

# Структура бд

Объяснить вопрос с лайками

id pydantic field UUID

## Структура того как хранится

"Secured info about user" - сами сделаете

```
{
  "user": {
    "id" : String,

    "mainInfo" : {
      "nickname": String,
      "description" : String,
      "image" : String,
    },
    "information" : {
      "firstName" : String,
      "secondName" : String,
      "dateOfBirth" : String,
      "gender" : String,
    },
    "followers" :[
      {
        "id" : String,
      },
      {
        "id" : String,
      },
      ....
    ],
    "followings" :[
      {
        "id" : String,
      },
      {
        "id" : String,
      },
      ....
    ],
    "posts" : [
      {
        "id" : String,
      },
      {
        "id" : String,
      },
      ....
    ],
    "likesId" : List<String>
  },
}
```

```
}
```

```
"post":  
{  
  "id" : String,  
  "userId" : String,  
  "nickname" : String,  
  
  "date" : String,  
  
  "image" : List<image>,  
  "description" : String,  
  
  "numberOfLikes" : int,  
  
  "likesId" : List<String>,  
  "commentsId" : List<String>,  
  
}
```

```
"comment":  
{  
  "id" : String,  
  "userId" : String,  
  "text" : String,  
  
  "date" : String,  
  
}
```

```
"like" : {  
  "id" : String,  
  "userId" : String,  
  "postId" : String,  
  
}
```

## Эндпоинты

"https://osetinogram/post\${post\_id}"

Вызывается при открытии конкретного поста

1. По id определяете пост.
2. По userId из поста определяете пользователя и получаете оттуда его аватарку.
3. Из множества commentsId выбираете (как? это вопрос отдельный, пока что условно рандомный, в будущем можно по дате сортировать) один. Через userId коммента получаете никнейм и аватарку комментатора.

4. Все полученное объединяете и отправляете в формате:

```
{  
  
  "userImage" : string,  
  
  "post" :  
  {  
    "id": String,  
    "userId" : String,  
    "nickname": string,  
  
    "date" : String,  
  
    "image" : List<String>,  
    "description": string,  
  
    "numberOfLikes" : int,  
  
  },  
  "comments" : {  
    "comment" : {  
      "id" : String,  
      "userId" : String,  
  
      "date" : String,  
  
      "text" : String,  
    },  
    "userInfo" : {  
      "userImage" : String,  
      "nickname" : String,  
    },  
  },  
  
},  
  
}
```

```
"https://osetinogram/comments${post_id}"
```

==Вызывается для получения списка лайкнувших.==

1. В выбранном посте пройти по списку лайков и для каждого лайка получить лайкнувшего пользователя( его аватарку и никнейм).

```
``` json
{
  "like" : [{
    "userId" : String,
    "userImage" : String,
    "nickname" : String,

    "id" : String,
  },

  {
    "userId" : String,
    "userImage" : String,
    "nickname" : String,

    "id" : String,
  },

  ...
}
```

"https://osetinogram/comments\${post\_id}"

Вызывается для получения списков комментировавших

1. По id поста получаем список id комментариев и выгружаем часть комментариев. По userId для комментариев получаешь аватарку и никнейм пользователя:

```
{
  "comments" : [{
    "id" : String,
    "userId" : String,
    "text" : String,

    "date" : String,

    "nickname" : String,
    "userImage" : String,
  },

  {
    "id" : String,
    "userId" : String,
    "text" : String,

    "date" : String,

    "nickname" : String,
    "userImage" : String,
  },
}
```

```
...  
]  
}
```

"https://osetinogram/user\${user\_id}"

Вызывается для перехода в профиль

1. По id пользователя из бд берем инфу. И части id подписчиков, подписок, постов догружаем изображения

```
{  
  "user":  
  {  
    "id" : String,  
  
    "mainInfo" :  
    {  
      "nickname": String,  
      "description" : String,  
      "image" : String  
    },  
    "information" :  
    {  
      "firstName" : String,  
      "secondName" : String,  
      "dateOfBirth" : String,  
      "gender" : String  
    },  
  
    "followers" : [ (всего 6 штук)  
    {  
      "id" : String,  
      "image" : String,  
    },  
    {  
      "id" : String,  
      "image" : String,  
    },  
    ....  
  ],  
  
    "followings" : [ (всего 6 штук)  
    {  
      "id" : String,  
      "image" : String,  
    },  
    {  
      "id" : String,  
      "image" : String,  
    },  
    ....  
  ],  
  
    "posts" : [ (какую то обговоренную часть т.е. штук 20)  
    {  
      "id" : String,  
      "image" : String,  
    },  
    {  
      "id" : String,  
      "image" : String,  
    },  
  ],  
}
```

```

        ....
    ],
    },
}

```

При просмотре всех подписчиков или подписок догружаются остальные

"https://osetinogram/feed\${user\_id}"

Вызывается для просмотра ленты

Обсудить способ хранения ленты разной для каждого пользователя

1. Посты сортируются в базе данных (в идеале их уже держать отсортированными в любой момент времени)
2. Выбирается определенное количество постов (пусть 20)
3. У каждого

```

{
  "feed": [
    {
      "posts" : [ (20 штук)
        {
          "idPost": String,
          "userId" : String,
          "images" : list<String>,
          "nicknameUser" : String,
          "numberOfLikes" : int,
        },
        {
          "idPost": String,
          "userId" : String,
          "images" : list<String>,
          "nicknameUser" : String,
          "numberOfLikes" : int,
        },
        ...
      ],
    },
    // ...
  ]
}

```

5. "https://osetinogram/settings\${user\_id}"

```

{
  "info": [
    {
      "id" : String,
      "mainInfo" :
      {

```

```
        "nickname": String,  
        "description" : String,  
        "image" : String,  
    },  
    "information" :  
    {  
        "firstName" : String,  
        "secondName" : String,  
        "dateOfBirth" : String,  
        "gender" : String  
    },  
    "secure" :  
    {  
        "password" : String,  
    }  
},  
]  
}
```

"https://osetinogram/search\${nickname}"

Вызывается при поиске

```
{  
    "user" :  
    {  
        "id" : String,  
        "userImage" : String,  
        "nickname" : String,  
    }  
}
```