

MIS_GBBookingAPI

Описание API интеграции платформы GBooking с Медицинскими Информационными Системами.

Rev v1.1

Authors: Vitaliy Sverchkov, Alex Zhukov

Last Modified: 10.12.2018



Содержание

Содержание	1
Общая схема взаимодействия.	2
Механизмы интеграции	3
1. Базовый механизм	3
1.1. Запрос на обновление схемы бизнеса.	4
1.2. Запрос на обновление расписания.	5
1.3. “Ленивая” загрузка записей.	7
1.3.1. Получение списка идентификаторов “ленивых” записей	7
1.3.2. Загрузка записей (создание, изменение).	8
1.4. Получение статуса обработки запроса.	10
2. Push механизм	11
2.1. Обработчик запроса на резервирование записи	12
2.2. Обработчик запроса подтверждение записи	13
2.3. Обработчик запроса на удаление записи	15
2.4. Обработчик запроса на изменение записи.	16
2.5. Обработчик запроса на изменение клиента в записи.	17
3. Callback механизм	18
3.1. Изменение расписания работника	19
3.1.1. Создание/изменение/удаление записи на стороне МИС.	20
3.2. Создание записи на стороне МИС.	20
3.3. Изменение записи на стороне МИС.	23
3.4. Удаление записи на стороне МИС.	24
3.5. Перевод записи в состояние не подтверждено на стороне МИС.	25
3.6. Изменение статуса доставки записи.	26
4. Приложение А (дополнительные callback вызовы)	27
4.1. Удаление сотрудника (из филиала/из сети).	27
4.2. У сотрудника удалена специальность (на филиале/на сети).	28
CHANGELOG	29

Общая схема взаимодействия.

Интеграция с системой GBooking производится через публичное API. Безопасность данных обеспечивается протоколом шифрования данных ssl поверх http (https).

Взаимодействие осуществляется через протокол https в обе стороны.

Интеграция предполагает, что расписание клиники в режиме реального времени будет доступно для записи конечному клиенту. При этом записи будут появляться в вашей медицинской информационной системе. Любые их изменения и отмена записи также будет отображаться на стороне используемой вами МИС.

Для интеграции сервиса GBooking и МИС используется 3 механизма:

1. Базовый механизм.
Синхронизация всего объема данных по расписанию. Гарантирует, что в конечном счете данные будут синхронизированы.
2. Push механизм.
Отправка запросов на резервирование/подтверждение/изменение/отмену записи. HTTP сервер на стороне клиники/МИСа должен принимать эти запросы, синхронно применять изменения и БД клиники и возвращать ответ.
3. Callback механизм.
Изменения расписания, добавление записей на стороне клиники должны отправлять запросы на GBooking API, чтобы решетка расписания была актуальной.

Разработка разделена на 2 этапа:

1. 1-я итерация - пп.1.1, 1.2., раздел 2 (Push механизм). После нее запускаемся.
2. 2-я итерация - п.3. Callback механизм.

Базовый GBooking API URL - <https://apiv2.gbooking.ru/rpc>

Тип запроса - HTTP/1.1 POST

Обязательный HTTP заголовок - Content-Type: application/json

Формат запроса - json-rpc

С примерами запросов так же можно ознакомиться по ссылке

<https://documenter.getpostman.com/view/608267/6nATBX4>

Переменные тестового окружения:

BusinessID: 4000000005820

Token: c368eaebc29967dd5b7bdf5eae6fc6658e73cc0f

UserID: 59955503fb1d89051031d672

Для получения авторизационных данных, ssl сертификатов и специфических настроек бизнеса вначале процесса разработки интеграции необходимо обратиться к сотрудникам GBooking.

На первой итерации необходимо предоставить запрос для проверки работоспособности сервиса ("ping" запрос) или указать запрос, который будет выполнять его роль. Данный запрос необходим нам для мониторинга доступности вашего HTTP сервера с целью своевременного исправления ошибок передачи данных (он не для проверки, например, записи в бд). "ping" запрос будет выполняться раз в 10 минут. Поэтому сделайте его максимально легким - без лишних библиотек, просто выдача 200 статуса и строки "ok".

Обратите внимание!

В случае выполнения "тяжелых" запросов, например, обновление схемы бизнеса (пп.1.1), расписания (пп.1.2) задача синхронизации уходит на предназначенные для этого сервера. В ответ на http запрос, в таком случае, приходит jobID, по которой можно узнать статус выполнения задачи, используя для этого запрос `business.get_bg_job_info` (пп.1.4). Данный метод вы можете использовать как в отладочных целях, так и с целью мониторинга работоспособности интеграции на вашей стороне. Ожидать мгновенного обновления бизнеса, расписания или чего-то еще после получения ответа с jobID не нужно - задача может выполняться от нескольких секунд до нескольких минут.

