

# GBooking API for Online Appointment Scheduling

GBooking API is a standard json-rpc2 API supporting multiple transactions for managing business, clients, providing stats, booking time, and many more. It checks user acl roles for each request using the following parameters: `"cred": {"user": "...", "token": "..."}.`

The present documentation offers examples of requests for a test environment. You can run them for development purposes.

Example of request:

## Plain Text

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "business.get_top_services",
  "params": {
    "business": {
      "id": "4000000001992"
    }
  }
}
```

- **jsonrpc** — protocol version (2.0)
- **id** — numeric value identifying the request at the server
- **method** — name of requested method
- **params** — array of parameters for requested method
- **cred** — set of data for identification and authorization

Example of successful response:

## Plain Text

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "status": true,
    "services": []
  }
}
```

Example of error response:

## Plain Text

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -10200,
    "message": "Failed to retrieve business",
    "data": ""
  }
}
```

By access level, all the requests are classified into several basic roles:

- **guest** — public request without specifying access parameters
- **client** — request providing and managing the data of a specific client
- **admin** — request with access to the data pertaining to a business or a set of businesses

For each request, there is a description including its access role. There can be multiple access roles. In such a case, the request can return different results.

For online booking, public methods\* with the *guest* role are used.

---

*\* except for the method of showcase booking. The easiest way to get authorized in this case is to obtain a long live token in the showcase's back office.*

## Glossary

Below are the basic concepts of the *GBooking API for Online Appointment Scheduling*:

- A **Business** contains complete data about a business:
  - list of employees
  - list of services
  - settings of GBooking Widget's display
  - settings of GBooking BackOffice display
  - business information (name, address, phone number, working schedule)
  - list of other branches in the network
- **Business Branch** or **Branch** stand for one branch, subsidiary, shop, salon, etc.
- **Business Showcase** or **Showcase** provides access to the data of a random group of branch businesses
- **Branch Network** means a set of branches united into one. The GBooking Widget allows the user to switch between branches or to obtain data for all the branches. In the BackOffice, an administrator with sufficient access rights can switch between branches. The client data is shared throughout the branch network.
- An **Employee** contains information about the employee:
  - full name
  - list of services provided by them
  - per-service performance speed
  - branch

- phone number and email
- additional textual information, including profession and further information about the employee
- **Service, Taxonomy** — service or class of services that can be signed up for. Contains name, description, duration (optional for every employee), price, and discounts.
- **Timetable slot** — a period available for making appointments
- **Appointment** — an element of the timetable containing the following information: *client*, *employee*, *starting date*, *total duration*, *list of services*, *whether the client has paid*, and so on.
- **Reserve, Reserve appointment** — provisional appointment that reserves the required time for a short period
- **Client** contains data about a client of one business or a branch network

For detailed information about the data schema, please refer to repository: <https://github.com/GbookingLTD/corev2-schemata/tree/master/docs>.

Data schemata in the json-schema format are contained here: <https://github.com/GbookingLTD/corev2-schemata/tree/master/schemas>.

Error codes:

- [full list of errors](#)
- [authorization errors](#)
- [errors from external systems](#)
- There are lists of possible errors for every method to be used.

As of 02.10.2018, the request-response format has been described for the following methods:

- Getting Business Data (business.get\_profile\_by\_id)
- Adding a Client (client.add\_client)

---

## Getting Business Data

The first step consists in getting data about employees and services, as well as detailed information about the business, including its name, address, coordinates, phone number, working hours, etc.

---

### POST Getting static business data

<https://apiv2.gbooking.ru/rpc>

Access roles: **guest**, **admin**.

A business schema contains a large amount of data. For the Widget to be displayed, the most important elements are as follows:

- List of services — taxonomies array
- List of resources — resources array

Pseudocode for displaying list of specialists:

#### Plain Text

```
function renderWorkers(business) {
  for (worker in business.resources) {
    if (worker.status === "ACTIVE" && worker.displayInWidget) {
      render(worker);
    }
  }
}
```

Here, business means the data obtained from the request.

Every employee can have a **scheduleIsEmpty** field. If it equals to **true**, it means that the employee does not have free timetable. In this case, there is no need to display such an employee or display a message stating that no appointments for this employees can be made.

Another possible situation is when an employee does not have free timetable for a specific service that he/she provides. In this case, the **scheduleIsEmpty** field will not equal to **true** (since there is actually timetable but for other services). You can use the request for *getting next appointment date* to specify if this employee has free timetable available for a specific service.

The **worker\_sorting\_type** parameter in the body of the request refers to the employee sorting type. It can take one of the following values:

- "none" (as is)
- "workload" (by workload, starting with the least loaded)
- "most\_free" (by closest appointment date)

The sorting shall be performed by increase of the **order** index in the **resources** array (in other words, resources[i].order).

The **only\_active\_workers** parameter in the body of the request means that only active employees available for appointments will be returned.

The **show\_inactive\_workers** parameter in the body of the request means that all the employees will be returned, not only the active ones.

By default, a **guest** access type request only returns active employees, while a request with **admin** access type covers all of them.

A request provides information on all the businesses/network branches in the **networks** field if the **with\_networks** parameter is used. This data should be used together with the business.get\_network\_data request.

Pricing information is contained in the **taxonomies** array.

See data descriptions [json-schema](#) (including field descriptions)

- BusinessTaxonomyPrice — for **price** field
- AdditionalBusinessTaxonomyPrice — for **additionalPrice** field

The **price** field contains the default price (unless a price has been found for the employee in **additionalPrices**). A price

range ("A to B") can be calculated based on **additionalPrices** and **price**, meaning the minimum and maximum prices for a service (considering the **price** field).

If the **price.amount** field equals to "0", then no price is defined for the corresponding service.

## Business Showcase

The following changes apply to the business data schema (data returned in response):

- **business.general\_info.isShowcase** should be "true" for the showcase
- **business.resources[i].origin\_general\_info** stores information about the business branch where the employee works (see Screenshot 1: branch-wise data in the list of employees was obtained from this field)
- **business.resources[i].originBusinessID** — branch business ID
- **business.resources[i].originTaxonomyIDs** — array of IDs of the services that brought the employee to the showcase (since only one service from one branch can be attached to a showcase service, there can be only one element like this, and it will at the same time be guaranteed)
- **business.taxonomies[i].showcases** — IDs of showcase connections

Please note that the **business.resources[i].taxonomies** array includes showcase service IDs for each employee. Therefore, you can use the data as if it were local (within one business branch).

## HEADERS

---

**Content-Type** application/json

## Body raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.get_profile_by_id",
  "params": {
    "business": {
      "id": "4000000006304"
    },
    "with_networks": true,
    "worker_sorting_type": "workload"
  }
}
```

