

Important notes:

- 1) All STRING parameters should be submitted in quotes (like this 'some_string')
- 2) All parameters of all queries should be called like the same fields in the DB tables
- 3) All parameters in all queries are strings by default
- 4) Parameters in squared brackets are not required
- 5) According http standards status is a code 2** for success, 3** and 4** for different errors so status is self-documented variable in response
- 6) In all responses log is some message that may be useful for user (maybe showed in the error window)
- 7) All parameters except the {id} are transferred in JSON format

Users management

Post

path:/login

Description: Login to system

Parameters: email,password, [name]

Response Format: {status, log}

Post

path: /login/forgot-password

Parameters: email:STRING

Description: check if the email is valid and generates a new password (should be removed in the future)

Response Format: {status, log, new_password}

Post

path:/users

Description: Creates a new user

Parameters: name, email, password :STRING

Response Format: {status, log, new_user_id}

Get

path:/users

Description: Returns the list containing all users in JSON format

Parameters:

Response Format: {status, log, users: ARRAY[id: INTEGER, email, name]}

Get

path:/users/{id}/

Description: Returns the name and an email user with current id

Parameters: id: INTEGER

Response Format: {status, log, id: INTEGER, email, name}

Put

path:/users/{id}/

Description: Updates the information about user

Parameters: id: INTEGER, [email, name, password]

Response Format: {status, log}

Delete

path:/users/{id}/

Description: Deletes the user

Parameters: id: INTEGER

Response Format: {status, log}

Projects management

Post

path:/projects

Description: Creates a new project; user_id is the id of the owner

Parameters: name, user_id: INTEGER, [start_time, end_time]

Response Format: {status, log, id}

Get

path:/projects

Description: Returns the list containing all projects

Parameters:

Response Format: {status, log, projects: ARRAY[id: INTEGER, name, user_id: INTEGER, start_time, end_time]}

Get

path:/projects/{id}/

Description: Returns the information about project with the id provided; user_id is the id of the project owner

Parameters: id: INTEGER

Response Format: {status, log, id: INTEGER, name, user_id: INTEGER, start_time, end_time}

Put

path:/projects/{id}/

Description: Updates the information about project

Parameters: id: INTEGER, [name, user_id: INTEGER, start_time, end_time]

Response Format: {status, log}

Delete

path:/projects/{id}/

Description: Deletes the project

Parameters: id: INTEGER

Response Format: {status, log}

Sprints management

Post

path:/sprints

Description: Creates a new sprint;

Parameters: name, project_id: INTEGER, [start_time, end_time]

Response Format: {status, log, id}

Get

path:/sprints

Description: Returns the list containing all sprints

Parameters:

Response Format: {status, log, sprints: ARRAY[id: INTEGER, name, user_id: INTEGER, start_time, end_time]}

Get

path:/sprints/{id}/

Description: Returns the information about sprint with the id provided

Parameters: id: INTEGER

Response Format: {status, log, id: INTEGER, name, project_id: INTEGER, start_time, end_time}

Put

path:/sprints/{id}/

Description: Updates the information about sprint

Parameters: id: INTEGER, [name, project_id: INTEGER, start_time, end_time]

Response Format: {status, log}

Delete

path:/sprints/{id}/

Description: Deletes the sprint

Parameters: id: INTEGER

Response Format: {status, log}

Stories management

Post

path:/stories

Description: Creates a new story;

Parameters: name, sprint_id: INTEGER

Response Format: {status, log, id}

Get

path:/stories

Description: Returns the list containing all stories

Parameters:

Response Format: {status, log, stories: ARRAY[id: INTEGER, name, sprint_id: INTEGER]}

Get

path:/stories/{id}/

Description: Returns the information about story with the id provided

Parameters: id: INTEGER

Response Format: {status, log, id: INTEGER, name, sprint_id: INTEGER}

Put

path:/stories/{id}/

Description: Updates the information about story

Parameters: id: INTEGER, [name, sprint_id: INTEGER]

Response Format: {status, log}

Delete

path:/stories/{id}/

Description: Deletes the story

Parameters: id: INTEGER

Response Format: {status, log}

Tasks management

Post

path:/tasks

Description: Creates a new task;

Parameters: name, story_id: INTEGER, [points: INTEGER, status: INTEGER, completeness: INTEGER]

Response Format: {status, log, id}

Get

path:/tasks

Description: Returns the list containing all tasks

Parameters:

Response Format: {status, log, tasks: ARRAY[id: INTEGER, name, story_id: INTEGER, points: INTEGER, status: INTEGER, completeness: INTEGER]}

Get

path:/tasks/{id}/

Description: Returns the information about task with the id provided

Parameters: id: INTEGER

Response Format: {status, log, id: INTEGER, name, story_id: INTEGER, points: INTEGER, status: INTEGER, completeness: INTEGER}

Put

path:/tasks/{id}/

Description: Updates the information about the task

Parameters: id: INTEGER, [name, story_id: INTEGER, points: INTEGER, status: INTEGER, completeness: INTEGER]

Response Format: {status, log}

Delete

path:/tasks/{id}/

Description: Deletes the task

Parameters: id: INTEGER

Response Format: {status, log}

Comments management

Post

path:/projects-comments

Description: Creates a new project comment;

Parameters: content, user_id: INTEGER, project_id: INTEGER

Response Format: {status, log, id}

Get

path:/projects-comments

Description: Returns the list containing all comments

Parameters:

Response Format: {status, log, tasks: ARRAY[id: INTEGER, content, user_id: INTEGER, project_id: INTEGER]}

Get

path:/projects-comments/{id}/

Description: Returns the information about task with the id provided

Parameters: id: INTEGER

Response Format: {status, log, id: INTEGER, name, story_id: INTEGER, points: INTEGER, status: INTEGER, completeness: INTEGER}

Put

path:/projects-comments/{id}/

Description: Updates the information about the project comment

Parameters: id: INTEGER, [content, user_id: INTEGER, project_id: INTEGER]

Response Format: {status, log}

Delete

path:/project-comments/{id}/

Description: Deletes the project comment

Parameters: id: INTEGER

Response Format: {status, log}

The same API for sprints-comments, stories-comments and tasks-comments.

Relationship information

Get

path:/projects-sprints/{id}

Description: get all the sprints of a given project

Parameters: none

Response Format: {sprints: ARRAY[id: INTEGER, name, project_id: INTEGER, start_time, end_time]}

Get

path:/sprint-stories/{id}

Description: get all the stories of a given sprint

Parameters: none

Response Format: {stories: ARRAY[id: INTEGER, name, sprint_id: INTEGER]}

Get

path:/story-task/{id}

Description: get all the tasks of a given story

Parameters: none

Response Format: {tasks: ARRAY[id: INTEGER, name, story_id: INTEGER, points: INTEGER, status: INTEGER, completeness: INTEGER]}