полей для поиска содержится подстрока «Test»:
/api/companydocument?search=Test

б) Выведем все документы, у которых в одном из полей для поиска содержится подстрока «Test», и вывод представим только 2мя полями для каждого документа – «Название файла» и «Тип документа»
/api/companydocument?fields=FileName,TypeId&search=Test

page, pagesize *– параметры отображения таблицы*

{ filterFields } *поля для фильтрации*

Все условия фильтрации склеиваются по условию И. Подробнее в разделе о фильтрации документов компаний

Пример вывода в xml

GET: /api/companydocument

```
<CompanyDocumentList>
    <Documents>
        <Document>
            <Id>2</Id>
            <FileName>Документ_договора.docx</FileName>
            <NameFromUser>Договор на обслуживание</NameFromUser>
            <IsActive>True</IsActive>
            <CompanyId>93</CompanyId>
            <CompanyFullName>Тестовый клиент</CompanyFullName>
            <CreatorId>43</CreatorId>
            <CreatorName>Администратор</CreatorName>
            <EditorId>43</EditorId>
            <EditorName>Администратор</EditorName>
            <FileId>3</FileId>
            <TypeId>2</TypeId>
            <TypeName>Договор</TypeName>
            <Changed>24.11.2017 13:14:09</Changed>
            <Created>24.11.2017 13:14:09</Created>
            <DateBegin>01.11.2017 19:00:00</DateBegin>
            <DateEnd>01.11.2018 19:00:00</DateEnd>
```

```
                </Document>
            <Document>
                <Id>1</Id>
                <FileName>Image-1.jpg</FileName>
                <NameFromUser>Логотип_компании</NameFromUser>
                <IsActive/>
                <CompanyId>93</CompanyId>
                <CompanyFullName>Тестовый клиент</CompanyFullName>
                <CreatorId>43</CreatorId>
                <CreatorName>Администратор</CreatorName>
                <EditorId>43</EditorId>
                <EditorName>Администратор</EditorName>
                <FileId>2</FileId>
                <TypeId>1</TypeId>
                <TypeName>Другое</TypeName>
                <Changed>24.11.2017 13:13:34</Changed>
                <Created>24.11.2017 13:13:34</Created>
                <DateBegin/>
                <DateEnd/>
            </Document>
        </Documents>
        <Paginator>
            <Count>2</Count>
            <Page>1</Page>
            <PageCount>1</PageCount>
            <PageSize>25</PageSize>
            <CountOnPage>2</CountOnPage>
        </Paginator>
</CompanyDocumentList>
```

Фильтрация списка документов компаний

Поля фильтрации

Поле фильтрации	Тип передаваемого значения	Описание фильтрации
FileName	string	Документы, в названии файла которых содержится указанная строка.
NameFromUser	string	Документы, в пользовательском названии которых содержится указанная строка.
CompanyIds	string	Документы, которые относятся к одной из указанных компаний. Перечисление идентификаторов (Id) через запятую.
TypeId	int	Документы, которые имею тип: 1 – другое, 2 - договор
IsActive	bool	Договора, признак активности у которых стоит в указанном положении. Только для типа 2 – договор.
DateBeginLessThan	DateTime	Дата начала действия договора меньше или равна указанной дате. Только для типа 2 – договор.

DateBeginMoreThan	DateTime	Дата начала действия договора больше или равна указанной дате. Только для типа 2 – договор.
DateBegin	DateTime	Дата начала действия договора договора равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня). Только для типа 2 – договор.
DateEndLessThan	DateTime	Дата окончания действия договора меньше или равна указанной дате. Только для типа 2 – договор.
DateEndMoreThan	DateTime	Дата окончания действия договора больше или равна указанной дате. Только для типа 2 – договор.
DateEnd	DateTime	Дата окончания действия договора равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня). Только для типа 2 – договор.
ChangedLessThan	DateTime	Дата последнего изменения документа меньше или равна указанной дате
ChangedMoreThan	DateTime	Дата последнего изменения документа больше или равна указанной дате
Changed	DateTime	Дата последнего изменения документа равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
CreatedLessThan	DateTime	Дата создания документа меньше или равна указанной дате
CreatedMoreThan	DateTime	Дата создания документа больше или равна указанной дате
Created	DateTime	Дата создания документа равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)

Примеры запросов:

1) Получим документы типа «договор» для компаний с идентификаторами 100 и 101, в название файла которых содержится подстрока «обслуживание» и которые активны:
   GET:
   /api/companydocument?typeid=2&companyids=100,101&filename=обслуживание&isactive=true
2) Получим все документа типа «договор», у которых договор закончится после 12 декабря 2017 года, т.е. дата окончания договора больше 12 декабря 2017 года:
   GET: /api/companydocument?typeid=2&dateendmorethan=2017-12-12

Получение определенного документа компании

Вызов:

GET: api/companydocument/{id}

Где *{id}* – идентификатор документа компании

Пример вывода в xml:

```
<CompanyDocumentView>
        <Id>2</Id>
        <FileName>Документ_договора.docx</FileName>
        <NameFromUser>Договор на обслуживание</NameFromUser>
        <CompanyId>93</CompanyId>
        <CompanyFullName>Тестовый клиент</CompanyFullName>
        <IsActive>True</IsActive>
        <TypeId>2</TypeId>
        <TypeName>Договор</TypeName>
        <FileId>3</FileId>
        <DateBegin>01.11.2017 19:00:00</DateBegin>
        <DateEnd>01.11.2018 19:00:00</DateEnd>
        <EditorId>43</EditorId>
        <EditorName>Администратор</EditorName>
        <CreatorId>43</CreatorId>
        <CreatorName>Администратор</CreatorName>
        <Created>24.11.2017 13:14:09</Created>
        <Changed>24.11.2017 13:14:09</Changed>
</CompanyDocumentView>
```

Метод POST. Создание нового документа компании

Поля для создания документа

Поле	Тип значения	Обязательность	Комментарий
TypeId	int	+	Типа документа: 1 – другое, 2 - договор
CompanyId	int	+	Идентификатор компании, к которой относится документ.
FileId	Int	+	Идентификатор файла документа. Для загрузки файла и получения его идентификатора используйте
DateBegin	DateTime	+/-	Начало действия договора. Поле **обязательно**, если тип договора TypeId равен 2 – «договор»
DateEnd	DateTime	+/-	Окончание действия договора. Поле **обязательно**, если тип договора TypeId равен 2 – «договор»

NameFromUser	string		Пользовательское название документа. Если не указано, то будет совпадать с именем загруженного ранее файла FileId
IsActive	Bool?		Признак активности договора. Если не указан, то для типа 2 «договор» по умолчанию станет True.

