

Документация
API проекта "Запись на приём"
(appointment-web)

Содержание

RESTful API	3
Авторизация	3
Авторизация запросов	3
Алгоритм UsernameToken	4
Запросы	5
Общая информация по запросам	5
Информация по представлению дат и времени	6
Получение информации о основном сервере GET /api/srv/info	7
Выбор района GET /api/srv/district	8
Выбор ЛПУ GET /api/srv/district/{idDistrict}/lpu	9
Получение информации о ЛПУ сервисе GET /api/srv/lpu/{lpuId}/info	10
Добавление пациента POST /api/srv/lpu/{lpuId}/patient	11
Идентификация пациента GET /api/srv/lpu/{lpuId}/patient/check	13
Передача (обновление) номера телефона пациента PUT /api/srv/lpu/{lpuId}/patient/{patientId}/updatephone	14
Выбор специальности GET /api/srv/lpu/{lpuId}/speciality	15
Выбор врача GET /api/srv/lpu/{lpuId}/speciality/{specId}/doctor	16
Выбор даты GET /api/srv/lpu/{lpuId}/doctor/{doctorId}/availabledates	17
Выбор времени приема GET /api/srv/lpu/{lpuId}/doctor/{doctorId}/availableappointments	18
Подтверждение записи POST /api/srv/lpu/{lpuId}/appointment	19
Просмотр предстоящих записей пациента GET /api/srv/lpu/{lpuId}/patient/{patientId}/history	20
Отмена записи на прием пациента DELETE /api/srv/lpu/{lpuId}/patient/{patientId}/appointment/{appointmentId}	21
Валидация кода направления GET /api/srv/lpu/{lpuId}/referral/{referralNumber}/inspect	22
Просмотр расписаний GET /api/srv/lpu/{lpuId}/doctor/{doctorId}/workingtime	23

RESTful API

Авторизация

Авторизация запросов

Используется WSSE UsernameToken авторизация.

Для передачи авторизационных данных используется заголовок запроса (Header)

Пример заголовка авторизованного запроса

Пример для пользователя test с паролем 7f5144f9-de8c-458e-bb1f-99e5b17be3d3

Заголовок (Header)

```
Content-Type: application/json
Authorization: WSSE profile="UsernameToken"
X-WSSE: UsernameToken Username="test", PasswordDigest="6TuDuSwBDtrqZ44andTm0Ws6oGY=",
Nonce="0gBN2nWRYu12xbrRtamX", Created="2016-12-08T15:28:20.000+03:00"
```

где

- Username - имя пользователя (текст);
- PasswordDigest - дайджест пароля (Base64);
- Nonce - одноразовый случайный код (Base64);
- Created - дата и время создания запроса (текст);

Ответ при успешной авторизации

HTTP status code - 200

... объект или объекты в формате JSON

Ответ при неудачной авторизации

HTTP status code - 403

```
{
  "errorCode": "AuthenticationException",
  "errorMsg": "Full authentication is required to access this resource"
}
```

Алгоритм UsernameToken

```
PasswordDigest = Base64(SHA1(BinaryNonce + Created + Password))
```

где

- BinaryNonce - (от англ. nonce — «специально для данного случая») одноразовый код, выбранный случайным или псевдослучайным образом (для каждого запроса уникальный) в бинарном виде (набор байт);
- Created - дата создания запроса, примеры поддерживаемых форматов (2016-12-08T15:28:20.000+03:00 - текущая дата и время с миллисекундами и часовым поясом, 2016-12-08T15:28:20+03:00 - текущая дата и время с часовым поясом, 2016-12-08T12:56:52.000Z дата и время UTC с миллисекундами, 2016-12-08T12:56:52Z дата и время UTC)
- Password - пароль пользователя
- PasswordDigest - дайджест пароля - результат работы алгоритма.

Запросы

Общая информация по запросам

Кодировка параметров запросов - **UTF-8**

Заголовок (Header) запроса обязательно содержит **Content-Type: application/json**

```
Content-Type: application/json
...
авторизационные данные
...
```

Ответ в случае успеха

HTTP status code - 200

```
... объект или объекты в формате JSON
возможно пустое тело ответа в случае отсутствия данных
```

Ответ в случае ошибки

В случае ошибки авторизации - HTTP status code - 403

```
{
  "errorCode": "AuthenticationException",
  "errorMsg": "Full authentication is required to access this resource"
}
```

В случае обрабатываемых приложением ошибок - HTTP status code - 400

```
{
  "errorCode": "... код ошибки (пример кодов NotFound, LimitExceeded),
  "errorMsg": "... сообщение
}
```

В случае не обрабатываемых приложением ошибок - HTTP status code - 500

```
{
  "errorCode": Exception,
  "errorMsg": "... стек ошибки приложения
}
```

Информация по представлению дат и времени

Все входные и выходные данные в которых присутствует время и в запросе используется **lpuId** (идентификатор лечебного учреждения) используют местное время и местный часовой пояс (соответствующего лечебного учреждения).