Время в параметрах запроса/ответа передается в формате ISO8601, например, 2017-03-30T00:00:00.000Z.

Механизмы интеграции

1. Базовый механизм

Полная синхронизация схемы бизнеса (списка работников, услуг и их связи), полная синхронизация расписания всех работников, загрузка всех актуальных записей. Каждый из этих запросов выполняется по расписанию раз в несколько минут. Может быть использован для восстановления актуальности расписания для failover сценария и при первом запуске синхронизации.

Предназначен для поддержания стабильной работы интеграции, например, в случае потери соединения при передаче частичных данных на какое-либо событие в МИСе через Callback механизм (см. п.3).

1.1. Запрос на обновление схемы бизнеса.

Обновление списка работников, списка услуг, а также привязки работника к услуге. Запрос либо создает соответствующие сущности, либо обновляет, найдя предварительно их по ид, указанном в запросе.

Рекомендация по частоте вызова запроса - 6-24 часа.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.update_business",
  "params": {
    "provider": "ident", // данный параметр указывает, что данные поступают уже
    // во внутреннем представлении (обязательно)
    "queue": "queue_name", // произвольное имя очереди (не заполнять,
    // передается по согласованию)
    "business": {
      "id": "{{BusinessID}}" // идентификатор бизнеса (обязательно)
    },
    "taxonomies": [
      {
        "isDefaultService": false, // дочерняя услуга, выбираемая по умолчанию,
        // если запись необходимо вести на корневую (по умолчанию false)
        "id": "extra_test_tax1", // идентификатор МИС услуги (обязательно)
        "name": "Тестовая услуга", // название услуги (обязательно)
        "description": "Описание тестовой услуги", // описание (по умолчанию
        // пустая строка)
        "parentId": null, // идентификатор МИС родительской услуги (по умолчанию
        // корневая таксономия)
        "order": null, // положение услуги в списке (по умолчанию добавляется в
        // конец списка, при изменении не меняется)
        "duration": 45, // продолжительность услуги в минутах (обязательно)
        "priceAmount": 1200.0, // стоимость полного времени услуги (обязательно)
        "priceCurrency": "RUB", // валюта (по умолчанию RUB)
        "stockAmount": 900, // цена со скидкой (необязательно)
        "priceType": "equal", // тип цены ("equal" – точная, "begin_with" – "от",
        // "average" – средняя) (необязательно)
        "additionalDurations": [ // массив с набором альтернативных длительностей
        // услуги для различных работников (необязательно)
        {
          "level": 5, // условный уровень работника. Должен быть указан у
          // работника в resources.taxonomyLevels (обязательно)
          "duration": 45 // длительность, минут (обязательно)
        }
      ]
    }
  }
}
```

```

    }],
    "resources": [{
        "id": "extra_res_id", // идентификатор МИС врача (обязательно)
        "fname": "Иван", // имя врача (обязательно)
        "mname": "Иванович", // отчество (по умолчанию пустая строка)
        "lname": "Иванов", // фамилия (обязательно)
        "description": "Отличный врач!", // описание (по умолчанию пустая строка)
        "order": null, // положение врача в списке (по умолчанию добавляется в
        // конец списка, при изменении не меняется)
        "taxonomies": ["extra_test_tax1"], // список идентификаторов МИС услуг,
        // которые выполняет врач (обязательно)
        "taxonomyChildren": [{ // является ли для врача данная услуга детской
        // (возможно указать 2 элемента с одинаковым taxonomyID) (необязательно)
            "taxonomyID" : "mis_tax_id",
            "children" : true|false,
        }],
        "taxonomyLevels": [{ //массив установки альтернативной длительности
        // услуги для работника (необязательно)
            "level": 5, // условный уровень работника, (обязательно)
            "id": "1" // идентификатор МИС услуги (обязательно)
        }]
    }]
}
}

```

Формат ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "queued": true,
    "jobID": "5805e88a4f292d520da38b52"
  }
}

```

Для работы со скидками ознакомьтесь с соответствующими запросами в [документации по онлайн записи](#).

1.2. Запрос на обновление расписания.

