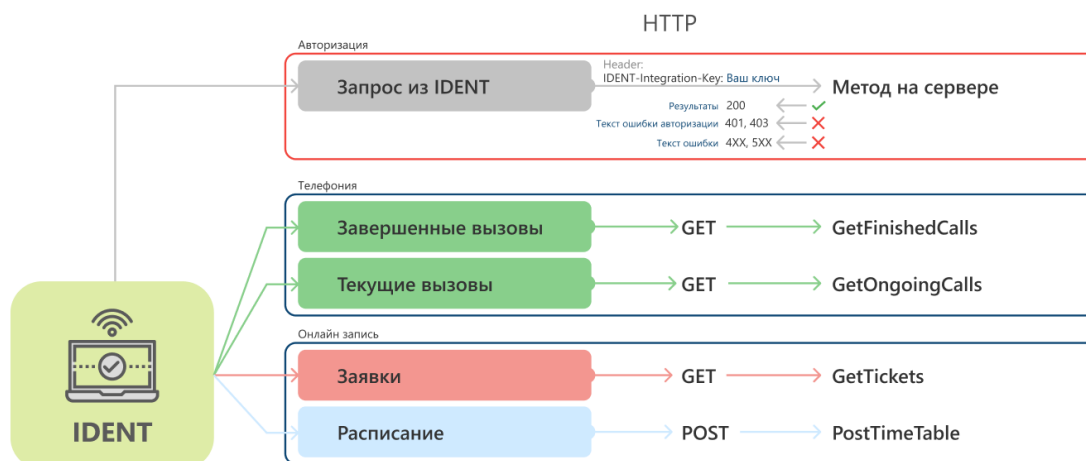


Вариант с HTTP

Последнее изменение 29/06/2020 4:35 pm MSK



Общие сведения

Загрузка данных производится GET-запросами, посланными IDENT. В случае успешного выполнения запроса на стороне сервиса в качестве ответа должно вернуться HTTP-сообщение с кодом статуса 200 OK, а в теле ответа возвращаются данные запроса, сериализованные в JSON или XML.

Выгрузка данных производится POST-запросами, посланными IDENT. Тело запросов содержит выгружаемые данные, сериализованные в формате JSON или XML. В случае успешного выполнения запроса на стороне сервиса в качестве ответа должно вернуться HTTP-сообщение с кодом статуса 200 OK, тело ответа POST-методов в IDENT игнорируется.

Авторизация

В каждый запрос передается header «IDENT-Integration-Key», значение из настройки HTTP-сервиса в IDENT. Сервис должен обрабатывать значение этого параметра как ключ доступа.

В случае отказа в доступе сервис должен вернуть HTTP-сообщение с кодом статуса 403 Forbidden или 401 Unauthorized. В теле ответа достаточно вернуть в понятном для пользователя виде текст сообщения (строка без форматирования) об отказе в доступе. Этот текст будет записан в лог обмена данными с сервисом.

Обработка ошибок

Кроме ошибок авторизации с кодами статуса 401, 403 все остальные ошибки должны передаваться с кодами статуса 4XX, 5XX, а не с кодом 2XX. Только при корректных кодах это интерпретируется как неизвестная ошибка, и работа задачи прекращается. При ошибке с кодом 2XX корректное поведение программы не гарантируется.

Сама ошибка должна передаваться в теле ответа текстом, а не в виде html-страницы.

Сериализация и кодировка

IDENT поддерживает сериализацию JSON и XML, по умолчанию используется кодировка utf-8.

Если кодировка передаваемого контента отличается от utf-8, то в headers ответа сервера необходимо указать заголовок с ключом Content-Type, в котором указана кодировка передаваемого контента в параметре charset.

Например:

- {application/json; charset=utf-8} — для сериализации json кодировка utf-8
- {application/xml; charset=unicode} — для сериализации xml, кодировка unicode

XML:

В случае сериализации XML следует использовать спецификацию [W3](#). Поля объектов передаются в виде атрибутов — с соответствующими названиями.