https://apiv2.gbooking.ru/rpc

Access role: **guest**.

This method allows you to get a list of *IDs* for the branches available for making appointments under a pre-arranged source or the *GENERAL* source.

The list of business branch data shall be provided using the rpc method **business.get\_profile\_by\_id** performed with the **with\_networks** parameters for one of the branches in the network. The data for the branches shall be returned in the **networks** field, containing the minimum data required for displaying a list of branches.

To get a list of network branches IDs, take the data from **networkWidgetConfiguration** by the source you have been assigned or by the *GENERAL* source.

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {},
  "method": "business.get_network_data",
  "params": {
    "networkID": 315
  }
}
```

## Getting a Timetable

Getting a timetable is made possible using the CRAC (The Collection of Results for Appointments Calculations) subsystem.

CRAC handles binary vectors of the timetables (0 for busy, 1 for free). A CRAC vector encodes information about an employee's free time.

All the requests for CRAC described herein share one special nuance regarding the dates. The time value is not taken into consideration when indicating a date. The space can be filled with zeros.

In other words, if you want to specify a period between 2018-09-06 and 2018-09-10, you can write down the following two dates: 2018-09-06T00:00:00.000Z  
2018-09-10T00:00:00.000Z

This feature concerning the dates is because the CRAC subsystem saves a vector for an entire day.

To obtain timetable slots, you should first obtain business data using the **business.get\_profile\_by\_id** request.

---

## POST Getting a timetable from CRAC (deprecated)

<https://crac-prod3.gbooking.ru/rpc>

Access role: **guest**.

Use requests: "Getting timetable slots ..."

### HEADERS

Content-Type	application/json
--------------	------------------

Body raw

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {},
  "method": "Crac.GetCRACResourcesAndRooms",
  "params": [{
    "business": {
      "id": "4000000006304"
    },
    "filters": {
      "resources": ["59f05563854202eb6f86569c"],
      "date": {
        "from": "2017-11-01T00:00:00.000Z",
        "to": "2017-11-30T00:00:00.000Z"
      }
    }
  }]
}
```

---

## POST Getting showcase timetable slots

<https://cracslots.gbooking.ru/rpc>

Access role: **guest**.

This request sends a set of slots that are ready for rendering in the timetable and also for making appointments.

In this request, the **business** field contains an object with the following fields:

- **business.id** — showcase business ID
- **business.widget\_configuration.cracServer** — alias of a CRAC server
- **business.widget\_configuration.mostFreeEnable** — returns the closest appointment date for each employee
- **business.general\_info.timezone** — time zone of business

All these fields should be previously obtained from the **business.get\_profile\_by\_id** request (see above) and sent as is. Further on, as the data schema for this request might be extended, the format of the **business** field shall still be compatible with the format of the data from the **business.get\_profile\_by\_id** request.

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {

  },
  "method": "CracSlots.GetCRACDistributedResourcesAndRooms",
  "params": {
    "business": {
      "id": "4000000006304",
      "widget_configuration": {
        "cracServer": "CRAC_PROD3"
      },
      "general_info": {
        "timezone": "Europe/Moscow"
      }
    },
    "filters": {
      "resources": [
        {
          "resource": "5afafb9fc28333643ea29636",
          "business": {
            "id": "4000000006291"
          }
        },
        {
          "resource": "5afafb9fc28333643ea2963a",
          "business": {
```



```
        "id": "40000000006291"
      }
    },
    {
      "resource": "5afafb9fc28333643ea2963c",
      "business": {
        "id": "40000000006291"
      }
    }
  ],
  "taxonomies": [
    "9175610"
  ],
  "rooms": [