Расписание работников обновляется целиком для всех указанных работников. Если работник до этого имел свободное время на день, не указанный в запросе, то он будет стерт. Другими словами, в этом запросе расписание каждого работника нужно передавать целиком за тот период, за который вы хотите предоставить онлайн запись. Обратите внимание, что зачастую нет необходимости включать расписание больше, чем на месяц, поскольку клиент спустя месяц может даже не вспомнить, что он записывался.

Работники, не указанные в запросе не изменяют своего расписания.

При передаче OFF интервалов они имеют больший приоритет, чем ON, что можно использовать для учета занятого времени (существующих записей, обедов, перерывов в работе). При этом такие интервалы можно передавать как есть как OFF интервалы.

На практике, большая часть изменений в расписании происходит в ближайшие дни, поэтому имеет смысл передавать расписания на ближайшую неделю, например, чаще, чем на месяц. Чтобы обновить расписание только на необходимый временной интервал и не стереть расписание на последующие дни передавайте параметр `skipNotPassedDays: true`. Это является лишь рекомендацией и лишь для тех случаев, когда получение расписание на стороне МИСа является высоконагруженной операцией.

Обратите внимание! Время в `resources.timetable` выгружается во временной зоне UTC. Есть возможность выгружать время во временной зоне бизнеса. Для этого укажите параметр запроса `isLocalTime: true`.

При указании временных интервалов значение времени начала временного интервала, указанное в поле `start`, должно быть строго меньше времени конца — поле `end`.

Рекомендация по частоте вызова запроса - 30 секунд - 7 минут.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "method": "mis.update_schedule",
  "params": {
    "provider": "{{Provider}}",
    "queue": "custom_queue_name", // произвольное имя очереди (не заполнять,
    // передается по согласованию)
    "skipNotPassedDays": true|false, // оставить ли свободное расписание для
    // тех дней, которые не были указаны в запросе (необязательно, по умолчанию false)
    "business": {
      "id": "{{BusinessID}}"
    },
    "isLocalTime": true, // если передано true, то время считается локальным
    // временем
    "resources": [{
```

```

        "id": "extra_res_id", // идентификатор МИС врача (обязательно)
        "timetable": [{ // список слотов врачей
            "start": "2016-10-21 09:00:00.000Z",
            "end": "2016-10-21 10:00:00.000Z",
            "type": "ON"|"OFF", // включить/выключить время расписания (по
умолчанию ON)
            "appointmentId": "mis_id" // ид записи, которой соответствует интервал
(не обязательно, по умолчанию не передается)
        }]
    }]
}
}

```

Формат ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "queued": true,
    "jobID": "5805e88a4f292d520da38b52"
  }
}

```

1.3. “Ленивая” загрузка записей.

При обновлении расписания пп.1.2. во временном интервале есть возможность передать идентификатор записи - поле appointmentId. В случае, если передан идентификатор записи, которой еще нет в системе, она будет зарегистрирована для загрузки и предоставлена в списке при следующем вызове mis.get_lazy_appointments. Таким образом, вызывая последовательно методы mis.get_lazy_appointments и mis.load_appointments мы загружаем новые записи.

Механизм “ленивой” загрузки будет корректно выполняться в случае восстановления после отказа одного или нескольких из компонентов, участвующих в процессе синхронизации либо канала передачи данных. Таким образом, он гарантирует отказоустойчивость процесса синхронизации. Также данный механизм будет применяться и при инициализации данных (при первом запуске синхронизации).

Рекомендация по частоте вызовов - 1-10 минут.

1.3.1. Получение списка идентификаторов “ленивых” записей

Запрос - mis.get_lazy_appointments.

Возвращает список идентификаторов записей, которые были указаны для загрузки в запросе формирования расписания, но еще не были загружены

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "method": "mis.get_lazy_appointments",
  "params": {
    "provider": "{{Provider}}",
    "business": {
      "id": "{{BusinessID}}"
    },
    "network": {
      "id": "{{NetworkID}}"
    }
  }
}
```

Формат ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "appointments": ["mis_id1", "mis_id2", ... "mis_idN"]
  }
}
```

1.3.2. Загрузка записей (создание, изменение).

Запрос - mis.load_appointments.

Загружает данные по новым записям.

Количество элементов в массиве appointments не должно превышать 500.

