API REST
if the resource need authorization, must use the session id

POST

Path: /api/login

Description: Login to system

Authorization: No

Parameters:

- “email”: string
- “pass”: string

Response format:

```
{
“status”: “ok / error”,
“reason”: reason”,
“session_id”: session_id
}
```

GET

Path: /api/logout

Description: Logout from system

Authorization: Yes

Parameters: None

Response format:

```
{

“status”: status

}
```

GET

Path: /api/users

Description: Obtain a list of users registered in the system

Authorization: Yes

Parameters: None

Reponse format:

list of users [{“id”: Id},] if the role is superuser else {“status”: “error”}

GET

Path: /api/users/{user_num}

Description: Obtain a user info

Authorization: Yes

Parameters: None

Response fomat:

```
{
“id”:id,

 “name”: name,

“lastname”: lastname,

“email”:email
}
```

POST

Path: /api/users

Description: Create a new user with info provided in req

Authorization: No

Parameters:

- “name”: string
- “lastname”: string
- “email”: string
- “pass”: string

Response format:

```
{
“status”: ok / error,
“reason”: reason,
“id”: id
}
```

PUT

Path: /api/users/{user_num}

Description: Update user info with info provided in req.

Authorization: Yes

Parameters: {“name”: name, “lastname”: lastname, “pass”:pass}

Response format:

```
{
“status”: ok / error
“reason”: reason
}
```

DELETE

Path: /api/users/{user_num}

Description: Delete a user account

Authorization: Yes

Parameters: None

Response format:

```
{
“status”: ok / error
“reason”:
Reason
}
```

GET

Path: /api/users/{user_num}/projects

Description: Obtain a list of user projects

Authorization: Yes

Parameters: None

Response format:

list of projects of user [{“id”:id},]

GET

Path: /api/users/{id_user}/projects/{project_num}

Description: Obtain project info

Authorization: Yes

Parameters: “id_user”: Integer
“project_num”: Integer

Response:

```
{
        “id”: Integer
        “name”: Text
        “info”: Text
}
```

POST

Path: /api/users/{user_num}/projects

Description: Create a new project

Authorization: Yes

Parameters: {“info”: info}

Response format:

```
{
“status”: ok / error
 “reason”: reason,
“id”: id
```

```
        }
```

PUT

Path: /api/users/{user_num}/projects/{project_num}

Description: Update project info

*Auth*orization: Yes

Parameters:

- “info”: info

Response format

```
        {
        “status”: ok / error
        “reason”: reason,
        “id”: id
        }
```

DELETE

Path: /api/users/{num}/project/{num}

*Desc*ription: Delete project

Authorization: Yes

Parameters: None

Response format:

```
        {
        “status”: ok / error
        “reason”: reason
        }
```

GET

Path: /api/users/{user_num}/project/{project_num}/ranking

Description: Get list of ranking in project

Authorization: Yes

Parameters: user_num: Integer
project_num: Integer

Response format:

Return list of ranking[{"id_user":id, "name":name, "lastname":lastname, "points":points}]

GET

Path: /api/users/{user_num}/project/{project_num}/developers

Description: Obtain a list of users from project

Authorization: Yes

Parameters: user_num: Integer
project_num: Integer

Response format:

Return list of devs [{“id”: Integer}]

POST

Path: /api/users/{user_num}/project/{project_num}/developers/

Description: Add developer to the project

Authorization: TL / PM

Parameters: *“user_num”: Integer*
“project_num”: Integer

Response format:

```
{
“status”: ok / error
“reason”: reason
}
```

DELETE

Path: /api/users/{user_num}/project/{project_num}/developers/{developer_id}

Description: Eliminate a developer from the project

Authorization: TL / PM

Response format:

```
{
“status”: ok / error

}
```

PUT

Path: /api/users/{user_num}/project/{project_num}/developers/{developer_id}

Description: Update developer info

Authorization: TL

Parameters:

- “task”: task
- “role”: role

Response format:

```
{
“task”: task
“role”: role
}
```

GET

Path: /api/users/{id_user}/projects/{project_num}/sprints

Description: Obtain list of sprint of project

*Auth*orization: Yes

Parameters: “id_user”: Integer
“project_num”: Integer

Response format:

```
        {
                Sprint_list: [
                {
                “id”: integer

                ]
        }
```

GET

Path: /api/users/{id_user}/projects/{id_project}/sprints/{sprint_id}

Description: Get sprint info

Authorization: Yes

Parameters: “id_user”: Integer
“id_project”: Integer
“sprint_id”: Integer

Response format:

```
        {
        “id”: integer
        “duration”: integer;
        “id_project”: integer;
        },
```

POST

Path: /api/users/{num}/project/{num}/sprints

Description: Create new sprint

Authorization: Yes

Parameters: {

- “id_proj”: integer
- “duration”: integer

Response format:

```
{
“id”: integer
“duration”: integer;
“id_project”: integer;
},
```

DELETE

Path: /api/users/{user_num}/project/{project_num}/sprints/{sprint_id}

Description: Delete a sprint

Authorization: Yes

Parameters: None
Response:

```
{
“status”: “deleted / error”
“reason”: “reason” / null
}
```

PUT

Path: /api/users/{num}/project/{num}/sprints/{id}

Description: Update sprint

Authorization: Yes

Parameters:

- “duration”: integer

Response format:

```
{
“duration”: integer;
}
```

GET

Path: /api/users/{user_num}/projects/{id_projects}/sprints/{sprint_id}/tasks

Description: Get list of tasks of sprint

Authorization: Yes

Parameters: “user_num”: Integer
“id_project”: Integer
“sprint_id”: Integer

Response format:

```
{
        Task_list: [
        {
        “id”: integer

        },

}
```

GET

Path: /api/users/{id_user}/projects/{id_project}/sprint/{id_sprint}/tasks/{id_task}

Description: Get task info

Authorization: Yes

Parameters: “id_user”: Integer
“id_project”: Integer
“id_sprint”: Integer
“id_task”: Integer

Response format:

```
{
        “id”: integer
        “desc”: text
        “comment”: text
        “priority”: integer
        “duration”: integer;
        “points”: integer;
        “finalized”: integer;
        “id_user”: integer;
        “id_requirement”: integer;

}
```

POST

Path: /api/users/{user_num}/project/{project_num}/sprints/{sprint_id}/tasks

Description: Create a new task

Authorization: Yes

Parameters:

- “desc”: desc,
- “comment”: comment,
- “priority”: priority,
- “duration”: duration,
- “points”: points,
- “finalized”: finalized,
- “id_user”: id_user,
- “id_req”: id_req

Response format:

```
{
“desc”: desc,
“comment”: comment,
“priority”: priority,
“duration”: duration,
“points”: points,
“finalized”: finalized,
“id_user”: id_user,
“id_req”: id_req
}
```

- Delete a task

DELETE

Path: /api/users/{user_num}/project/{project_num}/sprints/{sprint_id}/tasks/{task_id}

Description: Delete a task

Authorization: Yes

Parameters: None

Response format:

```
{
“status”: “deleted”/ “error”
“reason”: “reason” / null
}
```

- Update task

PUT

Path: /api/users/{num}/project/{num}/sprints/{id}/tasks/{id}

Description: Update task info

Authorization: Yes

Parameters:

- “desc”: desc,
- “comment”: comment,
- “priority”: priority,
- “duration”: duration,
- “points”: integer,
- “finalized”: ineger,
- “id_user”: integer,
- “id_req”: integer

Response format:

{
“desc”: desc,
“comment”: comment,
“priority”: priority,
“duration”: duration,
“points”: integer,
“finalized”: integer,
“id_user”: integer
“id_req”: integer
“status”: “updated” / “error”
“reason”: “reason” / null
}