Вызов и пример использования

POST: /api/companydocument

В теле запроса передается объект CompanyDocument. Допускается передавать только те поля, которые содержат значения. Обязательно должны быть заполнены поля, отмеченные в таблице как обязательные.
Запрос возвращает все поля созданной компании - аналогично запросу Получение определенного документа компании
Пример:

1) Добавим обычный документ типа «Другое» для компании с идентификатором 111 и именем «Мой документ». Файлом этого документа является предварительно загруженный файл «» с идентификатором 21 (как загрузить документ и получить его идентификатор): *{“TypeId": 1, “* NameFromUser *”:”Мой документ”, “CompanyId”:111, “FileId”:21}*
2) Добавим документ типа «Договор» с файлом «Документ_договора.docx», который был предварительно загружен и его идентификатор – 23 (как загрузить документ и получить его идентификатор) для компании с идентификатором 112, датой начала договора 1 ноября 2017 года и датой окончания договора 1 ноября 2018 года: *{“TypeId": 2, “CompanyId”:112, “FileId”:23, “DateBegin”:”01.11.2017”, “DateEnd”: “01.11.2018”}*

Метод PUT. Изменение существующего документа

Поля документа для изменения

Поля документа для изменения аналогичны, как и при создании (указаны в таблице).

Вызов и пример использования

PUT: /api/companydocument/{id}
Где ***id*** – идентификатор документа

Пример:
Изменение документа с id = 123: изменение названия на «Тестовый документ изменение»,
PUT: /api/companydocument/123
*{“*NameFromUser*” : ”Тестовый документ изменение”}*

Файл документа компании

/api/companydocumentfile

Метод GET. Получение файла документа

Вызов:

GET: /api/companydocumentfile/{id}

Где {id} – идентификатор файла документа (поле **FileId** из полей документа компании)

Метод POST. Сохранение файла документа и прикрепление его к документу.

Привязывание файла к документу состоит из 2х этапов.

Сохранение файла и получение уникального идентификатора

Вызов для сохранения файла

POST /api/companydocumentfile

Параметрами запроса служат:

{fileKey} - файлы (количество файлов не ограничено, можно загрузить сразу для нескольких документов);

***{filekey}*Name** имя файла(Необязательный параметр).

Например:

<input name="file0" type="file"/>

<input name="file0Name" type="text"/>

Примечание:

В заголовках запроса необходимо указывать ***content-type: multipart/form-data*** и отсылать файлы соответственно этому заголовку -более подробно можно ознакомиться в документациях к языку программирования, на котором вы пишете, также вы можете запросить у нас пример кода на c#.

Получаемый ответ на сохранение файла

В ответе получаем ***{"FileTokens":"23"}***

Прикрепление сохраненного файла к документу

Полученный ранее идентификатор файла из ***FileTokens*** – передаем в запросе для создания или изменения документа в поле **FileId.**

Пользователь

/api/user

Метод GET. Получение списка пользователей и конкретного пользователя

Поля пользователя для получения

Название	Тип	Используется в search	Описание
Id	Int		Идентификатор пользователя. Присваивается автоматически при создании
RoleId	Int		Идентификатор системной роли

CompanyId	Int		Идентификатор компании
CompanyName	String	+	Название компании
CompanyPath			
Login	String	+	Логин пользователя в системе
Name	String	+	Имя
Email	String	+	Email
Phone	String	+	Телефон
Comments	String		Описание
IsArchive	Boolean		Архивный
Position	String	+	Должность
MobilePhone	String	+	Мобильный
TimeZone	String		Часовой пояс
CurrentLanguage	String		Язык
ADGuid	String		Идентификатор пользователя в AD
Changed	DateTime		Последняя дата изменения пользователя.
UtcOffset	string		Смещение от мирового времени (указано только при получении конкретного пользователя)

Получение списка пользователей:

Вызов

GET: /api/user?fields={fieldList}&search={searchString}&{filterFields}&pagesize={value}&page={value}

Возможные параметры

fields *- перечень полей ресурса «Пользователь» для вывода (из таблицы)*

Например: Выведем все имена и системные роли пользователей
/api/user?fields=Name, RoleId

search – строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке, а также в ***названии сервиса по умолчанию*** *для пользователя и в* ***названии его системной роли****.*

Например:

а) Выведем всех пользователей, у которых в одном из полей для поиска содержится подстрока «Test»:

/api/user?search=Test

б) Выведем всех пользователей, у которых в одном из полей для поиска содержится подстрока «Test», и вывод представим только 2мя полями для каждого пользователя – «Название» и «Тип»

/api/user?fields=FullName, TypeId&search=Test

page, pagesize *– параметры отображения таблицы*

{filterFields} - *поля для фильтрации*

Все условия фильтрации склеиваются по условию И. Подробнее в разделе о фильтрации пользователей

Пример вывода в xml

GET: /api/user?fields=Id,Name,RoleId&pagesize=2

```
<UserList>
    <Users>
```

```
            <User>
                <Id>45</Id>
                <Name>admin2</Name>
                <RoleId>65</RoleId>
            </User><User>
                <Id>44</Id>
                <Name>test 1</Name>
                <RoleId>37</RoleId>
            </User>
        </Users>
        <Paginator>
            <Count>3</Count>
            <Page>1</Page>
            <PageCount>2</PageCount>
            <PageSize>2</PageSize>
            <CountOnPage>2</CountOnPage>
        </Paginator>
</UserList>
```

Фильтрация списка пользователей

Поля фильтрации

Поле фильтрации	Тип передаваемого значения	Описание фильтрации
ChangedLessThan	DateTime	Дата последнего изменения пользователя меньше или равна указанной дате
ChangedMoreThan	DateTime	Дата последнего изменения пользователя больше или равна указанной дате
Changed	DateTime	Дата последнего изменения пользователя равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
CompanyIds	string	Пользователи из указанных подразделений. Перечисление идентификаторов (Id) через запятую.
RoleIds	string	Пользователи с указанными системными ролями. Перечисление идентификаторов (Id) через запятую.
Ids	string	Пользователи с указанными идентификаторами. Перечисление идентификаторов (Id) через запятую.
CompanyName	string	Пользователи, в названии компании которых содержится указанная строка.

CompanyPath	string	Пользователи, путь компаний которых начинается с указанного. Используется для поиска пользователей из подразделения и всех его дочерних.
Login	string	Пользователи, в логине которых содержится указанная строка
Name	string	Пользователи, в имени которых содержится указанная строка
Email	string	Пользователи, email которых содержит указанную строку
MobilePhone	string	Пользователи, мобильный телефон которых содержит указанную строку
Phone	string	Пользователи, телефон которых содержит указанную строку
Position	string	Пользователи, название должности которых содержит указанную строку
Comments	string	Пользователи, для которых комментарий содержит указанную строку.
ADGuid	guid	Пользователи, идентификатор в Active Directory которых равен указанному (должен быть один в выводе)
IsArchive	bool	Пользователи, признак архивный у которых стоит в указанном положении

Примеры запросов:

1) Получим пользователей, email которых сдержит домен «mycompany.ru» и которые не архивны:
   GET: /api/user?email=mycompany.ru&isarchive=false
2) Получим пользователя с идентификатором в AD «1617e62f-1c4e-496d-9819-31bc80487718»:
   GET: /api/user?adguid=1617e62f-1c4e-496d-9819-31bc80487718
3) Получим всех пользователей, у которых системная роль с id = 22 или 25, а также информация о которых была изменена после 13 ноября 2015 года:
   GET: /api/user?roleids=22,25&changedmorethan=2015-11-13

Получение определенного пользователя

Вызов:

GET: api/user/userid

Где userid– идентификатор пользователя

Пример вывода в xml:

```
<UserView>
        <ADGuid i:nil="true"/>
        <Changed>2015-11-16T12:30:03.443</Changed>
        <Comments i:nil="true"/>
        <CompanyId>91</CompanyId>
        <CompanyName>Подразделение 1</CompanyName>
        <CompanyPath i:nil="true"/>
        <CurrentLanguage i:nil="true"/>
        <Email>nr@intravision.ru</Email>
        <Id>44</Id>
        <IsArchive>false</IsArchive>
        <Login>test1</Login>
        <MobilePhone i:nil="true"/>
        <Name>test 1</Name>
        <Phone i:nil="true"/>
        <Position i:nil="true"/>
        <RoleId>37</RoleId>
        <TimeZone>Russian Standard Time</TimeZone>
        <UtcOffset>+03:00</UtcOffset>
</UserView>
```

Получение информации по текущему пользователю

Вызов

GET /api/user?getcurrentuserinfo=true

Запрос вернет следующие поля для текущего пользователя:

Language - текущий язык

UtcOffset - числовое значение смещения от мирового времени (UTC)

Id – идентификатор пользователя

Login – логин пользователя

Email – емейл пользователя

Name – имя пользователя

RoleId – идентификатор системной роли пользователя

RoleType – тип системной роли пользователя (1 – Исполнитель, 2 - Клиент)

CompanyId – идентификатор компании пользователя

IsArchive – признак архивности пользователя

DefaultTaskFilterId – идентификатор фильтра по умолчанию для списка заявок

А также поля – *Phone*, *MobilePhone*, *Position*, *Comments* – Телефон, Мобильный телефон, Должность, Описание

Пример вывода в xml

```
        <CurrenUserInfo>
                <Comments i:nil="true"/>
                <CompanyId>30</CompanyId>
                <DefaultTaskFilterId>106</DefaultTaskFilterId>
                <Email>test@test.ru</Email>
                <Id>1</Id>
                <IsArchive>false</IsArchive>
                <Language>ru</Language>
```

```
        <Login>admin</Login>
        <MobilePhone i:nil="true"/>
        <Name>Администратор</Name>
        <Phone i:nil="true"/>
        <Position i:nil="true"/>
        <RoleId>37</RoleId>
        <RoleType>1</RoleType>
        <UtcOffset>+03:00</UtcOffset>
</CurrenUserInfo>
```

Метод POST. Создание пользователя

Поля для создания пользователя

Название	Тип	Обязательность	Описание
RoleId	Int	+	Идентификатор системной роли
CompanyId	Int	+	Идентификатор компании
Login	String	+	Логин пользователя в системе
Name	String	+	Имя
Password	String	+	Пароль.
ConfirmPassword	String	+	Подтверждение пароля.
Email	String		Email
Phone	String		Телефон
Comments	String		Описание
IsArchive	Bool		Архивный
Position	String		Должность
MobilePhone	String		Мобильный
TimeZone	String		Часовой пояс
CurrentLanguage	String		Язык

Вызов и пример использования

POST: /api/user

В теле запроса передается объект User. Допускается передавать только те поля, которые содержат значения. Обязательно должны быть заполнены поля, отмеченные в таблице как обязательные. Запрос возвращает все поля созданного пользователя (аналогично запросу Получение определенного пользователя).

Пример:

Создадим пользователя с ролью 2, компанией 10, именем «Иванов Андрей», логином «ivanov», e-mail'ом «ivanov@mail.ru» и паролем «test123»:

POST: /api/user

{"RoleId": 2, "CompanyId":10, "Login":"ivanov", "Name ":"Иванов Андрей", "Password":"test123"," ConfirmPassword":"test123","Email":"ivanov@mail.ru"}

Метод PUT. Изменение пользователя

Поля для изменения пользователя

Поля пользователя для изменения аналогичны, как и при создании (указаны в таблице).

Как и везде в методах изменения PUT достаточно отсылать только те поля, которые изменяются, кроме поля **Password**: чтобы изменить пароль, нужно отослать пару значений – **Password** и **ConfirmPassword**.

Вызов и пример использования

PUT: /api/user/userid
Где ***userid*** – идентификатор конкретного пользователя

Пример:
Изменим имя пользователя с id= 35 на «Имя35» и переведем его в компанию 64:

PUT: /api/user/35
{"CompanyId":64, "Name":"Имя35"}

Группа исполнителей

/api/taskexecutorgroup

Метод GET. Получение групп исполнителей для сервиса

Поля группы исполнителей для получения

Название	Тип	Используется в search	Описание
Id	Int		Идентификатор группы исполнителей
Name	String	+	Название группы исполнителей
Description	String		Описание группы исполнителей
ExecutorIds	String		Список идентификаторов исполнителей через запятую
Executors	String		Список имен исполнителей через запятую

Получение списка групп исполнителей, назначенных на сервис

Вызов

GET:
/api/taskexecutorgroup?serviceid={serviced}&fields={fieldList}&search={searchString}&pagesize={value}&page={value}

Возможные параметры

serviceid *– обязательный параметр.*

Если не будет указан – вернется пустой список.

fields *- перечень полей ресурса «Группа исполнителей» для вывода (из таблицы)*

Например: Выведем все список названий групп исполнителей со списком имен исполнителей в каждой группе
/api/taskexecutorgroup?fields=Name,ExecutorIds

search *– строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке.*

Например:
а) Выведем все группы исполнителей, у которых в названии содержится подстрока «Test»:

/api/user?search=Test

б) Выведем все группы исполнителей, у которых в названии содержится подстрока «Тестовая», и вывод представим только 2мя полями для каждой группы – «Название» и «Список исполнителей»

/api/taskexecutorgroup?fields=Name,ExecutorIds&search=Тестовая

***page**, **pagesize** – параметры отображения таблицы*

Пример вывода в xml

GET /api/taskexecutorgroup?serviceid=16

```
<ExecutorGroupList>
    <ExecutorGroups>
        <ExecutorGroup>
            <Id>1</Id>
            <Name>Группа 1</Name>
            <Description/>
            <ExecutorIds>45, 43</ExecutorIds>
            <Executors>admin2, Администратор</Executors>
        </ExecutorGroup>
        <ExecutorGroup>
            <Id>2</Id>
            <Name>Группа 2</Name>
            <Description/>
            <ExecutorIds>45, 44</ExecutorIds>
            <Executors>admin2, test 1</Executors>
        </ExecutorGroup>
    </ExecutorGroups>
    <Paginator>
        <Count>2</Count>
        <Page>1</Page>
        <PageCount>1</PageCount>
        <PageSize>10</PageSize>
        <CountOnPage>2</CountOnPage>
    </Paginator>
</ExecutorGroupList>
```

Тип заявки

/api/tasktype

Метод GET. Получение списка типов заявки и определенного типа заявки

Поля типа заявки для получения

Название	Тип	Описание
Id	Int	Идентификатор типа заявки
Name	String	Название типа заявки
Description	String	Описание типа заявки
IsArchive	Bool	Архивный
Image16Url	String	Url картинки
Changed	DateTime	Дата изменения типа заявки.
CanCreateTask	Bool	Можно ли в этом типе текущему пользователю создавать заявки. Выводится только при указании serviced>0
TaskTypeFields	TaskTypeFieldView Array	Список дополнительных полей типа заявки (только при получении определенного типа заявки)

Получение списка типов заявки

Вызов

GET
/api/tasktype?fields={fieldList}&serviceid={value}&archive={value}&pagesize={value}&page={value}

