

MongoDB

База данных в проекте «ЦИТИК»



Петров Михаил Иванович

333 группа

Содержание

1. Введение
2. Описание базы данных
3. Схема
4. Команда
5. Коллекции
6. Запросы
7. Демо

Введение

Цитик (Центр IT И Инженерных Компетенций).

Сайт с информацией о центре, выполненных и предстоящих работах, с инструментами коммуникации. На сайте студенты могут выкладывать проекты. Сайт генерирует по нему статистику используя API github. Работодатель анализирует, выбирает что понравилось и может предложить сотрудничество. Возможность просмотреть законченные, текущие и будущие проекты.

Описание базы данных

Основное:

Документоориентированная нереляционная СУБД NoSQL

Тип:

Облачная СУБД

Особенности:

Неограниченное горизонтальное масштабирование и высокая
производительность

Команда

Михаил:

1. Моделирование структуры бд
2. Реализация дизайна по макету
3. Взаимодействие бэкенда с фронтендом

Екатерина:

1. Продумывание структуры бд
2. Создание дизайнерского макета
3. Тестирование проекта на всех этапах разработки




Леонид:

1. Реализация структуры бд
2. Разработка бэкенда
3. Исследование стека технологий

Схема

Коллекции:

1. Пользователи
2. Проекты
3. Работодатели

Collection Name ^	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
employers	3	371.7 B	1.1 KB	1	32.8 KB	
projects	8	452.6 B	3.6 KB	1	36.9 KB	
users	3	155.0 B	465.0 B	1	36.9 KB	

Коллекции

Пользователи:

```
db.createCollection("users")
```

```
db.users.insertOne({
```

```
    "login": "mike_petrov",
```

```
    "password": "123456",
```

```
    "name": "Mike Petrov",
```

```
    "email": "mis5116@yandex.ru",
```

```
    "likes": [],
```

```
    "dislikes": [],
```

```
})
```

```
1    login : "mike "           String
2    password : "123456 "      String
3    name : "Mike Petrov "     String
4    mail : "mis5116@yandex.ru " String
5    > likes : Array           Array
6    > dislikes : Array        Array
```

Коллекции

Работодатели:

```
db.createCollection("employers")
```

```
db.users.insertOne({
```

```
    "surname": "Севрюков",
```

```
    "name": "Сергей",
```

```
    "patronymic": "Юрьевич",
```

```
    "contacts": "...",
```

```
    "description": "...",
```

```
})
```

```
1    surname : "Севрюков "           String
2    name : "Сергей "               String
3    patronymic : "Юрьевич "        String
4    contacts : "http://ssevryukov.moikrug.ru/ " String
5    description : "Старший преподаватель кафедры технологии п..." String
```


Коллекции

Проекты:

```
db.createCollection("projects")
```

```
db.users.insertOne({
```

```
    "name": "CITEK",
```

```
    "description": "ЦИТИК ...",
```

```
    "category": "Web-app",
```

```
    "status": "В разработке",
```

```
    "likes": [],
```

```
    "dislikes": [],
```

```
    "linkGit": "...",
```

```
    "date": timestamp,
```

```
})
```

```
1   name : "CITEK "                               String
2   description : "ЦИТИК– центр информационных технологий и и..." String
3   category : "Web-app "                          String
4   status : "В разработке "                       String
5   > likes : Array                                Array
6   > dislikes : Array                             Array
7   linkGit : "https://github.com/NikitaKnyazev/CITEK " String
8   date : 1575120000                               Int32
```

Запросы

Подключение

```
cluster = MongoClient("mongodb+srv://Leonid:Factor_9@cluster0-e2dix.mongodb.net/test?retryWrites=true&w=majority")
db = cluster['test']
projects = db['projects']
users = db['users']
employers = db['employers']
```

Запросы

Стягивание всех проектов

```
@app.route('/projects', methods=['POST'])
def post():
    results = projects.find({})
    ans = []
    for result in results:
        ans.append(result)
    return json.dumps(ans)
```

Запросы

CRUD

```
@app.route('/project/update', methods=['POST'])
def update_project():
    x = request.json

    name = x['name']
    projects.update_one({ "name": "Mike" }, { "$set": { "name": name } })

    return true
```

```
@app.route('/project/delete', methods=['POST'])
def delete_project():
    x = request.json

    id = x['id']
    projects.delete_one({ "id": id })

    return true
```

```
@app.route('/project/create', methods=['POST'])
def create_project():
    x = request.json

    post = {
        'id': projects.count() + 2,
        'name': x['name'],
        'description': x['description'],
        'category': x['category'],
        'status': x['status'],
        'likes': [],
        'dislikes': [],
        'linkGit': x['linkGit'],
        'date': int(time.time()),
    }

    projects.insert_one(post)

    return post
```

Запросы

Сохранение лайков проекту и пользователю

```
@app.route('/project/like', methods=['POST'])
def update():
    x = request.json

    projectId = x['id']
    userMail = x['user_mail']

    flag = x['type']
    if flag:
        projects.update_one({'id': projectId}, {'$push':{ 'likes': userMail }})
        users.update_one({'mail': userMail}, {'$push':{ 'likes': projectId }})
        projects.update_one({'id': projectId}, {'$pull':{ 'dislikes': userMail }})
        users.update_one({'mail': userMail}, {'$pull':{ 'dislikes': projectId }})
    else:
        users.update_one({'mail': userMail}, {'$push':{ 'dislikes': projectId }})
        projects.update_one({'id': projectId}, {'$push':{ 'dislikes': userMail }})
        projects.update_one({'id': projectId}, {'$pull':{ 'likes': userMail }})
        users.update_one({'mail': userMail}, {'$pull':{ 'likes': projectId }})
```

Запросы

Реализация авторизации и регистрации

```
@app.route('/auth', methods=['POST'])
def authorization():
    x = request.json
    login = x['login']
    password = x['password']
    person = users.find_one({'login': login})

    if not person:
        return json.dumps({'error': 'login'})

    if person['password'] != password:
        return json.dumps({'error': 'password'})
    else:
        return json.dumps(person)
```

```
@app.route('/reg', methods=['POST'])
def registration():
    x = request.json
    login = x['login']
    password = x['password']
    name = x['name']
    mail = x['mail']
    status = x['status']

    if users.find_one({'mail': mail}):
        return json.dumps({'error': 'mail'})

    post = {
        '_id': users.count() + 1,
        'login': login,
        'password': password,
        'name': name,
        'mail': mail,
        'status': status,
        'likes': [],
        'dislikes': [],
    }

    users.insert_one(post)
    return json.dumps(post)
```

Запросы

Фильтрация проектов по времени

```
@app.route('/projects/filter', methods=['POST'])
def filter():
    x = request.json

    timestamp = x['timestamp']
    type = x['type']

    if type == 'last':
        results = projects.find({'date': {'$lt': timestamp}}, {'_id': False})
    elif type == 'current':
        results = projects.find({'$and' : [{'date' : {'$gt': timestamp[0]}}, {'date': {'$lt': timestamp[1]}}] }, {'_id': False})
    else:
        results = projects.find({'date': {'$gt': timestamp}}, {'_id': False})
    ans = []
    for result in results:
        ans.append(result)
    return json.dumps(ans)
```

Демо

<https://askpro.online/>