Формат запроса (аналогичен 3.1, за исключением, что передается список записей и данные находятся внутри элементов массива appointments):

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.load_appointments",
  "params": {
    "provider": "{{Provider}}",
    "business": {
      id: "4000000003543" // бизнес ид (обязательно)
    },
    "appointments": [{
      "active": true,
      "id": "mis_appointment_id", // идентификатор записи (обязательно)
      "start": "2017-01-26T15:45:00.000Z", // время начала записи
      "duration": 135, // продолжительность записи
      "price": { // цена записи
        "amount": 1500,
        "currency": "RUB",
        "originalAmount": 1500
      },
      "client": { // данные клиента (обязательно)
        "id": "mis_client_id", // идентификатор клиента. Нужен для поддержания
        // ссылочной целостности клиентской базы (опционально, если не передан
        // идентификация происходит по переданному номеру телефона)
        "name": "Василий",
        "surname": "Пупкин",
        "phone": [{
          "country_code": "7",
          "area_code": "000",
          "number": "0300000"
        }]
      },
      "source": "MIS_SCHEDULE_SYNC", // источник записи (обязательно)
      "taxonomy": {
        "id": "mis_extra_id" // идентификатор МИСа услуги/отделения
        // (обязательно)
      },
      "resource": {
        "id": "mis_extra_id" // идентификатор МИСа врача (обязательно)
      },
    }]
  }
}

```

Формат успешного ответа:

```

{
  "jsonrpc": "2.0",

```

```
"result": true
}
```

Формат ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  "id": 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}
```

1.4. Получение статуса обработки запроса.

Запрос на обновление схемы бизнеса делегируется на сервера “тяжелых” задач. Поэтому статус выполнения запроса (успешно или нет) не может быть получен сразу. Для получения статуса такого запроса создается задача, идентификатор которой должен быть получен в ответе. Затем для получения информации по этой задаче используется отдельный запрос.

Формат запроса:

```
{
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "jsonrpc": "2.0",
  "method": "business.get_bg_job_info",
  "params": {
    "jobID": "5805e88a4f292d520da38b52" // идентификатор задачи
  }
}
```

Формат ответа:

```
{
  "jsonrpc": "2.0",
  "result": {
    "job": {
      "created": "2016-10-18T09:16:58.298Z",

```

```

        "state": "pending", // статус выполнения задачи. Принимает одно из
возможных значений ("initial", "pending", "done", "error")
        "jobID": "5805e88a4f292d520da38b52"
    }
}
}

```

В случае, когда задача завершилась с ошибкой появляется дополнительный атрибут "error":

Пример ответа с ошибкой:

```

{
  "jsonrpc": "2.0",
  "result": {
    "job": {
      "created": "2016-10-18T00:03:11.359Z",
      "state": "error",
      "jobID": "580566bf3253100519408b22",
      "error": {
        "data": {
          "code": "ETIMEDOUT"
        },
        "message": "Failed to update business integration data",
        "code": -10600
      }
    }
  }
}

```

2. Push механизм

Проброс записи из GB в МИС, установленный на стороне клиники, чтобы администратор и другие роли в МИСе видели эту запись. Также проброс любых изменений, отмены записи.

В заголовке Authorization передается токен Basic авторизации, сформированный из пары логин/пароль соответствующей паре user/token, отправляемые к GBookingAPI в запросах на синхронизацию. Эти данные могут быть изменены по согласованию сторон.

2.1. Обработчик запроса на резервирование записи

Передаваемые параметры - время начала записи, ид врача, ид услуги (или длительность, цена).

Возвращаемое значение - ид записи.

В случае, если время, указанное в записи занято (есть пересечение переданного временного интервала с занятым), то необходимо вернуть статус ошибки.

Обратите внимание, что резервирование лишь временное ограничение на запись на данное время к данному ресурсу (работнику, врачу). В случае, если клиент не завершит бронирование записи запросом на подтверждение необходимо удалить резерв. Время жизни должно быть ограничено не менее 5 минутами. Очистка резервов должна производиться на стороне клиники/МИСа, кто реализует Push механизм.

Обратите внимание, что если вы указали параметр isDefaultService для услуги в запросе mis.update_business и имеет место запись на корневую услугу, то в этом запросе (резервирования) и запросе на подтверждение записи будет использован идентификатор той дочерней услуги, которой была установлена эта опция. Если она была установлена нескольким услугам, в рамках одной категории, то может быть выбрана любая (на практике будет выбрана первая по списку). Другими словами, необходимо указывать только одну услугу по умолчанию.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "method": "mis.reserve_appointment",
  "params": {
    "provider": "ident",
    "business": {
      "id": "{{BusinessID}}"
    },
    "appointment": {
      "resourceID": "mis_res_123",
      "taxonomyID": "mis_tax_123",
      "start": "2016-12-28T15:00:00.000Z",
      "end": "2016-12-28T15:30:00.000Z"
    }
  }
}
```