Возможные параметры

serviceid *– идентификатор сервиса. Необязательный параметр.*

Позволяет получить список типов заявки для определенного сервиса. Если не указан – то будут выведены все типы заявок
Пример: Выведем типы заявок для сервиса 43
GET /api/tasktype?serviceid=43

fields *– перечень полей ресурса «Тип заявки» для вывода (из таблицы)*

Пример: выведем все типы заявок списком из 2х полей: идентификатор и Название
GET /api/tasktype?fields=Id,Name

archive *– позволяет фильтровать список типов заявки по признаку «Архивный»*

Возможные значения

- **false** – выведет все неархивные типы заявок. Значение **по-умолчанию** (когда параметр не задан);
- **true** – выведет все типы заявок: как архивные, так и неархивные

page, pagesize *– параметры отображения таблицы*

Пример вывода в xml

GET: /api/tasktype?fields=Id,Name&archive=true&pagesize=2

```
<TaskTypeList>
    <TaskTypes>
        <TaskType>
            <Id>1004</Id>
            <Name>Стандартный</Name>
        </TaskType>
```

```
            <TaskType>
                <Id>3</Id>
                <Name>Инцидент</Name>
            </TaskType>
        </TaskTypes>
        <Paginator>
            <Count>10</Count>
            <Page>1</Page>
            <PageCount>5</PageCount>
            <PageSize>2</PageSize>
            <CountOnPage>2</CountOnPage>
        </Paginator>
    </TaskTypeList>
```

Фильтрация списка типов заявки

Поля фильтрации

Поле фильтрации	Тип передаваемого значения	Описание фильтрации
ChangedLessThan	DateTime	Дата последнего изменения компании меньше или равна указанной дате
ChangedMoreThan	DateTime	Дата последнего изменения компании больше или равна указанной дате
Changed	DateTime	Дата последнего изменения компании равна указанному дню (время не учитывается, т.е. с 00:00 до 23:59 указанного дня)
HasFields	bool	Имеются ли у типа заявки дополнительные поля

Примеры запросов:

1) Получим типы заявки, у которых дата изменения больше 3 апреля 2016 года:
   GET: /api/tasktype?ChangedMoreThan=2016-04-03
2) Получим типы заявок, у которых есть дополнительные поля и дата изменения меньше 1 апреля 2016 года
   GET: /api/tasktype?ChangedLessThan=2016-04-01&HasFields=true

Получение конкретного типа заявки

Вызов

GET: /api/tasktype/{tasktypeid}?serviceid={value}
где ***{tasktypeid}*** – идентификатор типа заявки
Используется при создании или изменении заявки, чтобы получить информацию о типе заявке и правах текущего пользователя на редактирование его дополнительных полей.

Возможные параметры

***serviceid** – обязательный параметр*

Получить тип заявки *tasktypeid*, привязанный к сервису *serviced*. Если на данном сервисе нет такого типа – то вернет ошибку. Идентификатор сервиса используется для определения прав пользователя: для пользователя определяется его роль на сервисе, и исходя из нее, права доступа к полям типа (чтение, редактирование)

Вывод дополнительного поля заявки TaskTypeFieldView в списке TaskTypeFields

Поля дополнительного поля TaskTypeFieldView

Название	Тип	Описание
Id	Int	Идентификатор поля
Name	String	Название поля
FieldTypeId	Int	Тип поля: 1 – Строка, 2 – Число, 3 – Дата, 4 – Чекбокс, 5 – Список, 6 – Файл, 7 - Текст
IsRequired	Boolean	Обязательное поле
DefaultValue	String	Значение по умолчанию
Unit	String	Единица измерения
Hint	String	Подсказка
TaskTypeComboBoxes	TaskTypeComboBoxView Array	Значения элемента выпадающий список (5). Представлены в виде списка из TaskTypeComboBoxView. Содержит элементы списка (только для типа поля «список»).
Rights	int	Сумма прав: 1 – чтение при изменении заявки 2 – редактирование при изменении заявки 4 – редактирование при создании заявки

Возвращаются в отсортированном виде.

Поля значения дополнительного поля «Выпадающий список» TaskTypeComboBoxView

Название	Тип	Описание
Id	Int	Идентификатор элемента
Name	String	Название элемента
Path	String	Путь
ParentId	Int	Идентификатор родительского элемента (для многоуровневых списков)

Возвращаются в отсортированном виде.

Пример вывода в xml

На сервисе 16 присутствует тип заявки 1005. Тип заявки имеет 2 дополнительных поля: «строка» и «выпадающий список» с 3мя значениями, включающими иерархию:

GET: /api/tasktype/1005?serviceid=16

```
<TaskTypeView>
    <Description/>
    <Id>1005</Id>
    <Image16Url>http://localhost/IntraService4.40p2/img/tasktypes/blue.jpg </Image16Url>
    <IsArchive>false</IsArchive>
    <Name>type 1</Name>
    <TaskTypeFields>
        <TaskTypeFieldView>
            <DefaultValue>мсст значение</DefaultValue>
            <FieldTypeId>1</FieldTypeId>
            <Hint/>
            <Id>1014</Id>
            <IsRequired>false</IsRequired>
            <Name>Строка</Name>
            <Rights>6</Rights>
            <TaskTypeComboBoxes i:nil="true"/>
            <Unit/>
        </TaskTypeFieldView>
        <TaskTypeFieldView>
            <DefaultValue i:nil="true"/>
            <FieldTypeId>5</FieldTypeId>
            <Hint/>
            <Id>1018</Id>
            <IsRequired>false</IsRequired>
            <Name>Список</Name>
            <Rights>6</Rights>
            <TaskTypeComboBoxes>
                <TaskTypeComboBoxView>
                    <Id>17</Id>
                    <Name>1</Name>
                    <ParentId i:nil="true"/>
                    <Path>17|</Path>
                </TaskTypeComboBoxView>
                <TaskTypeComboBoxView>
                    <Id>19</Id>
                    <Name>1.1</Name>
                    <ParentId>17</ParentId>
                    <Path>17|19|</Path>
                </TaskTypeComboBoxView>
                <TaskTypeComboBoxView>
                    <Id>18</Id>
                    <Name>2</Name>
                    <ParentId i:nil="true"/>
                    <Path>18|</Path>
                </TaskTypeComboBoxView>
            </TaskTypeComboBoxes>
            <Unit/>
        </TaskTypeFieldView>
```

</TaskTypeFields>
</TaskTypeView>

Жизненный цикл заявки

/api/tasklifetime

Метод GET. Получение событий жизненного цикла для заявки

Поля жизненного цикла для получения

Название	Тип	Описание
Date	DateTime	Дата изменения
EditorId	Int	Id пользователя, изменившего заявку
Editor	String	Имя пользователя, изменившего заявку
StatusId	Int	Id статуса
Comments	String	Комментарий
IsPublic	Boolean	Комментарий доступен клиенту
Executors	String	Список имен исполнителей через запятую
ExecutorsGroup	String	Группа исполнителей
Creator	String	Заявитель
Participants	String	Список имен наблюдателей через запятую
Name	String	Название
Description	String	Описание
Categories	String	Имена категорий через запятую
PriorityId	Int	Id приоритета
Files	String	Список **всех** имен и id файлов, разделенных «|», через запятую. Показывается в первой записи и в записи, в которой имеет место быть изменение списка файлов.
RemovedFiles	String	Список удаленных файлов в виде «<fileid>|<filename>» через запятую. Присутствует в первой записи и только если были удалены какие-либо файлы.
AddedFiles	String	Список добавленных файлов в виде «<fileid>|<filename>» через запятую. Присутствует в первой записи и только если были добавлены какие-либо файлы.
ServiceId	Int	Id сервиса
Deadline	DateTime	Срок исполнения

Получение жизненного цикла заявки

Вызов

GET:
/api/tasklifetime?taskid={task_id}&include={resource_name_list}&lastcommentsontop={value}&pagesize={value}&page={value}
Список отсортирован в соответствии с настройками в системе либо в соответствии с необязательным параметром *lastcommentsontop*.