  ],
  "date": {
    "from": "2018-09-06T00:00:00.000Z",
    "to": "2018-09-10T00:00:00.000Z"
  }
}
```

## POST Getting branch timetable slots

<https://cracslots.gbooking.ru/rpc>

Access role: **guest**.

This request returns a set of slots ready for rendering in the timetable and also for making appointments.

For obtaining the duration of a service, see **getServiceDuration** (*Getting closest appointment date from CRAC for a branch*).

### HEADERS

**Content-Type** application/json

**Body** raw

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
```

```

},
"method": "CracSlots.GetCRACResourcesAndRooms",
"params":
{
    "business": {
        "id": "40000000006289",
        "widget_configuration": {
            "cracServer": "CRAC_PROD3",
            "mostFreeEnable": true
        },
        "general_info": {
            "timezone": "Europe/Moscow"
        }
    },
    "filters": {
        "resources": [
            { "id": "5afafbc373b6066443311f24", "duration": 30 },
            { "id": "5afafbc373b6066443311f26", "duration": 30 }
        ],
        "taxonomies": [
            "9175163"
        ],
        "rooms": [
        ],
        "date": {
            "from": "2018-07-13T00:00:00.000Z",
            "to": "2018-07-17T00:00:00.000Z"
        }
    }
}
}
}

```

## POST Getting date of closest appointment for showcase

<https://crac-prod3.gbooking.ru/rpc>

Access role: **guest**.

Returns dates of closest appointment regarding a service in a list of employees. Request for business showcase.

Input parameters (**params**):

- **business.id** — ID of showcase business
- **taxonomy.id** — ID of showcase taxonomy (service)
- **resources** — list of employee IDs

All the IDs refer to showcase data only.

The response returns an array of objects in the `result.Free` field. Each element has the following properties:

The response returns an array of objects in the `result.items` field. Each element has the following properties:

- **resource** — ID of resource
- **taxonomy** — ID of taxonomy
- **date** — date of closest appointment, in ISO format (hours, minutes and seconds are filled with zeros)
- **maxFreeMinutes** — duration of the longest free interval in the employee's timetable for the specified day, in minutes

The closest appointment date shall be calculated based on how long it takes each employee to provide a service. This value is then compared with **maxFreeMinutes**. After that, the closest day is found where the duration of the service is less than the **maxFreeMinutes** parameter. This finding will be now considered the date of the closest appointment (**date** parameter).

If the response contains an element with the following fields: `"date": "0001-01-01T00:00:00Z"`, `"maxFreeMinutes": 0`, then the corresponding employee does not have free timetable for the service in question.

Please note that if you need to sort by the closest appointment date, it can be done more conveniently using the **worker\_sorting\_type** parameter in the request for getting business static data, **business.get\_profile\_by\_id**. Sorting of the employees shall be performed by increase of the **order** index in the **resources** array (that is to say, `resources[i].order`).

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 9,
  "cred": {
  },
  "method": "Crac.CRACDistributedResourcesFreeByDate",
  "params": [
    {
      "business": {
        "id": "4000000006304"
      },
      "taxonomy": {
        "id": "9175167"
      },
      "resources": [
        "5afafbc373b6066443311f24",
        "5afafbda6f64a76438ecde86",
        "5afafb9fc28333643ea29636",
        "5afafb9fc28333643ea2963c",
        "5afafb9fc28333643ea29638",
        "5afafbc373b6066443311f26"
      ]
    }
  ]
}
```

```
}  
  
]  
}
```

## POST Getting date of closest appointment for a branch

<https://crac-prod3.gbooking.ru/rpc>

Access role: **guest**.

Returns dates of closest appointment regarding a service for a list of employees. Request for branch business.

Input parameters (**params**):

- **duration** — duration of the service provided by the employee
- **taxonomy.id** — ID of showcase taxonomy (service)
- **resources** — list of employees' IDs

For the format of response data, see Request for Showcase.

As the duration of the service, you can take the longest duration for each one of the employees in question.

The following Javascript code shows how to calculate the duration of a service performed by each employee:

javascript

```
export function getServiceDuration(taxonomy, resource) {  
  if (resource) {  
    var taxLevel = (_.find(resource.taxonomyLevels, {id: taxonomy.id}) || {}).level;  
    if (typeof taxLevel !== 'undefined') {  
      var level = _.find(taxonomy.additionalDurations, {level: taxLevel});  
      if (level) {  
        return level.duration ? level.duration : taxonomy.duration;  
      }  
    }  
  }  
  return taxonomy.duration;  
}
```

This code is also valid for a request for slots.

## HEADERS

Content-Type

application/json

Body raw

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {

  },
  "method": "Crac.CRACResourcesFreeByDate",
  "params": [
    {
      "taxonomy": {
        "id": "9175163"
      },
      "resources": [
        "5afafbc373b6066443311f24",
        "5afafbc373b6066443311f26"
      ],
      "duration": 60
    }
  ]
}

```

## POST Getting date of closest appointment for a branch, v.2

<https://crac-prod3.gbooking.ru/rpc>

Access role: **guest**.

Returns a list of closest appointment dates for a list of employees.

If a service ID is specified, the response contains the date of closest appointment for each employee regarding the service in question.

If no service ID is specified (empty string), the response contains the date of closest appointment for each employee regarding all the services he/she provides.

The previous version of this request shows how to get the duration of the service provided by a specific employee.

What's new in version 2:

- Option to send duration of service provided by each employee.
- Correct calculation of the closest appointment date, provided that a timetable for the corresponding day is available.
- **duration** — duration of the service provided by the employee
- **taxonomy.id** — ID of showcase taxonomy (service)
- **resources** — list of employees' IDs
- **taxonomy.id** — service ID
- **duration** — default duration of service

If it is known that it takes all employees the same time to perform a service, this duration can be sent instead of a list.