```
}  
}
```

Формат ответа:

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "status": true,  
    "appointmentID": "{{индекс из МИС}}"  
  }  
}
```

Пример ответа с ошибкой:

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "status": false,  
    "errorMessage": "{{Error}}"  
  }  
}
```

2.2. Обработчик запроса подтверждение записи

Передаваемые параметры - **ид записи**, **ФИО клиента**, **телефон клиента** (дополнительные параметры — *пол, дата рождения*).

Возвращаемое значение - статус подтверждения.

При реализации данного метода, в зависимости от особенностей конкретной МИС и конкретных бизнес задач вы можете использовать поле `appointment.comment` для указания данных клиента. Они представлены в человекопонятной форме и могут быть изменены (добавлены дополнительные данные, изменено представление данных в строке). Данное поле следует применять для того, чтобы указать данные клиента в записи в свободной форме - например, в виде комментария к записи. Не следует извлекать данные клиента из этого поля (парсинг), тк оно может быть изменено. Для использования данных клиента, например, для создании записи в бд, следует использовать данные, указанные в поле `client`.

В случае включенной на виджете оплаты клиентом данные платежа будут переданы в поле `payment`.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "method": "mis.confirm_appointment",
  "params": {
    "provider": "{{Provider}}",
    "business": {
      "id": "{{BusinessID}}"
    },
    "appointment": {
      "comment": "Василий Пупкин,+71110001010", // данные клиента в виде строки
      (ФИО клиента, телефон клиента); разделитель - запятая
      "id": "mis_app_id",
      "resourceID": "mis_res_123",
      "taxonomyID": "mis_tax_123"
    },
    "client": { // данные клиента
      "name": "Василий",
      "surname": "Пупкин",
      "phone": [{
        "country_code": "7",
        "area_code": "000",
        "number": "0300000"
      }]
    },
    "payment": { // данные по оплате
      "status": "YES_PAYED_ONLINE", // Для online платежа принимает значение
      "YES_PAYED_ONLINE"
      "type": "cash", // одно из возможных значений cash, card
      "system": "YandexMoney", // одна из платежных систем
      "payed": true, // была ли оплата
      "externalInvoice": "transaction_id", // идентификатор транзакции в
      платежной системе
      "amount": 124.23, // сумма
      "issuedOn": "2016-12-28T15:00:00.000Z", // Дата создания платежа
      "payedOn": "2016-12-28T15:01:00.000Z" // Дата оплаты платежа
    }
  }
}
```

Формат ответа:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "status": true,
    "appointmentID": "mis_app_id",
    "clientID": "mis_client_id" // для поддержания ссылочной целостности
    клиентской базы (опционально)
  }
}
```

Пример ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "status": false,
    "errorMessage": "{{Error}}"
  }
}
```

2.3. Обработчик запроса на удаление записи

Запрос на удаление записи выполняется только для подтвержденных записей.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "method": "mis.delete_appointment",
  "params": {
    "provider": "{{Provider}}",
    "business": {
      "id": "{{BusinessID}}"
    },
    "appointment": {
      "id": "mis_app_123" // Идентификатор из МИС
    }
  }
}
```

```
}  
}
```

Формат ответа:

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "status": true,  
    "appointmentID": "{{индекс из МИС}}"  
  }  
}
```

Пример ответа с ошибкой:

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "status": false,  
    "errorMessage": "{{Error}}"  
  }  
}
```

2.4. Обработчик запроса на изменение записи.

Данный запрос возникает в случае, когда через БэкОфис (личный кабинет администратора GBooking) происходит изменение параметров записи. **Если БэкОфис GBooking не используется, то нет необходимости в реализации данного метода.**

Данный запрос предполагает передачу следующих параметров записи:

- Время начала - параметр appointment.start
- Время окончания (длительность) - параметр appointment.end
- Услуга, на которую записались - параметр appointment.taxonomyID
- Работник, на которого записались - параметр appointment.resourceID

Данный запрос вызывается в случае, если на стороне GBooking вышеуказанные параметры были изменены. Если в записи на стороне GBooking был изменен клиент, то вызывается метод mis.change_appointment_client.

Формат запроса:

```
{  
  "jsonrpc": "2.0",
```

```

    "id": 1,
    "cred": {
      "token": "{{Token}}",
      "user": "{{UserID}}"
    },
    "method": "mis.update_appointment",
    "params": {
      "provider": "ident",
      "business": {
        "id": "{{BusinessID}}"
      },
      "appointment": {
        "id": "mis_app_id",
        "resourceID": "mis_res_123",
        "taxonomyID": "mis_tax_123",
        "start": "2016-12-28 15:00:00.000Z",
        "end": "2016-12-28 15:30:00.000Z"
      }
    }
  }
}

```

Формат ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "status": true,
    "appointmentID": "{{индекс из МИС}}"
  }
}

```

2.5. Обработчик запроса на изменение клиента в записи.