Возможные параметры

***taskid** – обязательный параметр*

номер заявки

Пример: Введем жизненный цикл заявки № 255

GET: /api/tasklifetime?taskid=255

***include** – необязательный параметр.*

Позволяет добавлять связанные ресурсы.

Возможные значения:

- **PRIORITY** (приоритет);
- **SERVICE** (сервис);
- **STATUS** (статус)

***lastcommentsontop** – необязательный параметр.*

Позволяет отсортировать жизненный цикл в порядке «Последний комментарий вверху списка»

Возможные значения:

true или **false**

Пример вывода в xml

GET: /api/tasklifetime?taskid=151&include=status

```
<TaskLifetimeList>
    <TaskLifetimes>
        <TaskLifetime>
            <Date>12.11.2015 13:44:53</Date>
            <EditorId>43</EditorId>
            <Editor>Администратор</Editor>
            <StatusId>29</StatusId>
        </TaskLifetime>
        <TaskLifetime>
            <Date>11.11.2015 15:24:10</Date>
            <EditorId>43</EditorId>
            <Editor>Администратор</Editor>
            <StatusId>31</StatusId>
        </TaskLifetime>
    </TaskLifetimes>
    <Statuses>
        <Status>
            <Id>31</Id>
            <Name>Открыта</Name>
            <Description/>
            <Image16Url>http://localhost/IntraService4.40/img/statuses/actual16.png </Image16Url>
            <Image24Url>http://localhost/IntraService4.40/img/statuses/actual24.png </Image24Url>
            <IsCommentRequired>False</IsCommentRequired>
            <IsFinal>False</IsFinal>
            <IsFixed>False</IsFixed>
            <IsDeadlineAccountingPaused>False</IsDeadlineAccountingPaused>
            <IsConcurrence>False</IsConcurrence>
            <IsExternal>False</IsExternal>
```

```
            </Status>
            <Status>
                <Id>29</Id>
                <Name>Выполнена</Name>
                <Description/>
                <Image16Url>http://localhost/IntraService4.40/img/statuses/fixed16.png </Image16Url>
                <Image24Url>http://localhost/IntraService4.40/img/statuses/fixed24.png </Image24Url>
                <IsCommentRequired>False</IsCommentRequired>
                <IsFinal>False</IsFinal>
                <IsFixed>True</IsFixed>
                <IsDeadlineAccountingPaused>True</IsDeadlineAccountingPaused>
                <IsConcurrence>False</IsConcurrence>
                <IsExternal>False</IsExternal>
            </Status>
        </Statuses>
        <Paginator>
            <Count>2</Count><Page>1</Page><PageCount>1</PageCount><PageSize>25</PageSize><Coun
            tOnPage>2</CountOnPage>
        </Paginator>
</TaskLifetimeList>
```

Трудозатраты

api/taskexpenses

Метод GET. Получение трудозатрат по заявке и определенных трудозатрат

Поля трудозатрат для получения

Название	Тип	Описание
Id	Int	Идентификатор трудозатраты
UserId	Int	Идентификатор исполнителя, за которого списана трудозатрата
UserName	String	Имя исполнителя, за которого списана трудозатрата
Date	DateTime	Дата списания трудозатрат
Minutes	Int	Количество минут
Rate	Decimal	Рейт
Comments	String	Комментарий
TaskId	Int	Номер заявки, к которой относятся трудозатраты
EditorId	Int	Идентификатор пользователя, создавшего или изменившего последним трудозатрату
EditorName	String	Имя пользователя, создавшего или изменившего последним трудозатрату

Получение всех трудозатрат для заявки.

Вызов:

GET: api/taskexpenses?taskid={taskid}&fields={fieldList}&pagesize={value}&page={value}

Возможные параметры

***taskid** – обязательный параметр*

номер заявки, трудозатраты которой вы хотите получить

***fields** – список полей, которые вы хотите получить (указаны в таблице)*

Например:

Получить трудозатраты по заявке №143199 в виде списка идентификатора, минут и даты

GET api/taskexpenses/?taskid=143199&fields=Id,Minutes,Date

***pagesize**, **page** – параметры отображения таблицы*

Пример вывода в xml

```
<TaskExpensesList>
        <Expenses>
                <Expense>
                        <Id>53</Id>
                        <UserName>Администратор</UserName>
                        <UserId>43</UserId>
                        <Date>02.09.2015 0:00:00</Date>
                        …..
                </Expense>
                <Expense>
                        <Id>54</Id>
                        <UserName>Админ 2</UserName>
                        <UserId>70</UserId>
                        <Date>02.09.2015 0:00:00</Date>
                        …..
                </Expense>
        </Expenses>
        <Paginator>
                <Count>2</Count>
                <Page>1</Page>
                <PageCount>1</PageCount>
                <PageSize>25</PageSize>
                <CountOnPage>2</CountOnPage>
        </Paginator>
</TaskExpensesList>
```

Проверка доступа.

На сервисе, в котором находится заявка, трудозатраты которой мы пытаемся получить, у пользователя должна быть роль, у которой в типе заявке для поля «Трудозатраты» выставлено полномочие «Просмотр» или же «Изменение». Иначе вернет ошибку доступа.
Если у сервисной роли нет прав на «Просмотр стоимости работ», то Rate не будет присутствовать в ответе сервера.

Получение определенных трудозатрат

Вызов:

GET: /api/taskexpenses/id
Где id – идентификатор трудозатраты.

Пример вывода в xml

```
<Expense>
        <Id>53</Id>
        <UserId>43</UserId>
        <UserName>Администратор</UserName>
        <Date>02.09.2015 0:00:00</Date>
        <Minutes>63</Minutes>
        <Rate>11,000</Rate>
        <Comments/>
        <TaskId>243</TaskId>
        <EditorId>43</EditorId>
        <EditorName>Администратор</EditorName>
</Expense>
```

Проверка доступа.

Аналогично, как и для получения всех трудозатрат по заявке

Метод POST. Создание трудозатрат

Поля трудозатрат для создания

Поле	Тип значения	Обязательность	Комментарий
TaskId	Целое число (int)	Да	Номер заявки, к которой добавляются трудозатраты
Minutes	Целое число (int)	Да	Количество минут
UserId	Целое число (int)	Нет	Идентификатор пользователя, за которого списываются трудозатраты. Если значения нет – берется текущий пользователь. Получить список доступных исполнителей для списания трудозатрат.
Rate	Число с плавающей точкой (decimal)	Нет	Рейты. Если рейты не заданы, то возьмется значение по умолчанию для роли на сервисе для текущего пользователя.
Comments	Строка	Нет	Комментарий
Date	Дата (DateTime)	Нет	Дата трудозатрат без времени, если значения нет, то возьмется текущая дата

Вызов и пример использования

POST: /api/taskexpenses

Пример:
Добавим трудозатраты в заявку № 777 – 45 минут с комментарием:
{“TaskId”: 777, “Minutes”:45, “Comments”: ”Добавление трудозатраты в заявку 777”}

Проверка доступа.