Например: 2016-12-09T12:01:00 - это 12:01 местного времени лечебного учреждения.

Узнать текущее местное время и соответствующий текущий часовой пояс лечебного учреждения можно так

Получение информации о ЛПУ сервисе GET /api/srv/lpu/{lpuId}/info

Получение информации о основном сервере GET /api/srv/info

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/info
```

Ответ

```
{  
  "version": "0.0.1",  
  "date": "2016-12-09T10:41:45+03:00"  
}
```

Описание ответа:

- version - версия программного обеспечения сервера;
- date - текущая дата и время с часовым поясом.

Выбор района GET /api/srv/district

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/district
```

Ответ

```
[
  {
    "districtName": "Ломоносовский район",
    "idDistrict": 1,
    "okato": 40283000000
  },
  {
    "districtName": "Московский район",
    "idDistrict": 2,
    "okato": 40284000000
  }
]
```

Описание ответа:

- districtName - наименование района;
- idDistrict - идентификатор района;
- okato - код ОКАТО района.

Выбор ЛПУ

GET /api/srv/district/{idDistrict}/lpu

Обязательные параметры

- idDistrict - идентификатор района.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/district/1/lpu
```

Ответ

```
[
  {
    "description": "Описание больницы №1",
    "idDistrict": 1,
    "idLPU": 2,
    "lpuFullName": "Больница №1",
    "lpuShortName": "Б1",
    "lpuType": 1
  }
]
```

Описание ответа:

- description - описание;
- idDistrict - идентификатор района;
- idLPU - идентификатор ЛПУ;
- lpuFullName - юридическое наименование ЛПУ;
- lpuShortName - аббревиатура (краткое наименование ЛПУ);
- lpuType - тип ЛПУ.

Таблица 1: Типы ЛПУ.

LpuType	TypeName
1	амбулатория
2	стоматология
3	онкология
4	травматология
5	КВД
6	женская консультация
7	ПНД
8	ПТД
9	роддом
10	стационар

Получение информации о ЛПУ сервисе

GET /api/srv/lpu/{lpuId}/info

Обязательные параметры

- lpuId - идентификатор ЛПУ.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/info
```

Ответ

```
{  
  "version": "0.0.1",  
  "date": "2016-12-09T10:41:45+03:00"  
}
```

Описание ответа:

- version - версия программного обеспечения ЛПУ сервиса;
- date - текущая дата и время с часовым поясом ЛПУ сервиса.

Добавление пациента

POST /api/srv/lpu/{lpuId}/patient

Данный метод используется для добавления пациента (в случае, если пациент отсутствует в базе данных) при осуществлении записи на прием к врачу.

Используемые DB процедуры

p_web2.add_new_patient

Обязательные параметры

- lpuId - идентификатор ЛПУ.

Пример POST запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/patient
```

Тело запроса

```
{
  "birthday": "1977-09-22",
  "cellPhone": null,
  "homePhone": null,
  "documentNumber": null,
  "documentSeries": null,
  "surname": "Семенов",
  "name": "Егор",
  "secondname": "Семенович",
  "polisNumber": null,
  "polisSeries": null,
  "snils": null
}
```

Описание запроса (все параметры строки):

- birthday - дата рождения (**совместный параметр**) - формат уууу-ММ-дд;
- cellPhone - номер мобильного телефона (без плюса цифры) (**совместный параметр**);
- homePhone - номер домашнего телефона (без плюса цифры) (**совместный параметр**);
- documentNumber - номер документа;
- documentSeries - серия документа;
- surname - фамилия (**обязательный параметр**);
- name - имя (**обязательный параметр**);
- secondname - отчество;
- polisNumber - номер полиса ОМС;
- polisSeries - серия полиса ОМС;
- snils - СНИЛС.

Минимальные наборы параметров:

birthday + surname + name;
cellPhone + surname + name;
homePhone + surname + name.

Ответ в случае успешного выполнения

HTTP status code - 200

```
{
  "id": 384012750,
  "errorCode": 0
}
```

Ответ в случае ошибки

HTTP status code - 200