Данный метод вызывается в случае изменения клиента в записи.

Формат запроса:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },

```

```

"method": "mis.change_appointment_client",
"params": {
  "provider": "{{Provider}}",
  "business": {
    "id": "{{BusinessID}}"
  },
  "appointment": {
    "comment": "Василий Пупкин,+71110001010", // данные клиента в виде строки
    (ФИО клиента, телефон клиента); разделитель - запятая
    "id": "mis_app_id"
  },
  "client": { // данные клиента
    "name": "Василий",
    "surname": "Пупкин",
    "phone": [{
      "country_code": "7",
      "area_code": "000",
      "number": "0300000"
    }]
  },
}

```

Формат ответа:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "status": true,
    "appointmentID": "mis_app_id",
    "clientID": "mis_client_id" // для поддержания ссылочной целостности
    клиентской базы (опционально)
  }
}

```

Пример ответа с ошибкой:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "status": false,
    "errorMessage": "{{Error}}"
  }
}

```

3. Callback механизм

Вызов GBooking API Callback на изменение решетки расписания на стороне МИС:

- изменение расписания работника (обновление списка “исключений” в расписании работника)
- создание/изменение/удаление записи на стороне МИС

На оба этих события следует вызывать соответствующие методы GBooking API. По возможности необходимо делать это асинхронно основной транзакции, используя существующие механизмы.

3.1. Изменение расписания работника

Запрос `mis.patch_resource_timetable`.

Данный запрос берет текущее расписание работника и применяет временные отрезки, указанные самом запросе.

Обработчик этого запроса после добавления новых временных интервалов оптимизирует внутреннее представление расписания таким образом, чтобы пересекающиеся временные интервалы вычитались друг из друга, если они имеют разные типы ON и OFF, а примыкающие и имеющие одинаковый тип объединялись в один. Таким образом, можно не бояться линейного роста данных от количества запросов.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.patch_resource_timetable",
  "params": {
    "business": {
      "id": "{{BusinessID}}" // идентификатор бизнеса (обязательно)
    },
    "resource": {
      "id": "extra_res_id", // идентификатор МИС врача (обязательно)
      "timetable": [{ // список слотов врачей
        "start": "2016-10-21 09:00:00.000Z",
        "end": "2016-10-21 10:00:00.000Z",
        "type": "ON" // тип интервала. Допустимые значения - OFF, ON
      }]
    }
  }
}
```

```
}
```

* Передача параметров `sync_op_order` и `sync_op_created` используется при использовании очереди синхронизации.

Формат успешного ответа:

```
{
  "jsonrpc": "2.0",
  "result": true
}
```

Формат ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  "id": 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}
```

3.1.1. Создание/изменение/удаление записи на стороне МИС.

Запрос `mis.patch_resource_timetable` (см. пп. 3.1).

Здесь есть важное замечание, которое сильно упростит разработку. В запросе есть атрибут `type`, принимающий значения ON/OFF. Например, в случае создания записи вам нужно указать интервал OFF, в случае удаления - интервал ON, в случае изменения - интервал ON для старых временных рамок записи и OFF для новых.

Важное замечание! При использовании данного запроса на создание/изменение/удаление записи изменяется расписание работника. Это значит, в частности, что в этом случае нельзя будет получить уведомления на указанные события, полную статистику по записям, и другие фишки, связанные с записями, как существующие, так и будущие. В этом случае используйте запросы в пп 3.2, 3.3, 3.4.

3.2. Создание записи на стороне МИС.

Запрос `mis.reserve_appointment`.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.reserve_appointment",
  "params": {
    "business": {
      "id": "40000000003543" // бизнес ид (обязательно)
    },
    "appointment": {
      "active": true,
      "id": "mis_appointment_id", // идентификатор записи (обязательно)
      "start": "2017-01-26T15:45:00.000Z", // время начала записи
      "duration": 135, // продолжительность записи
      "price": { // цена записи
        "amount": 1500,
        "currency": "RUB",
        "originalAmount": 1500
      },
    },
    "source": "MIS_SCHEDULE_SYNC", // источник записи (обязательно)
    "taxonomy": {
      "id": "mis_extra_id" // идентификатор МИСа услуги/отделения
      (обязательно)
    },
    "resource": {
      "id": "mis_extra_id" // идентификатор МИСа врача (обязательно)
    },
    "client": { // данные клиента (обязательно*)
      "id": "mis_client_id", // идентификатор клиента. Нужен для поддержания
      // ссылочной целостности клиентской базы (опционально, если не передан
      // идентификация происходит по переданному номеру телефона)
      "name": "Василий",
      "surname": "Пупкин",
      "phone": [{
        "country_code": "7",
        "area_code": "000",
        "number": "0300000"
      }]
    }
  }
}
```