1) Текущий пользователь должен быть назначен на сервис, в котором находится заявка **TaskId**.
2) У пользователя в типе заявки для **TaskId** в «Основных полях заявки», поле «Трудозатраты», должно стоять полномочие «Изменение».
3) Если в запросе передается **UserId**, отличный от **Id** текущего пользователя, то у текущего пользователя должны быть права на «Редактирование чужих трудозатрат» (карточка роли) – в противном случае будет ошибка, что списать трудозатраты за другого нельзя.
4) Если **UserId** не является исполнителем на заявке, то списать трудозатраты не получится.
5) Если у текущего пользователя нет полномочия на карточки роли «Редактировать стоимость работ», а в запросе передается значение для **Rate**, то возьмется значение **Rate** по умолчанию для роли пользователя на сервисе, которое указано для каждой роли в системе.

Метод PUT. Изменение трудозатрат

Поля трудозатрат для изменения

Поле	Тип значения	Комментарий
Minutes	Целое число (int)	Количество минут
UserId	Целое число (int)	Идентификатор пользователя, за которого списываются трудозатраты. Получить список доступных исполнителей для списания трудозатрат.
Rate	Число с плавающей точкой (decimal)	Рейты
Comments	строка	Комментарий

Вызов и пример использования

PUT: api/taskexpenses/{id}
Где **{id}**– идентификатор трудозатрат.

Пример:
Изменим трудозатраты с Id = 666 –меняем количество минут с комментарием:
PUT /api/taskexpenses/666
{“Minutes”:45, “Comments”:”Изменение минут в трудозатрате 666”}

Проверка доступа.

1) Текущий пользователь должен быть назначен на сервис, в котором находится заявка **TaskId**.
2) У пользователя в типе заявки для **TaskId** в основных полях заявки у поля «Трудозатраты» должно быть выставлено полномочие «Изменение»

3) Если текущий пользователь пытается изменить трудозатраты другого **UserId**, то тогда у текущего пользователя должны быть права «Редактировать чужие трудозатраты»
4) Если в запросе передается новое значение **UserId**, то этот пользователь должен быть исполнителем на заявке и у текущего пользователя должно быть право «Редактировать чужие трудозатраты»
5) Если в запросе передается новое значение **Rate** и у текущего пользователя нет прав «Редактировать стоимость работ», то это значение будет игнорироваться.

Метод GET. Получение возможных исполнителей для списания трудозатрат

/api/expensesexecutors

Поля для получения

Название	Тип	Описание
Id	Int	Идентификатор пользователя
Name	String	Имя пользователя
Rate	decimal	Рейт пользователя в соответствии с сервисной ролью

Вызов

GET /api/expensesexecutors?taskid={taskid}&searchstring={searchstring}

где **{taskid}** – обязательный параметр. Номер заявки, для которой пытаемся добавить трудозатраты.

{searchstring} – необязательный параметр. Строка поиска в имени пользователя.

Если параметр не передается или передается пустая строка - покажет исполнителей заявки (также из группы исполнителей на заявке) и текущего пользователя.

Если параметр задан – покажет пользователей с ролью типа «Исполнитель» на сервисе, у которых есть вхождение заданной строки в имя пользователя.

Пример вывода в JSON-формате


```
[
	{
		"Id": 45,
		"Name": "тестовый исполнитель 2",
		"Rate": 5.00
	},
	{
		"Id": 43,
		"Name": "Администратор",
		"Rate": 0.00
	}
]
```


Проверка доступа

1) Текущему пользователю должна быть доступна на просмотр заявка taskid.
2) Поиск по условию searchstring осуществляется в соответствии с настройкой доступа «Не ограничивать просмотр/редактирование пользователей/подразделений своим подразделением и нижестоящими».

Категория

/api/category

Метод GET. Получение списка категорий и определенной категории

Поля категории для получения

Название	Тип	Используется в search	Описание
Id	Int		Идентификатор категории
Name	String	+	Название
Description	String		Описание
ParentId	Int		Идентификатор родительской категории
Path	String		Путь. Идентификаторы категорий с корневой по текущую через разделитель “|”
BreadCrumbs	String		Путь. Названия категорий с корневой по текущую через разделитель «/»

Получение списка категорий

Вызов

GET
/api/category?serviceid={serviceid}&search={searchstring}&fields={fieldlist}&pagesize={value}&page={value}&isTaskHintMode=true
Список категорий возвращается в отсортированном виде (по уровню иерархии).

Возможные параметры

serviceid *– необязательный параметр.*

Позволяет получить список категорий, привязанных к сервису. Чтобы получить список категорий для назначения на заявку, нужно передать в serviceid идентификатор сервиса этой заявки.

fields *- перечень полей ресурса «Категории» для вывода (из таблицы)*

Например:

1) Выведем список полных названий (с иерархией) всех категорий
   GET: /api/category?fields=Name
2) Выведем список идентификаторов и названий всех категорий сервиса 31
   GET: /api/category?serviceid=31&fields=Id,Name

search *– строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке.*

Например:

Выведем все категории, у которых в названии содержится подстрока «Test»:
GET: /api/category?search=Test

page, ***pagesize*** *– параметры отображения таблицы*

isTaskHintMode *– включает режим показа в виде подсказки выбора, как на карточке заявки. Работает только со значение serviced > 0.*

Пример вывода в xml

GET: /api/category

```
<CategoryList>
    <Categories>
        <Category>
            <Id>22</Id>
            <Name>Аппаратная ошибка</Name>
            <Description/>
            <ParentId/>
            <Path>22|</Path>
            <BreadCrumbs>Аппаратная ошибка</BreadCrumbs>
        </Category>
        <Category>
            <Id>23</Id>
            <Name>Программная ошибка</Name>
            <Description/>
            <ParentId/>
            <Path>23|</Path>
            <BreadCrumbs>Программная ошибка</BreadCrumbs>
        </Category>
        <Category>
            <Id>25</Id>
            <Name>Проблемы с сетью</Name>
            <Description/>
            <ParentId/>
            <Path>25|</Path>
            <BreadCrumbs>Проблемы с сетью</BreadCrumbs>
        </Category>
    </Categories>
    <Paginator>
        <Count>3</Count>
        <Page>1</Page>
        <PageCount>1</PageCount>
        <PageSize>25</PageSize>
        <CountOnPage>3</CountOnPage>
    </Paginator>
</CategoryList>
```

Получение определенной категории

Вызов

GET /api/category/{categoryid}

Где **{categoryid}** – идентификатор категории.