```
{  
  "id":null,  
  "errorCode":21  
}
```

Описание ответа:

- id - идентификатор созданного пациента (если null то ошибка)
- errorCode - код ошибки (0 - успешное выполнение, 20 - не создано, 21 - прочие ошибки).

Идентификация пациента GET /api/srv/lpu/{lpuId}/patient/check

Используемые ДВ процедуры

p_web2.check_patient

Обязательные параметры

- lpuId - идентификатор ЛПУ.

GET параметры (после знака ?)

- birthday - дата рождения - формат dd-ММ-уууу;
- cellphone - номер мобильного телефона;
- document_n - номер документа;
- document_s - серия документа;
- homerphone - номер домашнего телефона;
- idpat - идентификатор пациента;
- name - имя;
- surname - фамилия;
- secondname - отчество;
- polis_n - номер полиса ОМС;
- polis_s - серия полиса ОМС.

Необходимо использовать хотя бы один GET параметр.

Пример GET запроса

{lpuId} -> 1

birthday -> 24-01-1981; name -> александр; surname -> иванов.

```
http://localhost:8080/appointment-web/api/srv/lpu/1/patient/check?
surname=%D0%B8%D0%B2%D0%B0%D0%BD%D0%BE%D0%B2
&birthday=24-01-1981&name=%D0%B0%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80
```

Ответ в случае успешного выполнения

```
{
  "id":219834
}
```

Описание ответа:

- id - идентификатор пациента.

Ответ в случае если найдено несколько пациентов (больше одного)

```
{
  "errorMsg":null,
  "errorCode":"LimitExceeded"
}
```

Ответ если никто не найден

```
{
  "errorMsg":null,
  "errorCode":"NotFound"
}
```

Если найдено несколько пациентов то нужно уточнить запрос добавив необходимые дополнительные параметры.

Передача (обновление) номера телефона пациента PUT /api/srv/lpu/{lpuId}/patient/{patientId}/updatephone

Данный метод используется изменения номера телефона пациента.

Используемые DB процедуры
p_web2.update_patient_phone

Обязательные параметры

- lpuId - идентификатор ЛПУ.
- patientId - идентификатор пациента.

Пример PUT запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/patient/384012842/updatephone
```

Тело запроса

```
{  
  "patientId":384012842,  
  "cellPhone":"5783347",  
  "homePhone":"5783346"  
}
```

Описание запроса (все параметры строки):

- patientId - идентификатор пациента;
- cellPhone - номер мобильного телефона (без плюса цифры);
- homePhone - номер домашнего телефона (без плюса цифры).

Минимальные наборы параметров:

patientId + cellPhone;
patientId + homePhone.

Ответ в случае успеха

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

Ответ в случае неудачи

```
{  
  "result":false  
}
```

Выбор специальности GET /api/srv/lpu/{lpuId}/speciality

Используемые ДВ процедуры

p_web2.get_specs

Обязательные параметры

- lpuId - идентификатор ЛПУ.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/speciality
```

Ответ

```
[
  {
    "fedSpecId": "118",
    "specId": 26169,
    "specName": "Кардиолог",
    "countFree": 0,
    "firstDate": null,
    "lastDate": null
  },
  {
    "fedSpecId": "11",
    "specId": 85169,
    "specName": "Детский хирург",
    "countFree": 0,
    "firstDate": null,
    "lastDate": null
  },
  {
    "fedSpecId": "27",
    "specId": 85159,
    "specName": "Терапевт участковый",
    "countFree": 0,
    "firstDate": null,
    "lastDate": null
  },
  {
    "fedSpecId": "8",
    "specId": 26159,
    "specName": "Акушер-гинеколог",
    "countFree": 161,
    "firstDate": "2016-12-26T08:20:00",
    "lastDate": "2016-12-31T16:40:00"
  }
]
```

Описание ответа:

- fedSpecId - идентификатор врачебной специальности в федеральном справочнике специальностей (OID 1.2.643.5.1.13.2.1.1.181);
- specId - идентификатор врачебной специальности из соответствующего справочника целевой МИС;
- specName - наименование врачебной специальности из соответствующего справочника целевой МИС;
- countFree - количество свободных талонов по врачебной специальности;
- firstDate - дата приема по ближайшему свободному талону врачебной специальности - формат уууу-ММ-дд'Т'НН:мм:ss (пример - 2016-12-09T00:00:00);
- lastDate - дата приема по последнему свободному талону врачебной специальности - уууу-ММ-дд'Т'НН:мм:ss (пример - 2016-12-09T00:00:00).

Выбор врача

GET /api/srv/lpu/{lpuId}/speciality/{specId}/doctor

Используемые DB процедуры

p_web2.get_docs

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- specId - идентификатор специальности из соответствующего справочника целевой МИС;

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/speciality/26159/doctor
```