Формат запроса:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.reserve_appointment",
  "params": {
    "business": {
      "id": "40000000003543" // бизнес ид (обязательно)
    },
    "appointments": [{
      "active": true,
      "id": "mis_appointment_id", // идентификатор записи (обязательно,
одинаковый для всех записей в массиве)
      "start": "2017-01-26T15:45:00.000Z", // время начала записи
      "duration": 135, // продолжительность записи
      "price": { // цена записи
        "amount": 1500,
        "currency": "RUB",
        "originalAmount": 1500
      },
      "taxonomy": {
        "id": "mis_extra_id" // идентификатор МИСа услуги/отделения
(обязательно)
      },
      "resource": {
        "id": "mis_extra_id" // идентификатор МИСа врача (обязательно)
      },
    }],
    "source": "MIS_SCHEDULE_SYNC", // источник записи (обязательно)
    "client": { // данные клиента (обязательно*)
      "id": "mis_client_id", // идентификатор клиента. Нужен для поддержания
ссылочной целостности клиентской базы (опционально, если не передан
идентификация происходит по переданному номеру телефона)
      "name": "Василий",
      "surname": "Пупкин",
      "phone": [{
        "country_code": "7",
        "area_code": "000",
        "number": "03000000"
      }]
    }
  }
}

```

Формат успешного ответа:

```

{
  "jsonrpc": "2.0",

```

```
"result": true
}
```

Формат ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  "id": 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}
```

3.3. Изменение записи на стороне МИС.

Запрос `mis.update_appointment`.

```
{
  "jsonrpc": "2.0",
  "id": 17,
  "cred": {
    "token": "{{Token}}",
    "user": "{{UserID}}"
  },
  "method": "mis.update_appointment",
  "params": {
    "business": {
      "id": "{{BusinessID}}"
    },
    "appointment": {
      "id": "appl",
      "start": "2017-02-26 12:55:00.000Z"
    }
  }
}
```

Формат успешного ответа:

```
{
  "jsonrpc": "2.0",
  "result": true
}
```

Формат ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  "id": 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}
```

Данный запрос следует вызывать в следующих случаях:

- изменение времени начала записи
- изменение длительности записи

Формат запроса/ответа аналогичный как и при создании записи (см. пп. 3.2.)

Обязательными параметрами для авторизации и нахождения записи являются:

- cred
- params.business.id
- params.appointment.id

Другие параметры являются опциональными. Если они указаны, то они будут применены к найденной записи, если нет - то нет.

3.4. Удаление записи на стороне МИС.

Запрос `mis.cancel_appointment_by_business`.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.cancel_appointment_by_business",
  "params": {
    "business": {
      "id": "12345" // бизнес ид (обязательно)
    },
    "appointment": {
      "id": "mis_appointment_id", // идентификатор записи (обязательно)
    }
  }
}
```

```
}  
}
```

Формат успешного ответа:

```
{  
  "jsonrpc": "2.0",  
  "result": true  
}
```

Формат ответа с ошибкой:

```
{  
  "jsonrpc": "2.0",  
  "id": 7,  
  "error": {  
    "code": -10000, // код ошибки  
    "message": "Unknown error occurred" // текстовое описание  
  }  
}
```

3.5. Перевод записи в состояние не подтверждено на стороне МИС.

Запрос `mis.set_appointment_tentative`

Формат запроса:

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "cred": {  
    "token": "{{Token}}", // лонг лив токен (обязательно)  
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)  
  },  
  "method": "mis.set_appointment_tentative",  
  "params": {  
    "business": {  
      "id": "40000000003543" // бизнес ид (обязательно)  
    },  
    "appointment": {  
      "id": "mis_appointment_id", // идентификатор записи (обязательно)  
    }  
  }  
}
```

В случае успешного ответа запись будет переведена в состояние TENTATIVE, скрыта из бэкофиса. Если в течении 5 минут не придёт запрос на создание записи с таким же mis_appointment_id - запись будет удалена

Формат успешного ответа:

```
{
  "jsonrpc": "2.0",
  "result": true
}
```

Формат ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  id: 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}
```

3.6. Изменение статуса доставки записи.