JSON:

В случае сериализации JSON следует использовать спецификацию [JSON](#). Ответ является массивом объектов.

Сжатие

Для всех GET-запросов IDENT отправляет заголовок (header):

```
accept-encoding: gzip, deflate
```

Но будет корректно обрабатывать как сжатые, так и несжатые результаты. Определение наличия сжатия результатов происходит с помощью response header: content-encoding.

Для POST-запросов необходимость включения сжатия gzip выбирается пользователем при настройке интеграции в IDENT.

Формат даты и времени

Обратите внимание: часовой пояс клиники определяется по компьютеру с серверной частью IDENT, корректируется (при необходимости) при установке сотрудниками технической поддержки и не должен изменяться сотрудниками клиники! Часовой пояс на клиентских компьютерах значения не имеет.

В параметрах GET-запросов:

Даты в параметрах запроса передаются в виде encoded-строки в формате ISO 8601 YYYY-MM-DDTHH:mm:ss±hh:mm,

где:

- YYYY — год;

- MM — месяц;
- DD — день;
- HH — час в формате 24;
- mm — минуты;
- ss — секунды;
- ±hh:mm — сдвиг в часах и минутах от времени в формате UTC (универсального времени). *Будет передан как сдвиг до часового пояса клиники.*

Например: дата 05.12.2017 14:45:37 MCK будет преобразована в формат 2017-12-05T14:45:37+03:00, передана как 2017-12-05T14%3a45%3a37%2b03%3a00.

В объектах результатов:

В объектах результатов запроса поля даты и времени должны быть сериализованы в стандарте ISO 8601 в одном из форматов:

- YYYY-MM-DDTHH:mm:ss
- YYYY-MM-DDTHH:mm:ss±hh:mm

Например: дата 05.12.2017 14:45:37 MCK может быть передана в виде:

- 2017-12-05T14:45:37 — IDENT будет считать, что время уже в часовом поясе клиники, и использовать его, как есть;
- 2017-12-05T14:45:37+03:00 — IDENT автоматически скорректирует значение в часовой пояс клиники.

Итерационная загрузка для GET запросов

При большом объеме данных загрузка может осуществляться частями. В настройках IDENT для сервиса может быть указано максимальное количество записей за один запрос. Если такая настройка указана, то, чтобы получить весь набор данных, запрос будет выполняться несколько раз с передачей целочисленных параметров limit и offset:

- limit — количество записей, которое должен вернуть запрос;
- offset — смещение, индекс элемента с которого IDENT будет ожидать следующую порцию строк.

Просмотр данных завершается, если в результате приходит меньше строк, чем указано в limit. При обращении по несуществующему индексу сервер должен возвращать пустую коллекцию, а не ошибку (иначе при количестве значений, кратном лимиту, мы не сможем понять, где конец данных).

Методы, форматы, примеры

Для загрузки вызовов в сервисе должны быть реализованы два GET-метода:

- GET-метод `GetFinishedCalls` — для загрузки только завершившихся вызовов;
- GET-метод `GetOngoingCalls` — для загрузки только текущих вызовов.

Данные, возвращаемые обоими методами, должны соответствовать описанию в [таблице «Вызовы»](#).

Для загрузки заявок в сервисе должен быть реализован GET-метод `GetTickets`. Возвращаемые объекты должны соответствовать описанию в [таблице «Заявки»](#).

Для выгрузки расписания должен быть реализован POST-метод `PostTimeTable`, в теле которого передается сериализованный объект, содержащий коллекции `Branches`, `Doctors`, `Intervals`, объекты коллекций должны содержать поля из [таблиц «Филиалы», «Врачи», «Слоты расписания»](#) соответственно.

Загрузка завершенных вызовов

Запрос:

Метод: `GetFinishedCalls`

IDENT запрашивает завершенные вызовы за определенный период, поэтому на сервере обязательно должна быть реализована фильтрация данных по времени. В результаты должны попадать объекты, соответствующие крайним значениям фильтра (`dateTimeFrom <= ДатаВремяВызова <= dateTimeTo`).

