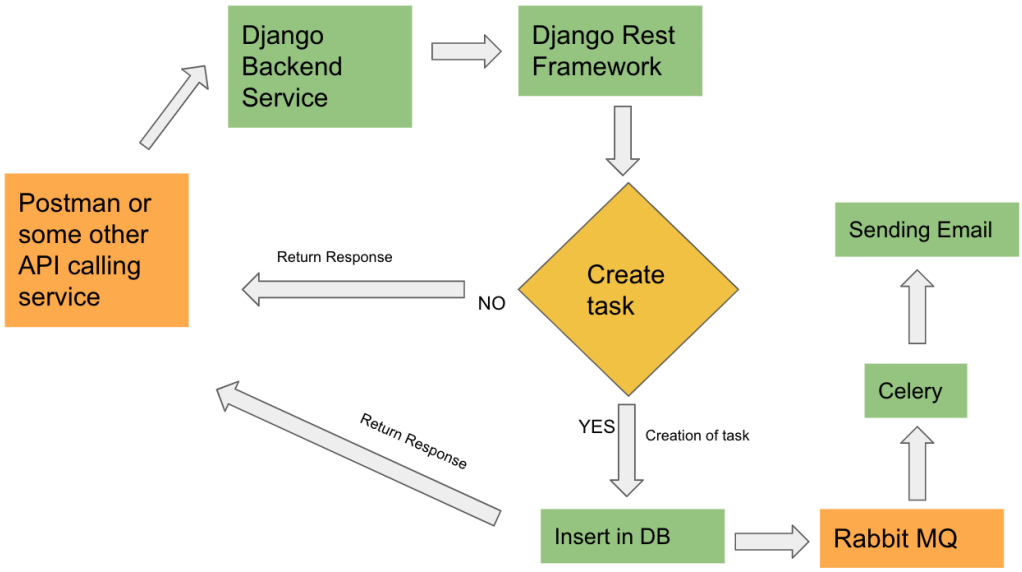
Design: FreJun assignment JIRA clone

Problem statement:

Build a simple web application which helps users to create a task and associate it with a team. The team leader of the team will assign the team member or team members to the task. The team member will update the status of the task.

Design:



In order to design this application we will be needing

1. Postman — To make requests
2. Django — A backend server
3. DRF — To make ResT APIs
4. RabbitMQ — Message broker for communication between celery and Server

Tables needed to be designed in schema:

1. Users — To store users, used AbstractUser to enhance usability
2. Tasks — To store all the tasks
3. Teams — To keep track of teams
4. Comments — To keep track of comments made on tasks

Note:

1. Used token based authentication to perform certain actions and restrict users.
2. Assigned users their separate groups and given permissions accordingly.
3. Following email template would be sent to user on creation of task

ashishbhardwajexcel@gmail.com

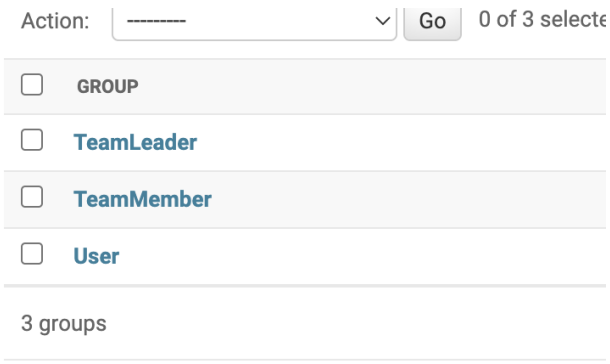
to me

Heyy Robin Sharma a new task has been created for your team please check your JIRA on time

Users

Have created three types of user groups, and have given permissions accordingly

1. User – General user of organisation – Product, operations, etc
2. TeamLeader — They are team leader
3. TeamMemeber — They are team members



Teams


```python
class Team(models.Model):
    name = models.CharField(max_length=20, blank=False, unique=True)
    team_leader = models.ForeignKey(User, related_name='team_leader',limit_choices_to={'groups__name': "TeamLeader"} ,on_delete=mod
    team_memebers = models.ManyToManyField(User, limit_choices_to={'groups__name': "TeamMember"})
    created_at = models.DateTimeField(editable=False, default=timezone.localtime(timezone.now()))
    modified_at = models.DateTimeField(default=timezone.localtime(timezone.now()))

    def __str__(self):
        return str(self.name)
```