- **durations** — list of durations of services' performance by employees  
The i-th position represents the duration of service performance by the employee whose ID stands in the i-th position in the **resources** array.
- **location** — time zone of business, "Europe/Moscow" format  
Can be obtained from the **general\_info.timezone** of the business data schema.

For data format, see Request for Showcase.

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {},
  "method": "Crac.CRACResourcesFreeByDateV2",
  "params": [
    {
      "business": {
        "id": "40000000006289"
      },
      "taxonomy": {
        "id": "9175163"
      },
      "resources": [
        "5afafbc373b6066443311f24",
        "5afafbc373b6066443311f26"
      ],
      "duration": 60,
      "durations": [60, 30],
      "location": "Europe/Moscow"
    }
  ]
}
```

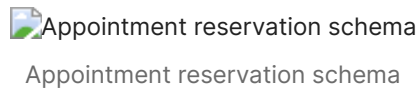
---

## Appointment Scheduling

The process of scheduling an appointment for a service comprises two stages: booking and confirmation of the appointment. Booking means that a time slot is reserved for a short time (about 5 minutes). This time is given to the user to complete the appointment or repeat the booking if the time has elapsed. At the 2nd stage (confirmation of the

appointment), the user enters his/her data into the field and clicks the confirmation button. Upon a successful response, the appointment becomes permanent.

The sequence of requests for reserving an appointment is shown on the following schema:



## POST Booking an appointment

<https://apiv2.gbooking.ru/rpc>

Access role: **guest**, **admin** (for showcase).

Request for booking an appointment. Creates an appointment marked as **TENTATIVE** for a short time (about 5 minutes). This time is given to the user to confirm the reserve. Otherwise, the booking is deleted.

- **appointment.start** — starting time of appointment (UTC for business time zone). Time zone cannot be changed (see example).
- *appointment.duration* — duration of appointment (if not submitted, is calculated as the aggregate duration of service performed by the employee in question)
- *appointment.price.amount* — appointment price value
- *appointment.price.currency* — appointment payment currency
- **appointment.source** — source of appointment  
You can indicate your own source of the appointments or one selected from a list, depending on where the appointment was made from. This will be included into the data regarding the appointment, the reports and the analytics.
- **business.id** — business ID
- **taxonomy.id** — service ID
- **resource.id** — employee ID
- **client\_appear** — client appearance status  
Can take one of the possible values — "YES\_APPEAR" (appeared), "NO\_APPEAR", (did not appear), "NONE" (not defined)
- *order.id* — order ID (if the appointment is added to an existing order)

Mandatory fields are marked in **bold**, optional fields are *italic*.

The full data schema for a request/response corresponds to the appointment data schema. You can make a test request, see what comes and set up the fields you need.

Possible error types:

- -10200 (BUSINESS\_RETRIEVE\_FAILED) — no business found by ID
- -13200 (TAXONOMY\_RETRIEVE\_FAILED) — taxonomy not found by ID
- -14100 (APPOINTMENT\_CREATE\_FAILED) — data specified incorrectly; validation error
- -14302 (APPOINTMENT\_CLIENT\_OVERLAPPED) — this client has already booked this time (there is an overlap

between an existing appointment and the one in question)

This error can appear when a corresponding option has been installed at BackOffice.

- -14500 (APPOINTMENT\_DATA\_MISSING) — a mandatory field skipped
- -14600 (APPOINTMENT\_RESOURCE\_TAXONOMY\_MISMATCH) — employee does not provide the specified service
- -14650 (APPOINTMENT\_RESOURCE\_BUSY) — for group appointments: group not found; for single appointments: employee not found or does not have free time
- -14700 (APPOINTMENT\_CAPACITY\_REACHED) — available space for service exceeded (time occupied)
- -14900 (APPOINTMENT\_TIME\_BEFORE\_NOW) — starting time stated in the request is already less than the current time at the business time zone
- -15200 (ORDER\_RETRIEVE\_FAILED) — order not found by *order.id*
- -15300 (ORDER\_UPDATE\_FAILED) — error while changing the specified order

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 19,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "appointment.reserve_appointment",
  "params": {
    "appointment": {
      "start": "2016-08-19T15:00:00",
      "duration": 30,
      "price": {
        "amount": 0,
        "currency": "RUB"
      }
    },
    "source": "{{Source}}",
    "business": {
      "id": "4000000006304"
    },
    "taxonomy": {
      "id": "9073638"
    },
    "client_appear": "NONE",
    "resource": {
      "id": "57b463de610c56383014c344"
    }
  }
}
```



## POST Booking an appointment with service from showcase

https://apiv2.gbooking.ru/rpc

Access role: **guest**, **admin** (for showcase).

This request should be used when working with a showcase business.

A request for booking an appointment should additionally include the branch business ID **originBusinessID**.

The branch business ID can be obtained from the employee's field in the showcase business schema (also called **originBusinessID**).

A mandatory condition is that the starting time of the slot during booking coincides with the starting time obtained from the request **CracSlots.GetCRACDistributedResourcesAndRooms**.

Requests for cancelling or modification of appointments should also include the branch business ID.

Possible error types are the same as for the *Booking an Appointment* request.