Параметры:

- `dateTimeFrom` — дата начала периода загрузки
- `dateTimeTo` — дата окончания периода загрузки
- `limit`, `offset` — необязательные параметры для итерационной загрузки

Пример запроса:

```
http[s]://[service_url]/GetFinishedCalls?dateTimeFrom=2017-01-01T00%3a00%3a00%2b03%3a00&dateTimeTo=2200-01-01T23%3a59%3a59%2b03%3a00&limit=500&offset=0
```

Ответ:

Ответом является коллекция объектов с полями из [таблицы «Вызовы»](#), сериализованных в JSON или XML. Необязательные поля могут отсутствовать, в случае сериализации XML значения полей `null` следует передавать в виде пустой строки, например, `RecordUrl=""`.

[Пример ответа JSON](#)

```
[
  {
    "DateAndTime": "2017-01-25T12:30:54+03:00",
    "Direction": "in",
    "PhoneFrom": "+79116844567",
    "PhoneTo": "+78126497035",

    "WaitInSeconds": 30,
    "TalkInSeconds": null,

    "RecordUrl": null
  },
  {
    "DateAndTime": "2017-02-25T12:32:54+03:00",
    "Direction": "in",
    "PhoneFrom": "+79126844567",
    "PhoneTo": "+78126497035",

    "WaitInSeconds": 10,
    "TalkInSeconds": 50,

    "RecordUrl": "https://myserver/asdfgh897383"
  },
  {
    "DateAndTime": "2017-03-25T12:40:54+03:00",
    "Direction": "in",
    "PhoneFrom": "+79136844567",
    "PhoneTo": "+78126497035",

    "WaitInSeconds": 30,
    "TalkInSeconds": 50,

    "RecordUrl": "https://myserver/asdfgh546456"
  }
]
```

[Пример ответа XML](#)

Загрузка текущих вызовов

Запрос:

Метод: GetOngoingCalls

Параметры:

- limit, offset — необязательные параметры для итерационной загрузки

Пример запроса:

http[s]://[service_url]/GetOngoingCalls?limit=500&offset=0

Ответ:

Формат такой же, как и у метода загрузки завершенных вызовов.

Загрузка заявок

Запрос:

Метод: GetTickets

IDENT запрашивает заявки за определенный период, поэтому на сервере обязательно должна быть реализована фильтрация данных по времени. В результаты должны попадать объекты, соответствующие крайним значениям фильтра (`dateTimeFrom <= ДатаВремяЗаявки <= dateTimeTo`).

Параметры:

- `dateTimeFrom` — дата начала периода загрузки
- `dateTimeTo` — дата окончания периода загрузки
- `limit, offset` — необязательные параметры для итерационной загрузки

Пример запроса:

```
http[s]://[service_url]/GetTickets?dateTimeFrom=2017-01-01T00%3a00%3a00%2b03%3a00&dateTimeTo=2200-01-01T23%3a59%3a59%2b03%3a00&limit=500&offset=0
```

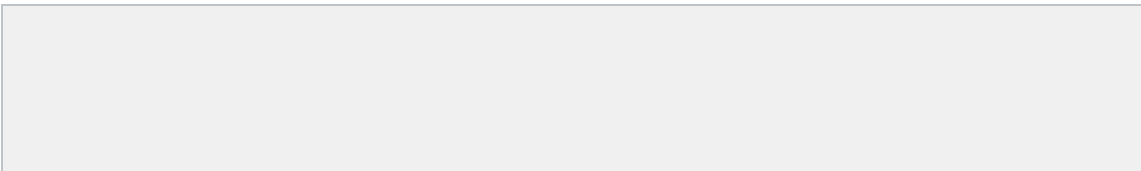
Ответ:

Ответом является коллекция объектов с полями из таблицы «[Заявки](#)», сериализованных в JSON или XML.

