Worker API

Приложение делает http-запросы к 1С:ERP. При успехе, получает ответ с телом в json-формате.

Оглавление

[Данные о пользователе 2](#_Toc85487804)

[Запрос (GET) 2](#_Toc85487805)

[Тело ответа 2](#_Toc85487806)

[Хеширование пароля (Kotlin) 2](#_Toc85487807)

[Пример 2](#_Toc85487808)

[Данные об операции 3](#_Toc85487809)

[Запрос (GET) 3](#_Toc85487810)

[Тело ответа 4](#_Toc85487811)

[Пример 4](#_Toc85487812)

[Выполнение операции 5](#_Toc85487813)

[Запрос (POST) 5](#_Toc85487814)

[Тело ответа 6](#_Toc85487815)

[Пример 6](#_Toc85487816)

[Список операций 7](#_Toc85487817)

[Запрос (GET) 7](#_Toc85487818)

[Тело ответа 7](#_Toc85487819)

[Пример 7](#_Toc85487820)

[Итоги операций 8](#_Toc85487821)

[Запрос (GET) 8](#_Toc85487822)

[Тело ответа 8](#_Toc85487823)

[Пример 8](#_Toc85487824)

# Данные о пользователе

## Запрос (GET)

<Базовый адрес>/user?id=<Штрих-код: String>

## Тело ответа

{

"Код": <Код пользователя: String>,

"Наименование": <Имя пользователя: String>,

"Пароль": <hash пароля: String>

}

## Хеширование пароля (Kotlin)

fun String.hash() = this.*toCharArray*()  
 .*mapIndexed* **{** i, c **->** ((c.toInt() \* 13 - (i + 1) \* 7) % 873).toChar()**}** .*toCharArray*().*concatToString*()

## Пример

Запрос

<http://10.0.2.2:8080/user?id=59_012-4-1_02>

Тело ответа

{

"Код": "12-0000223",

"Наименование": "Иванов Иван Иванович",

"Пароль": "ks853gf"

}

# Данные об операции

## Запрос (GET)

<Базовый адрес>/operation?barcode=<Штрих-код операции: String>

## Тело ответа

{

"Номер": <Код операции: String>,

"Наименование": <Наименование операции: String>,

"Вид работ": <Наименование вида работ: String>,

"Количество": <Количество операций: Float>,

"Ед. изм.": <Наименование единицы измерения: String>

}

## Пример

Запрос

<http://10.0.2.2:8080/operation?barcode=59_012-4-1_02>

Тело ответа

{

"Номер": "001",

"Наименование": "Обивка дивана",

"Вид работ": "Обивка дивана Классический 140",

"Количество": 2.0,

"Ед. изм.": "шт"

}

# Выполнение операции

## Запрос (POST)

<Базовый адрес>/operation?id=<Код операции: String>

## Тело ответа

{

"Номер": <Код операции: String>,

"Наименование": <Наименование операции: String>,

"Вид работ": <Наименование вида работ: String>,

"Количество": <Количество операций: Float>,

"Ед. изм.": <Наименование единицы измерения: String>,

"Выполнено": <Дата и время выполнения: yyyy-mm-ddThh:mm:ss>

}

## Пример

Запрос

[http://10.0.2.2:8080/operation?id=001](http://10.0.2.2:8080/operation?id=59_012-4-1_02)

Тело ответа

{

"Номер": "001",

"Наименование": "Обивка дивана",

"Вид работ": "Обивка дивана Классический 140",

"Количество": 2.0,

"Ед. изм.": "шт",

"Выполнено": "2021-10-11T14:06:08"

}

# Список операций

## Запрос (GET)

<Базовый адрес>/operations?id=<Код пользователя: String>&skip=<Количество записей пропустить: Float>&top=<Количество записей считать: Float>&orderby=<Сортировка операций[[1]](#footnote-1): String>

## Тело ответа

[

{

"Номер": <Код операции: String>,

"Наименование": <Наименование операции: String>,

"Вид работ": <Наименование вида работ: String>,

"Количество": <Количество операций: Float>,

"Ед. изм.": <Наименование единицы измерения: String>,

"Выполнено": <Дата и время выполнения: yyyy-mm-ddThh:mm:ss>,

"Расценка": <Расценка за выполнение 1 операции[[2]](#footnote-2): Float>,

"Сумма": <Сумма начисленной заработной платы: Float>

},

…

]

## Пример

Запрос

<http://10.0.2.2:8080/operations?id=12-0000223>&skip=0&top=125&orderby=Выполнено desc

Тело ответа

[

{

"Номер": "001",

"Наименование": "Обивка дивана",

"Вид работ": "Обивка дивана Классический 140",

"Количество": 2.0,

"Ед. изм.": "шт",

"Выполнено": "2021-10-11T14:06:07"

},

{

"Номер": "004",

"Наименование": "Обивка кресла",

"Вид работ": "Обивка кресла Классический 140",

"Количество": 2.0,

"Ед. изм.": "шт",

"Выполнено": "2021-10-11T14:06:08",

"Расценка": 101.0,

"Сумма": 202.0

}

]

# Итоги операций

## Запрос (GET)

<Базовый адрес>/totals?id=<Код пользователя: String>&start=<Дата начала периода: yyyy-mm-dd>&end=<Дата окончания периода: yyyy-mm-dd >&analytics =<Вид аналитики: String>

Параметр «Вид аналитики» может принимать одно из следующих значений:

* «вид» - получение итогов с группировкой по видам работ,
* «день» - получение итогов с группировкой по календарным дням,
* «месяц» - получение итогов с группировкой по месяцам.

## Тело ответа

[

{

"Наименование": <Объект аналитики[[3]](#footnote-3)>,

"Сумма": <Сумма начисленной зарплаты: Float>

},

…

]

## Пример

Запрос с группировкой по видам работ

<http://10.0.2.2:8080/totals?id=12-0000223>&start=2021-10-01&end=2021-10-31&analytics=вид

Тело ответа

[

{

"Наименование": "Обивка дивана 1",

"Сумма": 20000.0

},

{

"Наименование": "Обивка дивана 3",

"Сумма": 30000.0

}

]

Запрос с группировкой по дням

<http://10.0.2.2:8080/totals?id=12-0000223>&start=2021-10-04&end=2021-10-10&analytics=день

Тело ответа

[

{

"Наименование": "2021-10-04",

"Сумма": 20000.0

},

{

"Наименование": "2021-10-05",

"Сумма": 30000.0

}

]

Запрос с группировкой по месяцам

<http://10.0.2.2:8080/totals?id=12-0000223>&start=2021-01-01&end=2021-12-31&analytics=месяц

Тело ответа

[

{

"Наименование": "2021-09-01",

"Сумма": 60000.0

},

{

"Наименование": "2021-10-01",

"Сумма": 70000.0

}

]

1. Параметр «Сортировка операций» будет всегда иметь значение «Выполнено desc» - упорядочить записи по полю «Выполнено» в порядке убывания. [↑](#footnote-ref-1)
2. Красным выделены поля, которые выводятся по операциям, вошедшим в выработку. В противном случае данные поля отсутствуют. [↑](#footnote-ref-2)
3. Значение поля «Объект аналитики» зависит от указанного в запросе вида аналитики. Для вида аналитики «вид» в поле указывается наименование вида работ (String). Для других видов аналитики – дата в формате yyyy-mm-dd. Причем, для вида «месяц» указывается дата любого дня месяца группировки. [↑](#footnote-ref-3)