### HEADERS

Content-Type	application/json
--------------	------------------

Body raw

```
{
  "jsonrpc": "2.0",
  "id": 19,
  "cred": {
  },
  "method": "appointment.reserve_appointment",
  "params": {
    "originBusinessID": "{{OriginBusinessID}}",
    "appointment": {
      "start": "2018-02-01T13:00:00",
      "duration": 60,
      "price": {
        "amount": 0,
        "currency": "RUB"
      }
    }
  },
  "source": "GBOOKING_DESKTOP_WIDGET",
  "business": {
    "id": "4000000006304"
  }
}
```

```

    },
    "taxonomy": {
      "id": "9175160"
    },
    "client_appear": "NONE",
    "resource": {
      "id": "5afafbda764a76438ecde88"
    }
  }
}

```

## POST Adding a new client

<https://apiv2.gbooking.ru/rpc>

Access role: **guest**.

Adding a new client or obtaining an existing one.

This method starts with searching for an existing client. The search is made throughout the network covering the branch. If the business is a standalone one, the search is only performed within it.

The client's phone number is a mandatory search criterium. Additional criteria are the client's full name and email address.

If the client has not been found, it will be created for the specified branch business according to the data sent in the request.

- **business.id** — business ID
- **client.name** — client's first name
- **client.surname** — client's last name
- **client.phone[0].country\_code** — part of client's phone number — country code
- **client.phone[0].area\_code** — part of client's phone number — area/provider code
- **client.phone[0].number** — part of client's phone number — rest of number
- *client.email* — client's email address

Mandatory fields are marked in **bold**, optional fields are *italic*.

The full data schema of request/response corresponds to the client's data schema. You can make a test request, see what comes and set up the fields you need.

Possible error types:

- -11100 (CLIENT\_CREATE\_FAILED) — data specified incorrectly; validation error
- -11500 (CLIENT\_DATA\_MISSING) — mandatory information missing in the request

## HEADERS

**Body** raw

```
{
  "jsonrpc": "2.0",
  "id": 18,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "client.add_client",
  "params": {
    "business": {
      "id": "4000000006304"
    },
    "client": {
      "name": "new",
      "surname": "client",
      "phone": [
        {
          "country_code": "7",
          "area_code": "123",
          "number": "1231231"
        }
      ],
      "email": ["new_client@mail.com"]
    }
  }
}
```

**POST** Modifying an existing client

<https://apiv2.gbooking.ru/rpc>

Access role: **client**.

Modifying an existing client's data. Search for the client is performed by the ID submitted.

If a client has not been found, the CLIENT\_DATA\_MISSING error is returned.

- **client.id** — client ID
- **business.id** — business ID
- **client.name** — client's first name
- **client.surname** — client's last name
- **client.phone[0].country\_code** — part of client's phone number — country code
- **client.phone[0].area\_code** — part of client's phone number — area/provider code

- `client.phone[0].area_code` — part of client's phone number — area/provider code
- `client.phone[0].number` — part of client's phone number — rest of number
- `client.email` — client's email address

Mandatory fields are marked in **bold**, optional fields are *italic*.

The full request/response data schema corresponds to the client data schema. You can make a test request, see what comes and set up the fields you need.

Possible error types:

- -11300 (CLIENT\_UPDATE\_FAILED) — data specified incorrectly; validation error
- -11500 (CLIENT\_DATA\_MISSING) — mandatory information missing in request

## HEADERS

**Content-Type** `application/json`

**Body** raw

```
{
  "jsonrpc": "2.0",
  "id": 18,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "client.update_client",
  "params": {
    "business": {
      "id": "40000000006304"
    },
    "client": {
      "name": "test",
      "surname": "client",
      "phone": [
        {
          "country_code": "7",
          "area_code": "123",
          "number": "1231231"
        }
      ],
      "email": ["test_client@mail.com"]
    }
  }
}
```

https://apiv2.gbooking.ru/rpc

Access role: **guest**.

A request for confirmation of an appointment changes the status of a reserved appointment from **TENTATIVE** to **CONFIRMED\_BY\_CLIENT**.

- **appointment.id** — ID of a reserved appointment (marked as **TENTATIVE**)
- **client.id** — client's ID

At the moment when the request is performed, the reserved appointment can have been deleted since the lifetime of a reserve is about 5 minutes only. In this case, the **APPOINTMENT\_RETRIEVE\_FAILED** error shall be returned.

Every service has its "capacity", equalling to 1 by default. If you try to book or confirm an appointment for a service for the time that has been "filled", you will receive the **APPOINTMENT\_CAPACITY\_REACHED** error. If the "capacity" equals to 1, this error will mean that the time is already busy.

Possible error types:

- -10200 (**BUSINESS\_RETRIEVE\_FAILED**) — business not found by ID
- -11200 (**CLIENT\_RETRIEVE\_FAILED**) — client not found by ID
- -11800 (**CLIENT\_IN\_BLACKLIST**) — client is in blacklist
- -14200 (**APPOINTMENT\_RETRIEVE\_FAILED**) — appointment not found by ID
- -14300 (**APPOINTMENT\_UPDATE\_FAILED**) — data specified incorrectly; validation error
- -14302 (**APPOINTMENT\_CLIENT\_OVERLAPPED**) — this client has already made an appointment for this time (there is an overlap between an existing appointment and the one being confirmed)  
This error can appear when the corresponding option has been installed at BackOffice.
- -14500 (**APPOINTMENT\_DATA\_MISSING**) — a mandatory field is missing
- -14610 (**APPOINTMENT\_RESOURCE\_TAXONOMY\_UNAVAILABLE**) — employee does not provide the requested service
- -14700 (**APPOINTMENT\_CAPACITY\_REACHED**) — available space exceeded for service (time already busy)

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{  
  "jsonrpc": "2.0",  
  "id": 19,  
  "cred": {},
```