[Пример ответа JSON](#)

```
[
  {
    "Id": "1",
    "DateAndTime": "2017-01-25T12:30:54+03:00",
    "ClientPhone": "+79852345678",
    "ClientEmail": null,
    "FormName": null,
    "ClientFullName": "Иванов Иван"
  },
  {
    "Id": "2",
    "DateAndTime": "2017-02-24T08:10:54+03:00",
    "ClientPhone": "7(812)7474674",
    "ClientEmail": null,
    "FormName": null,
    "ClientFullName": "Петров Петр"
  },
  {
    "Id": "3",
    "DateAndTime": "2017-03-23T09:11:54+03:00",
    "ClientPhone": "+77774666776",
    "ClientEmail": null,
    "FormName": null,
    "ClientFullName": "Семен Семенов"
  },
  {
    "Id": "4",
    "DateAndTime": "2017-04-22T10:12:54+03:00",
    "ClientPhone": "8495-657-77-75",
    "ClientEmail": null,
    "FormName": null,
    "ClientFullName": "Владимир Смирнов"
  }
]
```

[Пример ответа XML](#)



Выгрузка расписания

Метод: PostTimeTable

Пример запроса:

`http[s]://[service_url]/PostTimeTable`

► Обратите внимание!

Тело запроса:

[JSON](#)

```
{
  "Doctors": [
```

```
{
  "Id": 2129,
  "Name": "Иванов Виталий Сергеевич"
},
{
  "Id": 13417,
  "Name": "Петров Александр Иванович"
},
{
  "Id": 4020,
  "Name": "Смирнов Константин Алексеевич"
},
{
  "Id": 8932,
  "Name": "Кузнецова Екатерина Андреевна"
},
{
  "Id": 12624,
  "Name": "Семенова Ирина Ивановна"
}
],
"Branches": [
  {
    "Id": 1,
    "Name": "Филиал в г. Санкт-Петербург"
  }
],
"Intervals": [
  {
    "DoctorId": 2129,
    "BranchId": 1,
    "StartDateTime": "2019-03-27T16:45:00+03:00",
    "LengthInMinutes": 255,
    "IsBusy": false
  },
  {
    "DoctorId": 4020,
    "BranchId": 1,
    "StartDateTime": "2019-03-27T16:45:00+03:00",
    "LengthInMinutes": 255,
    "IsBusy": false
  },
  {
    "DoctorId": 8932,
    "BranchId": 1,
    "StartDateTime": "2019-03-27T16:45:00+03:00",
    "LengthInMinutes": 255,
    "IsBusy": false
  },
  {
    "DoctorId": 2129,
    "BranchId": 1,
    "StartDateTime": "2019-03-28T15:00:00+03:00",
    "LengthInMinutes": 360,
    "IsBusy": false
  },
  {
    "DoctorId": 2129,
    "BranchId": 1,
    "StartDateTime": "2019-03-29T10:00:00+03:00",
    "LengthInMinutes": 270,
    "IsBusy": false
  },
  {
```



```
    "DoctorId": 4020,  
    "BranchId": 1,  
    "StartDateTime": "2019-03-29T09:00:00+03:00",  
    "LengthInMinutes": 720,  
    "IsBusy": false  
  },  
{  
    "DoctorId": 8932,  
    "BranchId": 1,  
    "StartDateTime": "2019-03-28T09:00:00+03:00",  
    "LengthInMinutes": 330,  
    "IsBusy": false  
  },  
{  
    "DoctorId": 12624,  
    "BranchId": 1,  
    "StartDateTime": "2019-03-28T09:00:00+03:00",  
    "LengthInMinutes": 720,  
    "IsBusy": false  
  },  
{  
    "DoctorId": 13417,  
    "BranchId": 1,  
    "StartDateTime": "2019-03-28T15:00:00+03:00",  
    "LengthInMinutes": 360,  
    "IsBusy": false  
  }  
]  
}
```

[XML](#)