Запрос mis.set_appointment_delivery_status.

В случае, если запись поступает в МИС асинхронно (опосредованно, используя какую-либо очередь), необходимо обновить статус доставки при обновлении ее на платформе, предоставляющей доступ к записи.

Формат запроса:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.set_appointment_delivery_status",
  "params": {
    "business": {
      "id": "{{BusinessID}}" // бизнес ид (обязательно)
    },
    "appointment": {
      "id": "mis_appointment_id", // идентификатор записи (обязательно,
      // одинаковый для всех записей в массиве)
    }
  }
}
```

```

        "exchangeID": "new_mis_appointment_id", // новый идентификатор записи,
в случае, если необходимо изменить его (необязательно)
        "deliveryStatus": "CONFIRMED_DELIVERED", // статус доставки записи в
рабочий инстанс МИС (RESERVED_DELIVERED – в случае доставки записи как резерва,
CONFIRMED_DELIVERED – в случае успешной доставки подтвержденной записи,
NOT_DELIVERED – в случае, если запись не была доставлена ни в каком виде)
    },
}
}

```

4. Приложение А (дополнительные callback вызовы)

В данном приложении описаны дополнительные callback вызовы на события, наступающие значительно реже, нежели описанные в п. 3 Callback механизм.

4.1. Удаление сотрудника (из филиала/из сети).

Запрос mis.update_resource.

Формат запроса:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.update_resource",
  "params": {
    "business": {
      "id": "{{BusinessID}}" // идентификатор бизнеса (обязательно*)
    },
    "network": {
      "id": "{{NetworkID}}" // идентификатор сети (обязательно*)
    },
    "resource": {
      "id": "mis_extra_id", // идентификатор врача из клиники
      (обязательно)
      "status": "ACTIVE"|"INACTIVE" // статус врача (обязательно)
    }
  }
}

```

* должен быть указан либо идентификатор сети, либо идентификатор бизнеса

Формат успешного ответа:

```
{
```

```

    "jsonrpc": "2.0",
    "result": true
}

```

Формат ответа с ошибкой:

```

{
  "jsonrpc": "2.0",
  id: 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}

```

4.2. У сотрудника удалена специальность (на филиале/на сети).

Запрос mis.update_resource.

Формат запроса:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "{{Token}}", // лонг лив токен (обязательно)
    "user": "{{UserID}}" // идентификатор профиля пользователя (обязательно)
  },
  "method": "mis.update_resource",
  "params": {
    "business": {
      "id": "{{BusinessID}}" // идентификатор бизнеса (обязательно*)
    },
    "network": {
      "id": "{{NetworkID}}" // идентификатор сети (обязательно*)
    },
    "resource": {
      "id": "mis_extra_id", // идентификатор врача из клиники
      (обязательно)
      "taxonomies": ["tax1", {$add:"tax2"}, {$remove:"tax3"}, ...] //
      идентификаторы МИС услуг (обязательно**)
    }
  }
}

```

Формат успешного ответа:

```
{
  "jsonrpc": "2.0",
  "result": true
}
```

Формат ответа с ошибкой:

```
{
  "jsonrpc": "2.0",
  id: 7,
  "error": {
    "code": -10000, // код ошибки
    "message": "Unknown error occurred" // текстовое описание
  }
}
```

****** в случае, если в качестве элементов массива указаны строки, то устанавливается переданный массив. Если же передано хотя бы одно действие, то массив обрабатывается как массив действий (строковый идентификатор будет интерпретироваться как операция добавления).

CHANGELOG

Версия 1.1 (2018-12-10):

1. убраны пункты с пометкой draft
2. убраны 2 устаревших запроса:
 - a. mis.confirm_appointment
 - b. mis.clear_tentative_appointment
3. Убрана устаревшая секция discounts в запросе mis.update_business. Вместо нее добавлена ссылка на документ описание работы со ссылками (папка в постман коллекции).
4. Добавлен пример запроса mis.update_appointment
5. Изменен значение параметра source в примерах запросов на MIS_SCHEDULE_SYNC
6. Изменена нумерация запросов в Разделе 3 (3.2.3 -> 3.5, 3.3.1 -> 3.6).