Пример вывода в xml

/api/category/22

```
<CategoryView>
    <BreadCrumbs>Аппаратная ошибка</BreadCrumbs>
    <Description/>
    <Id>22</Id>
    <Name>Аппаратная ошибка</Name>
```

```
        <ParentId i:nil="true"/>
        <Path>22|</Path>
</CategoryView>
```

Актив

/api/asset

Метод GET. Получение списка активов и конкретного актива

Поля актива для получения

Название	Тип	Используется в search	Описание
Id	Int	+	Идентификатор актива
InventoryNumber	String	+	Инвентарный номер
Name	String	+	Наименование
TypeId	Int		Идентификатор типа актива
ParentId	Int		Идентификатор родительского актива
Path	String	+	Путь. Идентификаторы активов с корневого по текущий через разделитель “|”
TypeName	String		Название типа
FullParentName	String		Полное наименование родителя (вся иерархия имен через разделитель «/»)
OwnerId	Int		Идентификатор владельца
Data	xml	+	Значения дополнительных полей актива
Changed	DateTime		Дата изменения актива

Получение списка активов

Вызов

GET /api/asset?serviceid={serviceid}&fields={fieldlist}&search={search}&pagesize={value}&page={value}

Возможные параметры

serviceid *– необязательный параметр.*

Позволяет вернуть активы, привязанные к сервису с идентификатором {serviced}
Пример: Выведем все активы сервиса 32
GET: /api/asset?serviceid=32

fields *– список полей, которые вы хотите получить (указаны в таблице)*

Пример:
Вывести для всех активов идентификаторы, названия и идентификаторы типа
GET: /api/asset?fields=Id,Name,TypeId

search *- строка поиска. Ищет данную строку как подстроку в полях, указанных в таблице в соответствующей колонке, а также в имени владельца актива и в названии его компании*

pagesize, ***page*** *– параметры отображения таблицы*

isTaskHintMode *– включает режим показа в виде подсказки выбора, как на карточке заявки. Работает только со значение serviced > 0.*

Пример вывода в xml

GET: /api/asset

```
<AssetList>
	<Assets>
		<Asset>
			<Id>43</Id>
			<InventoryNumber>12356</InventoryNumber>
			<Name>Office 13</Name>
			<TypeId>5</TypeId>
			<ParentId/>
			<Path>43|</Path>
			<TypeName>Room</TypeName>
			<FullParentName/>
			<OwnerId>43</OwnerId>
			<Changed>20.11.2015 11:33:29</Changed>
			<Data><field id="15">13</field></Data>
		</Asset>
		<Asset>
			<Id>42</Id>
			<InventoryNumber>1234</InventoryNumber>
			<Name>Office 12</Name>
			<TypeId>5</TypeId>
			<ParentId/>
			<Path>42|</Path>
			<TypeName>Room</TypeName>
			<FullParentName/>
			<OwnerId>43</OwnerId>
			<Changed>20.11.2015 11:33:19</Changed>
			<Data><field id="15">12</field></Data>
		</Asset>
	</Assets>
	<Paginator>
		<Count>2</Count>
		<Page>1</Page>
		<PageCount>1</PageCount>
		<PageSize>25</PageSize>
		<CountOnPage>2</CountOnPage>
	</Paginator>
</AssetList>
```

Получение определенного актива

Вызов

GET: /api/asset/{assetid}
Где {assetid} – идентификатор актива

Пример вывода в xml

GET: /api/asset/49

```
<AssetView>
        <Id>49</Id>
        <InventoryNumber>K123-45</InventoryNumber>
        <Name> K123-45: тестовое значение</Name>
        <TypeId>15</TypeId>
        <ParentId/>
        <Path>49|</Path>
        <TypeName>test 1</TypeName>
        <FullParentName/>
        <OwnerId>70</OwnerId>
        <Changed>28.09.2015 14:16:16</Changed>
        <Data><field id="54" >тестовое значение</field></Data>
</AssetView>
```

Метод POST. Добавление новых активов.

Поля актива для создания

Название	Тип значения	Обязательность	Комментарий
TypeId	int	Да	Идентификатор типа актива
InventoryNumber	string	Да	Инвентарный номер
OwnerId	int	Нет	Идентификатор владельца актива. Значение учитывается только если у текущего пользователя есть право просматривать пользователей. Иначе возьмется текущий пользователь.
ParentId	int	Нет	Идентификатор родительского актива
IsArchive	bool	Нет	Признак архивный
Field{id}	string	Нет	Значение дополнительного поля с идентификатором, например, Field8
Comment	string	Нет	Примечание

Вызов и пример использования

POST: api/asset

Пример:
Добавим в систему актив с инвентарным номером «R12542-45» и значением дополнительного поля типа «Строка» с Id=47:
POST: /api/asset

{“TypeId”: 12, “InventoryNumber”:”R12542-45”, “Field47”:”значение поля”}

Метод PUT. Изменение актива

Поля актива для изменения

Название	Тип значения	Комментарий
InventoryNumber	string	Инвентарный номер
OwnerId	int	Идентификатор владельца актива. Значение учитывается только если у текущего пользователя есть право просматривать пользователей. Иначе будет игнорироваться
ParentId	int	Идентификатор родительского актива
IsArchive	bool	Признак архивный
Field{id}	string	Значение дополнительного поля с идентификатором, например, Field8
Comment	string	Примечание

Вызов и пример использования

PUT: api/asset/id
Где id – идентификатор актива.

Пример:
Изменим для актива с Id = 345 инвентарный номер и владельца на пользователя с Id = 12:
PUT: /api/asset/345

{ “InventoryNumber”:”R222222”, “OwnerId”:12}

Тип актива

/api/assettype

Метод GET. Получение списка типов актива и конкретного типа актива.

Поля типа актива для получения

Название	Тип	Описание
Id	Int	Идентификатор типа актива
Name	String	Наименование
ExternalId	Int	Идентификатор типа актива в других системах (при синхронизации активов)
IsArchive	Int	Признак архивный
Image16Url	String	Путь до изображения
InventoryNumberInName	Bool	Использовать ли инвентарный номер в формировании активов данного типа
AssetTypeFields	-	Список дополнительных полей типа актива **(только при получении определенного актива)**

Получение списка типов актива

Вызов

GET: /api/assettype?fields={fieldList}&archive={value}&pagesize={value}&page={value}

Возможные параметры

fields *- перечень полей ресурса «Тип актива» для вывода (из таблицы)*

Например:

Выведем список названий и ссылок на изображение для всех типов актива

GET: /api/assettype?fields=Name,Image16Url

archive – *позволяет вывести заархивированные типы актива*

Возможные значения:

true/false

page, pagesize *– параметры отображения таблицы*

Пример вывода в xml

GET: /api/assettype?fields=Id,Name,InventoryNumber&pagesize=2

```
<AssetTypeList>
    <AssetTypes>
        <AssetType>
            <Id>5</Id>
            <Name>Room</Name>
        </AssetType>
        <AssetType>
            <Id>6</Id>
            <Name>Workstation</Name>
        </AssetType>
    </AssetTypes>
```

```
        <Paginator>
                <Count>10</Count>
                <Page>1</Page>
                <PageCount>5</PageCount>
                <PageSize>2</PageSize>
                <CountOnPage>2</CountOnPage>
        </Paginator>
</AssetTypeList>
```

Получение определенного типа актива

Вызов

GET /api/assettype/{assettypeid}
Где **{assettypeid}** – идентификатор типа актива

