**Спецификация Infodoctor WebAPI Ver. 1.0**

Содержание

[**Работа с юзерами 2**](#_gjdgxs)

[**Работа с клиниками 2**](#_30j0zll)

[**Работа api городов 3**](#_3znysh7)

[**Работа с чем-то 4**](#_2et92p0)

[**Работа с чем-то 4**](#_tyjcwt)

# Работа с юзерами

Регистрация:

URL: /api/account/register,

METHOD: POST,

HEADER:

Content-Type : application/json; charset=utf-8,

BODY:

JSON объекта

var data = {

Email: ,

Password: ,

UserName: ,

Phone:

};

*Поле "Phone" является необязательным, остальные обязательны!*

Ответ:

HttpResponse

Пример:

|  |  |
| --- | --- |
| Запрос | {  "Email":"test1@test.by",  "Password":"1234Aa",  "UserName":"TestName1",  "Phone":"+123456789"  } |
| Ответ если удачно | Status 200 OK |
| Ответ отсутствует обязательное поле | Status 400 Bad Request  {  "Message": "The request is invalid.",  "ModelState": {  "model.Email": [  "Требуется поле Email."  ]  }  } |
| Ответ если такой Email или UserName уже имеются в БД | Status 400 Bad Request  {  "Message": "The request is invalid.",  "ModelState": {  "": [  "Name test1@test.by is already taken.",  "Email 'test1@test.by' is already taken."  ]  }  } |

Авторизация:

URL: /token,

METHOD: POST,

HEADER:

Content-Type : application/x-www-form-urlencoded,

BODY:

grant\_type: "password",

username,

password

Ответ:

{

"access\_token": "token",

"token\_type": "bearer",

"expires\_in": 1209599,

"userName": "user",

".issued": "Wed, 22 Feb 2017 16:54:22 GMT",

".expires": "Wed, 08 Mar 2017 16:54:22 GMT"

}

# Работа с клиниками

URL: /api/clinic/{id} – общий вид

где {id} – необязательный параметр

Если id указан, то возвращается клиника с этим id, если не указан – то массив всех клиник

Пример:

|  |  |
| --- | --- |
| запрос клиники 3: /api/clinic/3  метод GET | Status 200 OK  {  "Id": 3,  "Name": "Медицинский центр Кравира",  "Email": "kravira@kravira.by",  "ClinicAddress": [  {  "Street": "пр-т. Победителей 75,",  "City": "Минск",  "ClinicPhones": [  {  "Desc": "",  "Phone": "+375 17 211 28 61"  },  {  "Desc": "",  "Phone": "+375 29 611 28 61"  }  ]  },  {  "Street": "ул.Скрипникова 11Б,",  "City": "Минск",  "ClinicPhones": [  {  "Desc": "",  "Phone": "+375 17 314 94 94"  },  {  "Desc": "",  "Phone": "+375 29 664 44 44"  }  ]  },  {  "Street": "ул.Захарова 50Д",  "City": "Минск",  "ClinicPhones": [  {  "Desc": "",  "Phone": "+375 17 253 33 33"  },  {  "Desc": "",  "Phone": "+375 29 103 43 43"  }  ]  }  ],  "ClinicSpecialization": [  "Гинекология",  "Дерматология", *и т.д.*  ]} |
| запрос клиники, которой нет в БД :  /api/clinic/666  метод GET | Status 500 Internal Server Error  {  "Message": "An error has occurred.",  "ExceptionMessage": "Clinic not found",  "ExceptionType": "System.ApplicationException",  "StackTrace": " в Infodoctor.BL.Services.ClinicService.GetClinicById(Int32 id) …  … и еще много букв  } |

Общий вид объекта clinic

"Id"– id клиники в базе данных

"Name" – название клиники

"Email" – адрес эл. Почты

"ClinicSpecialization" – массив специализаций клиники

"ClinicAddress" – массив адресов филиалов клиник. Если клиника без филиалов, то в массиве будет 1 адрес

"City" – название города, в котором находится клиника

"Street" – улица и дом клиники

"ClinicPhones" – массив телефонов, соответствующий каждому филиалу клиники

"Desc": описание номера, например, "Детское отделение",

"Phone": собственно, номер

Пагинация клиник

URL: /api/clinic/page/{perPage}/{numPage}

где perPage и numPage значения типа int

Запрос возвращает объект вида:

Clinics[] – массив объектов типа clinic (см. выше)

Page – номер страницы (совпадает с numPage в запросе)

PageSize – – номер страницы (совпадает с perPage в запросе)

TotalCount – общее количество объектов типа clinic в БД

TotalPages – общее количество страниц при условии PageSize

Пример:

|  |  |
| --- | --- |
| /api/clinic/page/2/2  метод GET | Status 200 OK  {  "Clinics": [],  "TotalCount": 5,  "Page": 2,  "PageSize": 2,  "TotalPages": 3  } |
| /api/clinic/page/10/666  метод GET  Запрос страницы с номером больше, чем страниц может существовать | Status 500 Internal Server Error  {  "Message": "An error has occurred.",  "ExceptionMessage": "Page not found",  "ExceptionType": "System.ApplicationException",  "StackTrace": " в …  … и еще много букв  } |
| /api/clinic/page/-10/0  метод GET  Значения переменных perPage или numPage < 1 | Status 500 Internal Server Error  {  "Message": "An error has occurred.",  "ExceptionMessage": "Page not found",  "ExceptionType": "System.ApplicationException",  "StackTrace": " в …  … и еще много букв  } |
| /api/clinic/page/+-=/xyz  метод GET  Переменные perPage или numPage не типа int | Status 404 Not Found |

# Работа api городов

Получение всего списка Городов:

METHOD: GET,

URL: /api/cities,

HEADER: none,

BODY: none,

Ответ:

Json List

[

{

"Id": 1,

"Name": "Минск"

},

{

"Id": 2,

"Name": "Брест"

},

...

]

Получение всего списка городов имеющих клиники:

METHOD: GET,

URL: /api/cities/clinicused,

HEADER: none,

BODY: none,

Ответ:

Json List

[

{

"Id": 1,

"Name": "Минск"

},

{

"Id": 2,

"Name": "Брест"

},

...

]

Получение города по численому id:

URL: /api/cities,

PARAM: id(int),

METHOD: GET,

HEADER: none,

BODY: none,

Ответ:

Json объект

{

"Id": 1,

"Name": "Минск"

}

Добавление нового города:

METHOD: POST,

URL: /api/cities,

PARAM: none,

HEADER:

Authorization : Bearer token,

Content-Type : application/x-www-form-urlencoded,

BODY: название нового элемента(string),

Ответ: none

Изменение существующего города:

METHOD: PUT,

URL: /api/cities,

PARAM: id(int),

HEADER:

Authorization : Bearer token,

Content-Type : application/x-www-form-urlencoded,

BODY: новое значение для существующего элемента(string),

Ответ: none

Удаление существующего города:

METHOD: DELETE,

URL: /api/cities,

PARAM: id(int),

HEADER:

Authorization : Bearer token,

BODY: none,

Ответ: none

# Работа с чем-то

Тут что-то будет

# Работа с чем-то