Ответ

```
[
  {
    "id":12617000,
    "name":"Абов Виталий Григорьевич",
    "countFree":161,
    "firstDate":"2016-12-26T08:20:00",
    "lastDate":"2016-12-31T16:40:00"
  },
  {
    "id":12730000,
    "name":"Арефьева Ольга Алексеевна",
    "countFree":0,
    "firstDate":null,
    "lastDate":null
  },
  ...
]
```

Описание ответа:

- id - идентификатор врача;
- name - ФИО врача (полностью);
- countFree - количество свободных талонов по врачебной специальности;
- firstDate - дата приема по ближайшему свободному талону врачебной специальности - формат уууу-ММ-дд'T'HH:mm:ss (пример - 2016-12-09T00:00:00);
- lastDate - дата приема по последнему свободному талону врачебной специальности - уууу-ММ-дд'T'HH:mm:ss (пример - 2016-12-09T00:00:00).

Выбор даты

GET /api/srv/lpu/{lpuId}/doctor/{doctorId}/availabledates

Используемые DB процедуры

p_web2.get_docs

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- doctorId - идентификатор врача на отделении.

GET параметры (после знака ?)

- visit_start - дата начала диапазона поиска свободных дней для записи (**обязательный параметр**) - формат dd-ММ-уууу;
- visit_end - дата окончания диапазона поиска свободных дней для записи (**обязательный параметр**) - формат dd-ММ-уууу.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/doctor/12617000/availabledates?visit_start=10-12-2016&visit_end=01-02-2017
```

Ответ

```
[
  {
    "value": "2016-12-26"
  },
  {
    "value": "2016-12-27"
  },
  {
    "value": "2016-12-28"
  },
  {
    "value": "2016-12-29"
  },
  {
    "value": "2016-12-30"
  },
  {
    "value": "2016-12-31"
  }
]
```

Описание ответа:

- value - Дата, на которую есть свободные талоны для записи - формат уууу-ММ-dd.

Выбор времени приема

GET /api/srv/lpu/{lpuId}/doctor/{doctorId}/availableappointments

Используемые ДВ процедуры

p_web2.get_available_appointments

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- doctorId - идентификатор врача на отделении.

GET параметры (после знака ?)

- visit_start - дата начала диапазона поиска свободных талонов (**обязательный параметр**) - формат dd-MM-yyuu;
- visit_end - дата окончания диапазона поиска свободных талонов (**обязательный параметр**) - формат dd-MM-yyuu.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/doctor/12617000/availableappointments?visit_start=26-12-2016&visit_end=27-12-2016
```

Ответ

```
[
  {
    "id":1263480,
    "visitStart":"2016-12-26T08:00:00",
    "visitEnd":"2016-12-26T08:20:00"
  },
  {
    "id":1263481,
    "visitStart":"2016-12-26T08:20:00",
    "visitEnd":"2016-12-26T08:40:00"
  },
  {
    "id":1263482,
    "visitStart":"2016-12-26T08:40:00",
    "visitEnd":"2016-12-26T09:00:00"
  },
  ...
]
```

Описание ответа:

- id - идентификатор талона для записи;
- visitStart - дата и время начала приема - формат yyuu-MM-dd'T'HH:mm:ss;
- visitEnd - дата и время окончания приема - формат yyuu-MM-dd'T'HH:mm:ss.

Подтверждение записи

POST /api/srv/lpu/{lpuId}/appointment

Данный метод используется для записи пациента на прием в выбранный пациентом временной интервал рабочего времени врача.

Используемые DB процедуры

p_web2.set_appointment

Обязательные параметры

- lpuId - идентификатор ЛПУ.

Пример POST запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/appointment
```

Тело запроса

```
{
  "patientId":219834,
  "appointmentId":1263480
}
```

Описание запроса:

- patientId - идентификатор пациента;
- appointmentId - идентификатор талона для записи.

Ответ в случае успешного выполнения

HTTP status code - 200

```
{
  "result":1,
  "errorCode":null,
  "errorText":null
}
```

Ответ в случае ошибки

HTTP status code - 200

```
{
  "result":0,
  "errorCode":39,
  "errorText":"Номерок более не доступен!\nНеобходимо выбрать другое время для посещения!"
}
```

Описание ответа:

- result - результат выполнения (1 - успешно, 0 - ошибочно);
- errorCode - код ошибки (число) (38 - Не задан идентификатор талона на запись, 37 - Не задан идентификатор пациента, 38 - Указан недопустимый идентификатор талона на запись, 39 - Номерок более не доступен, 40 - Имеется запись на данное время к другому специалисту, 35 - Имеется запись на выбранную дату к данному специалисту, 34 - Невозможно записаться к специалисту более 2 раз за 10 дней);
- errorText - ошибка, текстовое сообщение.

Просмотр предстоящих записей пациента

GET /api/srv/lpu/{lpuId}/patient/{patientId}/history

Данный метод используется для получения списка записей пациента на прием и отказов от записи на прием в указанном в запросе ЛПУ, даты по которым еще не наступили.

Используемые DB процедуры

p_web2.get_patient_history

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- patientId - идентификатор пациента.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/patient/219834/history
```