```
{
  "method": "appointment.client_confirm_appointment",
  "params": {
    "appointment": {
      "id": "58ff7a6fde30f0b622c24926"
    },
    "client": {
      "id": "57b467de3c27f82e221dade1"
    }
  }
}
```

## POST Cancelling of an appointment by client

<https://apiv2.gbooking.ru/rpc>

Access role: **guest**.

A request for a client's cancelling of an appointment changes its status for **CANCELLED\_BY\_CLIENT**.

- **appointment.id** — appointment ID
- **client.id** — client ID

Before requesting the method, make sure that the appointment has one of the following statuses:

CONFIRMED\_BY\_CLIENT, TENTATIVE, CONFIRMED\_BY\_BUSINESS. If the appointment's status is CANCELLED\_BY\_CLIENT or CANCELLED\_BY\_BUSINESS (was cancelled before), the method will return the APPOINTMENT\_CANCEL\_FAILED error.

Possible error types:

- -10200 (BUSINESS\_RETRIEVE\_FAILED) — business not found by ID (in appointment)
- -11200 (CLIENT\_RETRIEVE\_FAILED) — client not found by ID
- -14200 (APPOINTMENT\_RETRIEVE\_FAILED) — appointment not found by ID
- -14300 (APPOINTMENT\_UPDATE\_FAILED) — data specified incorrectly; validation error This error can appear when the corresponding option has been set at BackOffice.
- -14303 (APPOINTMENT\_CANCEL\_FAILED) — incorrect initial status of appointment; appointment cancelled before
- -14305 (APPOINTMENT\_PRE\_CANCEL\_FAILED) — appointment cancelled in an external system (e.g. in MIS)
- -14500 (APPOINTMENT\_DATA\_MISSING) — a mandatory field is missing
- -14700 (APPOINTMENT\_CAPACITY\_REACHED) — available space for service exceeded (time already busy)

## HEADERS

**Content-Type** application/json

**Body** raw

```
{
  "jsonrpc": "2.0",
  "id": 19,
  "cred": {},
  "method": "appointment.cancel_appointment_by_client",
  "params": {
    "appointment": { "id": "{{AppointmentID}}" },
    "client": { "clientID": "{{ClientID}}" }
  }
}
```

---

## POST Cancelling of an appointment by business

<https://apiv2.gbooking.ru/rpc>

Access role: **admin**.

A request for cancelling an appointment by a business changes its status for **CANCELLED\_BY\_BUSINESS**.

The input parameters, response parameters and possible error codes are the same as for the request for cancelling an appointment by client, except for the fact that authorization is also required in this case.

### HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 19,
  "cred": {
    "user": "5b1035dcaff15607133b523f",
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a"
  },
  "method": "appointment.cancel_appointment_by_business",
  "params": {
    "appointment": { "id": "{{AppointmentID}}" },
    "client": { "clientID": "{{ClientID}}" }
  }
}
```

---

## POST Un-reserving a reserved timetable slot

https://apiv2.gbooking.ru/rpc

Access role: **guest**.

Request for removing a reserve appointment. The reserve appointment should have the **TENTATIVE** status before this method is requested.

- **appointment.id** — appointment ID
- **business.id** — business ID

Possible error types:

- -14306 (APPOINTMENT\_GUEST\_CANCEL\_FAILED) — no appointment found by ID

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 19,
  "cred": {},
  "method": "appointment.client_remove_empty_appointment",
  "params": {
    "appointment": {
      "id": "58ff7a6fde30f0b622c24926"
    },
    "business": {
      "id": "40000000006304"
    }
  }
}
```

---

## POST Getting a list of appointments for showcase

https://apiv2.gbooking.ru/rpc

Access role: **admin**.



This request returns a list of appointments created for the showcase, applying the specified filter parameters.

- **business.id** — business ID
- **created.start**, **created.end** — select appointments by creation date. The difference in dates should not exceed 90 days
- **source** — source of appointment; the returned appointments should have this source
- **page** — page number, starting with 1
- **pageSize** — size of numbered page

Mandatory fields are marked in **bold**, optional fields are *italic*.

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "user": "5b1035dcaff15607133b523f",
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a"
  },
  "method": "appointment.get_appointment_by_showcase",
  "params": {
    "business": {
      "id": "40000000006304"
    },
    "pageSize": 100500,
    "page": 1,
    "source": "{{Source}}",
    "created": {
      "from": "2018-10-01T00:00:00",
      "to": "2018-11-01T00:00:00"
    }
  }
}
```

---

## POST Getting a list of appointments by business

<https://apiv2.gbooking.ru/rpc>

Access role: **admin**.

This request returns a list of appointments by business, applying the specified parameters

This request allows for the following filter options:

- **business.id** — business ID
- **network.id\*** — network ID
- **filter.start**, **filter.end** — choose appointments by starting time. The date difference should not exceed 90 days
- *filter.created.start*, *filter.created.end* — filter appointments by creation date. These filter parameters are of higher priority than **filter.start**, **filter.end**. The appointments shall be filtered by creation date if **filter.created.start** and **filter.created.end** have been submitted. The date difference should not exceed 90 days
- *filter.updated.start*, *filter.updated.end* — filter appointments by modification date. The date difference should not exceed 90 days.
- *filter.skipUpdated* — not to display modified appointments (**false** by default — modified appointments are displayed)
- *skipBusinessCancelled* — not to display cancelled appointments (**false** by default — cancelled appointments are displayed)
- **filter.services** — list of IDs for services for which appointments are to be received
- **filter.workers** — list of IDs for employees for which appointments are to be received
- **page** — page number starting with 1
- **pageSize** — size of numbered page

Mandatory fields are marked in **bold**, optional fields are *italic*.