Вывод дополнительного поля актива AssetTypeFieldView в списке AssetTypeFields

Поля AssetTypeFieldView

Название	Тип	Описание
Id	Int	Идентификатор дополнительного поля
FieldTypeId	Int	Тип дополнительного поля: 1 – строка, 2 – число, 3 – дата, 4 – чекбокс, 5 – выпадающий список, 6 – файл, 7 – текст, 8 – пустой интервал, 9 – произвольный html-элемент
Hint	String	Подсказка
Name	String	Название
Unit	String	Единица измерения
IsRequired	Bool	Обязательность поля
FieldSelectValues	List of string	Список значений для типа дополнительного поля «Выпадающий список» (5)

Пример вывода в xml

GET: /api/assettype/16

```
        <AssetTypeView>
                <AssetTypeFields>
                        <AssetTypeFieldView>
                                <FieldSelectValues>
                                        <d4p1:string>один</d4p1:string>
                                        <d4p1:string> два</d4p1:string>
                                        <d4p1:string> три</d4p1:string>
                                </FieldSelectValues>
                                <FieldTypeId>5</FieldTypeId>
                                <Hint i:nil="true"/>
                                <Id>55</Id>
                                <IsRequired>false</IsRequired>
                                <Name>Тестм список</Name>
                                <Unit i:nil="true"/>
                        </AssetTypeFieldView>
                        <AssetTypeFieldView>
                                <FieldSelectValues i:nil="true"/>
```

```
                <FieldTypeId>1</FieldTypeId>
                <Hint i:nil="true"/>
                <Id>56</Id>
                <IsRequired>false</IsRequired>
                <Name>Тем строка</Name>
                <Unit i:nil="true"/>
            </AssetTypeFieldView>
            <AssetTypeFieldView>
                <FieldSelectValues i:nil="true"/>
                <FieldTypeId>4</FieldTypeId>
                <Hint i:nil="true"/>
                <Id>57</Id>
                <IsRequired>false</IsRequired>
                <Name>Тесm чек</Name>
                <Unit i:nil="true"/>
            </AssetTypeFieldView>
        </AssetTypeFields>
        <ExternalId i:nil="true"/>
        <Id>16</Id>
        <Image16Url>http://localhost/IntraService4.40/img/assets/ava1.jpg</Image16Url>
        <InventoryNumberInName>true</InventoryNumberInName>
        <IsArchive>false</IsArchive>
        <Name>test type</Name>
    </AssetTypeView>
```

Общие настройки системы

/api/settings

Метод GET. Получение списка общих настроек системы

Поля для получения общих настроек

Название	Тип	Описание
Key	String	Ключ-идентификатор параметра
Value	String	Значение параметра
FieldTypeId	Int	Тип значения параметра (1-строка, 2-число, 3-булевый, 4-список через запятую)
Name	String	Название параметра
Description	String	Описание параметра

Получение списка общих настроек

Вызов

GET /api/settings?keys={keys_list}

Возможные параметры

keys *– необязательный параметр*

Позволяет получить настройки по списку ключей-идентификаторов настроек (ключи должны быть перечислены через запятую).

Пример:

Выведем информацию по следующим настройкам: «максимально допустимый размер файла» и «Отправлять уведомления инициатору»
GET: /api/settings?keys=maxfilesize,EditorTaskNotificationEvent

Пример вывода в xml

```
<ArrayOfSettingsView>
    <SettingsView>
        <Description i:nil="true"/>
        <FieldTypeId>2</FieldTypeId>
        <Key>maxfilesize</Key>
        <Name>Максимально допустимый размер файла, в КБ</Name>
        <Value>7000</Value>
    </SettingsView>
    <SettingsView>
        <Description i:nil="true"/>
        <FieldTypeId>3</FieldTypeId>
        <Key>EditorTaskNotificationEvent</Key>
        <Name>Отправлять уведомления инициатору</Name>
        <Value>True</Value>
    </SettingsView>
</ArrayOfSettingsView>
```

Токен устройства пользователя

/api/token

Метод POST. Добавление токена текущему пользователю

Поля для создания токена

Название	Тип данных	Обязательность	Описание
Token	string	+	Токен устройства, полученный с сервера Apple или Google
DeviceType	string	+	Тип устройства, строка “iOS” или “Android”
DeviceName	String		Название устройства. Отображается в интерфейсе настроек пользователя

Вызов и пример использования

POST: /api/token

Пример:

Добавим для текущего пользователя токен устройства Android «*dfkjDJFHd84948hdkd7hdjfkdf*» с именем «Мой телефон»:

Json : {“ Token *”: “dfkjDJFHd84948hdkd7hdjfkdf”, “* DeviceType *”:”Android”, “* DeviceName *”: ”Мой телефон ”}*

Дополнительная информация

Пользователь, для которого добавляется токен, определяется по данным авторизации в заголовке запроса.
В ответе на запрос возвращается стандартный код состояния HTTP, показывающий статус добавления (успешно или ошибка).
У пользователя может быть несколько токенов устройств. Допускаются повторные запросы на добавление одного и того же токена одному и тому же пользователю. В этом случае возвращается признак успешного добавления и больше ничего не происходит. Сервер сам отслеживает уникальность токенов у пользователей, т.е. при добавлении токена какому-то пользователю этот токен ищется и удаляется у всех других пользователей.
Добавленные токены отображаются на карточке пользователя во вкладке «Настройки».

Метод GET. Удаление токена у текущего пользователя

Поля для удаления токена

Название	Тип данных	Обязательность	Описание
Token	string	+	Токен устройства, полученный с сервера Apple или Google
DeviceType	string	+	Тип устройства, строка “iOS” или “Android”

Вызов и пример использования

GET: /api/token?token=<deviceToken>&devicetype=<devicetype>

Пример:
Удалим для текущего пользователя токен устройства Android «*dfkjDJFHd84948hdkd7hdjfkdf*»
GET: /api/token?token= dfkjDJFHd84948hdkd7hdjfkdf &devicetype=Android

Дополнительная информация

В ответе на запрос возвращается стандартный код состояния HTTP, показывающий статус добавления (успешно или ошибка).
Пользователь, для которого удаляется токен, определяется по данным авторизации в заголовке запроса. У чужих пользователей удалить токен нельзя. Если указанный токен для указанного типа устройства и текущего пользователя не найден – возвращается ошибка.

Тестовое push-уведомление для устройства пользователя

api/push

Метод POST. Добавление тестового Push-уведомления

Поля для добавления тестового push-а для устройства

Название	Тип данных	Обязательность	Описание
Token	string	+	Токен устройства, полученный с сервера Apple или Google

DeviceType	string	+	Тип устройства, строка “iOS” или “Android”

Вызов и пример использования

POST: /api/push

Пример:

Добавим для устройства Android с токеном «*dfkjDJFHd84948hdkd7hdjfkdf*» текущего пользователя тестовое push-уведомление:

POST: /api/push

Json: {“ Token *”: “dfkjDJFHd84948hdkd7hdjfkdf”, “* DeviceType *”:”Android”}*

Дополнительная информация

Пользователь, для которого добавляется тестовое push-уведомление, определяется по данным авторизации в заголовке запроса.

Текст тестового уведомления: «Тестовое push-уведомление системы IntraService».

Push-уведомления отправляются с помощью службы IntraServiceAgent, поэтому время получения пуша будет зависеть от настроек периодичности отправки push-уведомлений в конфигурационном файле службы.