Админское Api

1. Изменение системных параметров. (POST обычный, без form-data)

[http://localhost:8080/admin?secretKey=<тут](http://localhost:8080/admin?secretKey=%3cтут) секретный ключ> &deathTimeSize=6&irrelevanceLevelMax=16

&maxNumberMarksPerDay=20

secretKey – секретный ключ для безопасности, если он будет неправильный запрос не пройдёт

deathTimeSize – через сколько часов умрёт метка

irrelevanceLevelMax – максимальный уровень неактуальности метки

maxNumberMarksPerDay - количество меток в сутки

Наличие всех параметров в запросе не обязательно. Можно указать только те, которые изменились.

Если запрос прошёл удачно, то вернётся:

{

"id": 0,

"deathTimeSize": 5,

"irrelevanceLevelMax": 10,

"maxNumberMarksPerDay": 10

}

В случае неудачи NULL.

1. Удаление метки из БД. (POST обычный, без form-data)

[http://localhost:8080/](http://localhost:8080/mark?x=0&y=0&rad=10000)deleteMark?secretKey=<тут секретный ключ>&idMark=4

idMark – айдишник метки, которую нужно удалить

Метод ничего не возвращает.

Клиентское API

1. Получить все метки относительно точки в определённом радиусе.(GET)

<http://localhost:8080/mark?x=0&y=0&rad=10000>

x, y – центральные, начальные координаты(double).

rad – радиус области меток(int, возможно будет double).

Если что-то пойдёт не так, то вернётся null. Если всё хорошо, то вернётся массив меток в формате json, пример:

[ {

"id": 15,

"x": 3213,

"y": 3323,

"type": "test",

"description": "",

"photoPath": "C:\\photos\\874634871.png",

"irrelevanceLevel": 1,

"deathTime": 1480015063000

},

{

"id": 16,

"x": 3213,

"y": 3323,

"type": "test",

"description": "sadsa",

"photoPath": "C:\\photos\\874634871.png",

"irrelevanceLevel": 2,

"deathTime": 1480025339000

},

{

"id": 17,

"x": 7000,

"y": 7000,

"type": "fefesfs",

"description": null,

"photoPath": null,

"irrelevanceLevel": 0,

"deathTime": 1480125557000

}]

Соответственно, если меток нет, то придёт пустой массив.

1. Добавление метки.(POST)(тут необходимо изменить хедер определённым образом, иначе сервер не обработает запрос (<http://stackoverflow.com/questions/29680158/how-to-send-multipart-form-data-with-retrofit> - для андроида) )

<http://localhost:8080/mark?x=54&y=987.54&type=sacxzcz&desc=dcasascas>&userId=1

x,y,type, userId – обязательные.

desc – описание метки, ограничение в 500 символов.

userId – айдишник пользователя, который поставил метку.

Также к запросу может быть прикреплена фотография.

Отсылка файла происходит под параметром photoFile.

После отправки данных в случае успеха, вернётся json созданной метки.

{

"id": 25,

"x": 54,

"y": 987.54,

"type": "sacxzcz",

"description": "dcasascas",

"photoPath": "C:\\photos\\1642001484img1.jpg",

"irrelevanceLevel": 0,

"deathTime": 1480177654887,

"user": 1

}

В противном случае null.

1. Изменение метки.(Post) (тут необходимо изменить хедер определённым образом, иначе сервер не обработает запрос (<http://stackoverflow.com/questions/29680158/how-to-send-multipart-form-data-with-retrofit> - для андроида))

http://localhost:8080/changeMark?id=25&desc=новое описание&irrel=1&photo=C:\\photos\\1642001484img1.jpg

id – обязательный. Всё остальное ситуативно.

Irrel – указывает, что метку отметили неактуальной, по факту в неё можно передавать всё что угодно, главное, чтобы не была пустая, когда она нужна.

type – изменение типа метки.

Если мы хотим изменить фотку, то передаём путь до фото в параметре photo и прикрепляем по аналогии с прошлым запросом новую фотку. Для простого добавления фотки, путь в параметрах не указывается.

Если всё хорошо, то возвращает обновлённую метку.

{

"id": 25,

"x": 54,

"y": 987.54,

"type": "sacxzcz",

"description": "новое описание",

"photoPath": "C:\\photos\\1642001484img1.jpg",

"irrelevanceLevel": 2,

"deathTime": 1480177654000

}

В противном случае null.

1. Загрузка фотки метки с сервера(GET)

<http://localhost:8080/photo?path=C:\\photos\\1642001484img1.jpg>

path – кидается путь, где лежит фото метки.

Можно использовать как и для прямой загрузки и сохранения, так и можно указать в теге

<img src=”/photo?path=C:\\photos\\1642001484img1.jpg ”/>

1. Получение данных пользователя(GET)

<http://localhost:8080/user?vkId=72245>

vkId – вы получаете после того как пользователь прошёл аутентификацию через vk.

Если пользователя нет в базе вернётся null.

Если он есть вернётся json.

{

"id": 1,

"firstName": "dsfsd",

"lastName": "dsfds",

"nickname": null,

"rating": 0,

"photoPath": null,

"vkId": 72245

}

1. Добавление пользователя в базу(обычный POST)

<http://localhost:8080/user?fName=Чувак&lName=Чуваков&vkId=5415845>

lName, fName – фамилия и имя, их вы тоже должны получить после аутенфикации vk.

При успешном добавление вернётся json.

{

"id": 3,

"firstName": "Чувак",

"lastName": "Чуваков",

"nickname": null,

"rating": 0,

"photoPath": null,

"vkId": 5415845

}

Иначе null.