\* one of these two parameters is mandatory: **business.id** or **network.id**.

Possible error types:

- -10001 (BUSINESS\_DATA\_MISSING) — no business or network ID specified in the request
- -14500 (APPOINTMENT\_DATA\_MISSING) — **page** and **pageSize** parameters not specified

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "user": "5b1035dcaff15607133b523f",
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a"
  },
  "method": "appointment.get_appointment_by_filter",
  "params": {
    "business": {
      "id": "40000000006304"
    }
  }
}
```

```

    },
    "network": {
      "id": "315"
    },
    "extraFilters": {
      "sort": [{
        "dir": "desc",
        "field": "created"
      }]
    },
    "pageSize": 100500,
    "page": 1,
    "skipBusinessCancelled": true,
    "filter": {
      "start": "2018-11-01T00:00:00",
      "end": "2018-11-07T23:59:59",
      "created": {
        "start": "2018-11-01T00:00:00",
        "end": "2018-11-07T23:59:59"
      }
    },
    "skipUpdated": false,
    "services": [],
    "workers": []
  }
}

```

## Showcase (Vitrina)

A showcase is a tool providing unified access to the data and appointments belonging to a set of branch businesses. Access to showcase businesses is similar to getting data about one branch (**business.get\_profile\_by\_id** request), except for some differences that are described below.

A showcase provides access to the data of real branch businesses but does not store the data in itself. Upon reservation, the data is transferred to the business. The showcase's users have no access to business data (only to public information).

To initialize showcase data, you should run a request for building a showcase (**business.build\_showcase**). There are two ways of retrieving it: specifying an Excel file for remapping the services or without doing so (importing taxonomies as is). After this method is retrieved, the consistency of the showcase data is maintained automatically, eliminating the need to retrieve it again. Also, this method is idempotent, meaning that it can be retrieved again to restore showcase data.

### POST Building a showcase for a network of businesses

<https://apiv2.gbooking.ru/rpc>

Access role: **admin**.

Building with importing taxonomies "as is" from network branches.

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.build_showcase",
  "params": {
    "taxonomyMatching": "name_parent",
    "network": {
      "id": "315"
    },
    "business": {
      "id": "4000000006304"
    }
  }
}
```

---

## POST Building a showcase for random set of branches

<https://apiv2.gbooking.ru/rpc>

Access role: **admin**.

Building with importing taxonomies "as is" from a random set of businesses.

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

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

```
{
  "id":1,
  "cred":{
    "token":"8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user":"5b1035dcaff15607133b523f"
  },
  "method":"business.build_showcase",
  "params":{
    "taxonomyMatching": "name_parent",
    "originBusiness": {
      "id": ["4000000006290", "4000000006291", "4000000006289"]
    },
    "business": {
      "id":"4000000006304"
    }
  }
}
```

## POST Building a showcase from an Excel file



<https://apiv2.gbooking.ru/rpc>

Access role: **admin**.

Building a showcase and remapping services via an XLSX file.

A **fileToken** can be obtained by loading a file through the curl [http://apiv2.gbooking.ru/upload\\_temp\\_file](http://apiv2.gbooking.ru/upload_temp_file) -X POST -F filename=@file.xlsx request.

### AUTHORIZATION Basic Auth

### HEADERS

**Content-Type** application/json

**Body** raw

```
{
  "jsonrpc":"2.0",
  "id":1,
  "cred":{
    "token":"8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user":"5b1035dcaff15607133b523f"
  },
  "method":"business.build_showcase",
  "params":{
```

```
    "taxonomyMatching": "excel",
    "fileToken": "43373ed5e874a93202731b7678d3079fae5ef4c6",
    "originBusiness": {

      "id": ["40000000006290", "40000000006291", "40000000006289"]
    },
    "business": {
      "id": "40000000006304"
    }
  }
}
```

## POST Page-by-page loading of employees

<https://apiv2.gbooking.ru/rpc>

Access role: **admin**.

### HEADERS

**Content-Type** application/json

### Body raw

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "resource.get_showcase_resources",
  "params": {
    "page": 0,
    "pageSize": 100500,
    "worker_sorting_type": "workload",
    "business": {
      "id": "40000000006304"
    },
    "taxonomy": {
      "id": "9165602"
    }
  }
}
```

## POST Verifying showcase integrity

https://apiv2.gbooking.ru/rpc

Access role: **admin**.

### HEADERS

---

**Content-Type** application/json

### Body raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.check_showcase",
  "params": {
    "business": {
      "id": "4000000006304"
    }
  }
}
```

## Discounts

The discounts are applied to services (taxonomies). They can be classified into two types:

- **regular discounts** — act for several weeks and are repeated on specific days of the week
- **day-specific discounts** — act for specific dates

Each one of these discount types has specific timeframes during the day when they are applied.

Example of **discountTimeFrameSchema** timeframe:

#### Plain Text

```
{
  "start": 540,
  "end": 600,
```

```
"discount": 100,  
"provider": "LOCAL"  
}
```

In this timeframe, the starting and ending times (the **start** and **end** fields, respectively) are specified in minutes elapsed from start of the day (for time zone of the business), while the **discount** field is the amount in the currency specified for the price of the service.

