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}/devs**

*Description:* Add developer to the project

*Authorization:* TL / PM

*Parameters:*

* “id\_user”: id\_user

*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

“reason”: reason

}

**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*

*}*