Tasks

TASK NO	NAME	SPRINT	STATUS	TYPE OF TASK	CREATED AT	MODIFIED AT
7	Alphonse	1	TODO	BUG	Aug. 4, 2022, 7:30 a.m.	Aug. 4, 2022, 8:13 a.m.
6	testing	1	TODO	TASK	Aug. 4, 2022, 7:28 a.m.	Aug. 4, 2022, 7:28 a.m.
5	skit	1	TODO	TASK	Aug. 4, 2022, 7:22 a.m.	Aug. 4, 2022, 7:22 a.m.
4	fill spaces	1	TODO	TASK	Aug. 4, 2022, 7:17 a.m.	Aug. 4, 2022, 7:17 a.m.
3	fill items	1	TODO	TASK	Aug. 4, 2022, 6:10 a.m.	Aug. 4, 2022, 6:10 a.m.
2	fill metrics	1	TODO	TASK	Aug. 4, 2022, 6:06 a.m.	Aug. 4, 2022, 6:06 a.m.
1	fill data	1	TODO	BUG	Aug. 4, 2022, 6:06 a.m.	Aug. 4, 2022, 8:09 a.m.

7 tasks

```python
class Task(models.Model):
    name = models.CharField(max_length=200, blank=False)
    task_no = models.AutoField(primary_key=True)
    description = models.CharField(max_length=800, blank=False)
    team = models.ForeignKey(Team, on_delete=models.CASCADE)
    created_by = models.ForeignKey(User,on_delete=models.PROTECT)
    sprint = models.IntegerField()
    type_of_task = models.CharField(max_length=4, choices=TASK_TYPE, default='task')
    status = models.CharField(max_length=22, choices=TASK_STATUS, default='todo')
    priority = models.CharField(max_length=6, choices=PRIORITY, default='medium')
    created_at = models.DateTimeField(editable=False, default=timezone.localtime(timezone.now()))
    modified_at = models.DateTimeField(default=timezone.localtime(timezone.now()))

    def __str__(self):
        return str(self.name)
```

Some screenshot of the API Responses

All Tasks

Task

List all Tasks, or create a new Task.

```
GET /tasks/api/task/
```

```
HTTP 200 OK
Allow: GET, POST, PUT, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "tasks": [
        {
            "name": "fill data",
            "task_no": 1,
            "description": "This split missing data in the metrics",
            "team_id": 1,
            "created_by_id": 3,
            "sprint": 1,
            "type_of_task": "bug",
            "status": "todo",
            "priority": "medium",
            "created_at": "2022-08-04T06:06:26.744877Z",
            "modified_at": "2022-08-04T08:09:08.984524Z"
        },
        {
            "name": "fill metrics",
            "task_no": 2,
            "description": "This task is to fill missing metrics in the metrics",
            "team_id": 3,
            "created_by_id": 5,
            "sprint": 1,
            "type_of_task": "task",
            "status": "todo",
            "priority": "medium",
            "created_at": "2022-08-04T06:06:26.744877Z",
```

All teams

```
HTTP 200 OK
Allow: GET, POST, PUT, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "status": 1,
    "teams": [
        {
            "name": "infinite",
            "team_leader__name": "teamleader1"
        },
        {
            "name": "spark",
            "team_leader__name": "teamleader2"
        },
        {
            "name": "lucas",
            "team_leader__name": "teamleader2"
        }
    ]
}
```

Other requests can be made in postman

Closing points:

1. Task ID can be improved more likely to Project_name–some number, same like we have in JIRA
2. Haven't done much implementation in Comments, due to lesser bandwidth that can be implemented too.
3. Implementation of time logging assignee and reviewer can be done just like in JIRA.
4. Logging for errors can be done
5. Sending emails on each task update can also be done.
6. Though i was tight on bandwidth due to some ongoing project changes, still I have completed all the requirements that were there in the assignment.
7. Moreover its just a basic app, we can bring in a lot of changes and will do those in future.

Created By - Ashish Bhardwaj
ashishbhardwajexcel@gmail.com