The client business logic for handling discounts has been implemented in Javascript in the following module: [widget-utils](#)

## POST Getting regular discounts

<https://apiv2.gbooking.ru/rpc>

Obtaining the regular discounts for a specific branch business and a specific service.

- **business.id** — business ID
- *taxonomy.id* — service ID (optional)

Format of discount data schema, both for obtaining and for saving request:

- **active** — whether discount is applied
- **start** — starting time of discount
- **weeklyRepeat** — over 1 — duration of repeated instances of discounts, in weeks
- **unlimWeeklyRepeat** — if 1 — repeated without time limitation (ignoring the **weeklyRepeat** parameter)
- **week** — weekly schedule for discount. Has the following format:

### Plain Text

```
"week": {  
  "sun": [discountTimeFrameSchema],  
  "mon": [discountTimeFrameSchema],  
  "tue": [discountTimeFrameSchema],  
  "wed": [discountTimeFrameSchema],  
  "thu": [discountTimeFrameSchema],  
  "fri": [discountTimeFrameSchema],  
  "sat": [discountTimeFrameSchema]  
},
```

See description of **discountTimeFrameSchema** above.

## HEADERS

**Content-Type** `application/json`

**Body** raw



```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.get_business_taxonomy_discount",
  "params": {
    "business": {
      "id": "4000000006304"
    },
    "taxonomy": {
      "id": "{{TaxonomyID}}"
    }
  }
}
```

## POST Saving regular discounts

<https://apiv2.gbooking.ru/rpc>

Saving the regular discounts for a specific branch business and a specific service.

- **business.id** — business ID
- *taxonomy.id* — service ID (optional)

### HEADERS

**Content-Type** application/json

**Body** raw

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.set_business_taxonomy_discount",
  "params": {
```

```

params: {
  "business": {
    "id": "40000000006304"
  },
  "taxonomy": {
    "id": "{{TaxonomyID}}"
  },
  "discount": {
    "id": "52f3a6500eb2b3f21c000058",
    "active": true,
    "start": "2018-12-14T09:00:00.764Z",
    "weeklyRepeat": 3,
    "unlimWeeklyRepeat": false,
    "week": {
      "sun": [],
      "mon": [],
      "tue": [{
        "start": 540,
        "end": 600,
        "discount": 100,
        "provider": "LOCAL"
      }],
      "wed": [],
      "thu": [],
      "fri": [],
      "sat": []
    }
  }
}
}
}

```

## POST Getting day-specific discounts

<https://apiv2.gbooking.ru/rpc>

Obtaining the discounts for a specific branch business, a specific service, and also for a specific date.

- **business.id** — business ID
- *taxonomy.id* — service ID (optional)
- **start** — starting date of discounted slots
- **end** — ending date of discounted slots

The response will include a **slots** array. Its elements have the following fields:

- **id** — ID of day-specific discount
- **active** — whether the discount is activated
- **date** — day to which the discount applies
- **businessID** — branch business ID
- **taxonomyID** — service ID
- **slots** — discount slots array (see **discountTimeFrameSchema** data schema above)

## HEADERS

---

**Content-Type** application/json

**Body** raw

---

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.get_business_taxonomy_discount_exception",
  "params": {
    "business": {
      "id": "4000000006304"
    },
    "taxonomy": {
      "id": "{{TaxonomyID}}"
    },
    "start": "2018-12-14T09:00:00.764Z",
    "end": "2018-12-20T12:00:00.764Z"
  }
}
```

---

## POST Saving a single day-specific discount

<https://apiv2.gbooking.ru/rpc>

Saving a single day-specific discount.

- **business.id** — business ID
- *taxonomy.id* — service ID
- **discount** — discount data

If no slot ID is specified during saving, this slot will be added to the existing ones. If a slot ID is specified, the slot will be updated.

## HEADERS

---

**Content-Type** application/json

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.set_business_taxonomy_discount_exception",
  "params": {
    "business": {
      "id": "40000000006304"
    },
    "taxonomy": {
      "id": "{{TaxonomyID}}"
    },
    "discount": {
      "id": "52f3a6500eb2b3f21c000058",
      "active": true,
      "date": "2018-12-14T09:00:00.764Z",
      "slots": [{
        "start": 540,
        "end": 600,
        "discount": 100,
        "provider": "LOCAL"
      }]
    }
  }
}
```

## POST Saving day-specific discounts

https://apiv2.gbooking.ru/rpc

Saving day-specific discounts

- **business.id** — business ID
- *taxonomy.id* — service ID
- **discounts** — discount data

The request is similar to the single discount saving request, except that an array of slots is transmitted in the **discounts** field.

### HEADERS

**Content-Type** application/json

**Body** raw

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "cred": {
    "token": "8e7d61be4f200e39ea29b1231006a248de108d9a",
    "user": "5b1035dcaff15607133b523f"
  },
  "method": "business.set_business_taxonomy_discount_exceptions",
  "params": {
    "business": {
      "id": "4000000006304"
    },
    "taxonomy": {
      "id": "{{TaxonomyID}}"
    },
    "discounts": [
      {
        "id": "52f3a6500eb2b3f21c000058",
        "active": true,
        "date": "2018-12-14T09:00:00.764Z",
        "slots": [
          {
            "start": 540,
            "end": 600,
            "discount": 100,
            "provider": "LOCAL"
          }
        ]
      }
    ]
  }
}
```

}]

}]

}

}