Ответ

```
{
  "visit": [
    {
      "id": 1263480,
      "type": 0,
      "beginDate": "2016-12-26T08:00:00",
      "endDate": "2016-12-26T08:20:00",
      "specId": 26159,
      "specName": "Акушер-гинеколог",
      "depId": 201,
      "depName": "Эндоскопическое отделение",
      "doctorId": 12617000,
      "doctorName": "Абов Виталий Григорьевич",
      "room": null,
      "address": "Санкт-Петербург, \nУчебный пер., д. 5, к. 3"
    }
  ],
  "refusal": [
  ]
}
```

Описание ответа:

- visit - информация об актуальных предстоящих записях пациента на прием;
- refusal - информация о записях на прием к врачу, по которым произведен отказ;
- id - идентификатор талона на запись;
- type - тип записи (0 - актуальная запись, 1 - запись отменена, 2 - запись удалена);
- beginDate - дата и время начала приема - формат уууу-ММ-dd'T'HH:mm:ss;
- endDate - дата и время окончания приема - формат уууу-ММ-dd'T'HH:mm:ss;
- specId - идентификатор специальности;
- specName - название специальности;
- depId - идентификатор отделения;
- depName - название отделения;
- doctorId - идентификатор доктора на отделении;
- doctorName - ФИО доктора;
- room - номер кабинета;
- address - адрес ЛПУ.

Примечание:

Для удаленных записей (type - 2) будет заполнены поля id и type остальные поля будут иметь значение null.

Отмена записи на прием пациента

DELETE /api/srv/lpu/{lpuId}/patient/{patientId}/appointment/{appointmentId}

Данный метод используется для отмены талона пациента.

Используемые DB процедуры

p_web2.release_appointment

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- patientId - идентификатор пациента;
- appointmentId - идентификатор талона для записи.

Пример DELETE запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/patient/219834/appointment/1263480
```

Ответ

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

Валидация кода направления

GET /api/srv/lpu/{lpuId}/refferal/{referralNumber}/inspect

Проверка кода направления.

Используемые ДВ процедуры

p_web2.inspect_doctor_referral

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- referralNumber - номер направления.

Пример GET запроса

referralNumber -> PM124HM

```
http://localhost:8080/appointment-web/api/srv/lpu/1/refferal/%D0%A0%D0%9C124%D0%9D%D0%9C/inspect
```

Ответ

```
{
  "doctorId":12617000,
  "patientId":219834,
  "type":"консультация"
}
```

Описание ответа:

- doctorId- идентификатор врача на отделении;
- patientId - идентификатор пациента;
- type - тип направления (консультация, повторный).

Просмотр расписаний

GET /api/srv/lpu/{lpuId}/doctor/{doctorId}/workingtime

Используемые ДВ процедуры

p_web2.get_working_time

Обязательные параметры

- lpuId - идентификатор ЛПУ;
- doctorId - идентификатор врача на отделении.

GET параметры (после знака ?)

- visit_start - дата начала диапазона для получения информации о расписании (**обязательный параметр**) - формат dd-ММ-уууу;
- visit_end - дата окончания диапазона для получения информации о расписании (**обязательный параметр**) - формат dd-ММ-уууу.

Пример GET запроса

```
http://localhost:8080/appointment-web/api/srv/lpu/1/doctor/12617000/workingtime?
visit_start=26-12-2016&visit_end=27-12-2016
```

Ответ

```
[
  {
    "denyCause":null,
    "visitStart":"2016-12-26T08:00:00",
    "visitEnd":"2016-12-26T17:00:00",
    "recordableDay":true
  },
  {
    "denyCause":null,
    "visitStart":"2016-12-27T08:00:00",
    "visitEnd":"2016-12-27T17:00:00",
    "recordableDay":true
  }
]
```

Описание ответа:

- denyCause - описание причины отсутствия записи (Текст);
- visitStart - дата и время начала приема - формат уууу-ММ-dd'T'HH:mm:ss;
- visitEnd - дата и время окончания приема - формат уууу-ММ-dd'T'HH:mm:ss;
- recordableDay - возможна ли запись на прием.