Jiskefet

Amin Chihab, Bastiaan Reinalda, Calvin Huynh,
Misha Rigot, Ramon Gill

December 16, 2018

Overview

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

1 API

2 UI

3 Documentation

4 Performance

5 Planning

6 Deployment

Application Programming Interface

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

endpoint: `api/logs`

| GET | /logs | 🔒 |

```
▼{,…}
    count: 1002
  ▼ logs: [{logId: 1002, subtype: "run", origin: "human", creationTime: "2018-1
    ▶ 0: {logId: 1002, subtype: "run", origin: "human", creationTime: "2018-1
    ▶ 1: {logId: 1001, subtype: "run", origin: "human", creationTime: "2018-1
    ▶ 2: {logId: 363, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 3: {logId: 584, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 4: {logId: 954, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 5: {logId: 606, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 6: {logId: 385, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 7: {logId: 911, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 8: {logId: 56, subtype: "run", origin: "process", creationTime: "2018-1
    ▶ 9: {logId: 720, subtype: "run", origin: "process", creationTime: "2018-
    ▶ 10: {logId: 228, subtype: "run", origin: "process", creationTime: "2018
    ▶ 11: {logId: 132, subtype: "run", origin: "process", creationTime: "2018
    ▶ 12: {logId: 635, subtype: "run", origin: "process", creationTime: "2018
    ▶ 13: {logId: 430, subtype: "run", origin: "process", creationTime: "2018
    ▶ 14: {logId: 563, subtype: "run", origin: "process", creationTime: "2018
    ▶ 15: {logId: 439, subtype: "run", origin: "process", creationTime: "2018
```

Application Programming Interface

path: `src/controllers/log.controller.ts`

```typescript
/**
 * Get all logs. /logs
 */
@Get()
async findAll(@Query() query?: QueryLogDto): Promise<{ logs: Log[], count: number }> {
    return await this.logService.findAll(query);
}
```

Application Programming Interface

path: `src/services/log.service.ts`

```typescript
/**
 * Returns all Logs from the db.
 */
async findAll(queryLogDto: QueryLogDto): Promise<{ logs: Log[], count: number }> {
    let query = await this.repository.createQueryBuilder('log')
        .innerJoinAndSelect('log.user', 'user')
        .where('title like :title', {
            title: queryLogDto.searchterm ? `%${queryLogDto.searchterm}%` : '%'
        })
        .andWhere('subtype like :subtype', {…
        })
        .andWhere('origin like :origin', {…
        });

    if (queryLogDto.startCreationTime) {
        await query.andWhere('creation_time >= :startCreationTime', {
            startCreationTime: queryLogDto.startCreationTime
        });
    }

    if (queryLogDto.endCreationTime) {…
    }

    if (queryLogDto.logId) {…
    }

    if (queryLogDto.orderBy) {
        query = query.orderBy(
            `log.${queryLogDto.orderBy}`,
            queryLogDto.orderDirection || OrderDirection.asc
        );
    }
    const result = await query
        .skip((+queryLogDto.pageNumber - 1 || 0) * +queryLogDto.pageSize || 0)
        .take(+queryLogDto.pageSize || 25)
        .getManyAndCount();
```

Application Programming Interface

- Nest
- Typeorm (MariaDB)
- Jest
- Structure

User Interface

- Mithril
- Typescript
- Logs and Runs
- Markdown
- Authentication
- Token generation

Documentation

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

- Markdown
- Instructions

Performance

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
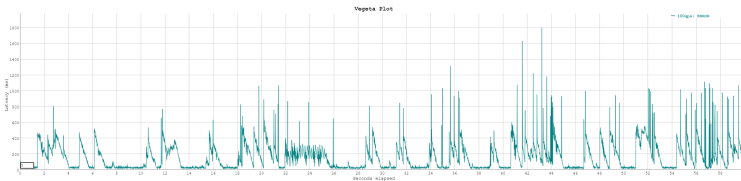Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

- Benchmarks
- Vegeta
- Jest-Artillery

Planning

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

- CERN SSO (logout)
- Front end testing
- Refactoring API
- Atomic Design
- Redux
- CI and more tests

Deployment

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

- Ansible
    - API
    - UI
    - Database
    - Jenkins CI
- Jenkins CI
    - Linting
    - Testing
    - Codecov

Deployment

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

jiskefet-deploy - Directory Structure Overview

- ci
- docs
- files
- tasks
  - change-env-variables.yml
  - create-jiskefet-user.yml
  - firewall.yml
  - git.yml
  - mysql.yml
  - nginx.yml
  - nodejs.yml
  - npm.yml
  - pm2.yml
  - unarchive.yml
- .gitignore
- ansible.cfg
- ansible.config.json
- ansible.config.json.template
- hosts
- LICENSE
- README.md
- site.yml

Deployment

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

Common tasks

path: `site.yml`

```
- name: apply common configuration to all nodes
  hosts: all
  remote_user: root
  tasks:

    # Create jiskefet-api user
    - include: tasks/create-jiskefet-user.yml
```

Deployment

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

Web server
path: `site.yml`

```
tasks:
  # Install git
  - include: tasks/git.yml
    when: use_local_repository == "no"

  # Unarchive projects
  - include: tasks/unarchive.yml
    when: use_local_repository == "yes"

  # Setting environment variables
  - include: tasks/change-env-variables.yml

  # Install Node.js
  - include: tasks/nodejs.yml

  # Do npm install
  - include: tasks/npm.yml

  # Starts firewalld service
  - include: tasks/firewall.yml

  # Install and run nginx
  - include: tasks/nginx.yml

  # Install and run pm2
  - include: tasks/pm2.yml
```

Deployment

Jiskefet

Amin Chihab,
Bastiaan
Reinalda, Calvin
Huynh, Misha
Rigot, Ramon Gill

API

UI

Documentation

Performance

Planning

Deployment

Database server
path: `site.yml`

```yaml
tasks:

    # Starts firewalld service
    - include: tasks/firewall.yml

    # Install MySql (MariaDB)
    - include: tasks/mysql.yml
```