

REST API Design

for

Professional Developer's Tool

Prepared by

Team Name: Milan3

Federico Reghenzani federico1.reghenzani@mail.polimi.it

Filippo Pagano filippo.pagano@mail.polimi.it

Mikel Vuka mikel.vuka@mail.polimi.it

University: Politecnico di Milano – Italy



Revision History

| Date (dd.mm.yyyy) | Version | Description | Author(s) |
|----------------------|---------|---|---|
| 09.11.14 | 0.1 | First draft | Federico Reghenzani, Mikel Vuka, Filippo Pagano |
| 14.11.14 | 0.2 | Added summary. Added stats and chat specification. Minor updates on other sections | Federico Reghenzani, Mikel Vuka, Filippo Pagano |
| 23.11.14 | 0.3.1 | Added possibility to retrieve project list of other user. | Federico Reghenzani |
| 25.11.14 | 0.3.2 | Added an exception to /account/userinfo | Federico Reghenzani |

Table of Contents

| | |
|---------------------------|----|
| Summary..... | 4 |
| Common exceptions..... | 5 |
| Data format..... | 5 |
| Account..... | 5 |
| Projects..... | 9 |
| Projects - Backlog..... | 10 |
| Projects - PBIs..... | 11 |
| Projects - Sprintlog..... | 12 |
| Projects - Tasks..... | 14 |
| Statistics..... | 15 |
| Chat system..... | 16 |

Summary

| Method | URL | Short Description |
|--------|--|---|
| GET | /account/userinfo | Retrives information about an user. |
| POST | /account/register | Registers a new account. |
| GET | /account/langs | Lists all possible languages and programmingLanguages. |
| POST | /account/edit | Edits the current user profile. |
| POST | /account/login | Login.. |
| POST | /account/recoverpassword | Sends a new password to user's email. |
| GET | /account/listdevelopers | Retrives list of all developers. |
| GET | /projects/list | Retrives list of projects. |
| POST | /projects/create | Creates a new project. |
| POST | /projects/{idproj}/adddeveloper | Adds a developer to a project. |
| POST | /projects/{idproj}/remdeveloper | Removes a developer from a project. |
| POST | /projects/{idproj}/edit | Edits a project. |
| GET | /projects/{idproj}/getbacklog | Returns project's backlog, if it exists. |
| POST | /projects/{idproj}/createbacklog | Creates a new backlog. |
| POST | /projects/{idproj}/removebacklog | Delete a backlog. |
| POST | /projects/{idproj}/pbis/create | Create a new PBI. |
| POST | /projects/{idproj}/pbis/{idpbis}/edit | Edit a PBI. |
| POST | /projects/{idproj}/pbis/{idpbis}/delete | Delete a PBI. |
| GET | /projects/{idproj}/sprintlogs/list | Lists all sprintlogs. |
| POST | /projects/{idproj}/sprintlogs/create | Creates a new sprintlogs. |
| GET | /projects/{idproj}/sprintlogs/{idsprintlog}/listpbis | Lists all PBIs assigned to specific sprintlog. |
| POST | /projects/{idproj}/sprintlogs/{idsprintlog}/addpbi | Assigns a PBI to a specific sprintlog. |
| POST | /projects/{idproj}/sprintlogs/{idsprintlog}/removepbi | Deassigns a PBI from a specific sprintlog. |
| POST | /projects/{idproj}/sprintlogs/{idsprintlog}/delete | Deletes a sprintlog. |
| GET | /projects/{idproj}/pbis/{idpbis}/listtasks | Lists all tasks of a PBI |
| POST | /projects/{idproj}/pbis/{idpbis}/createtask | Creates a new task to PBI |
| POST | /projects/{idproj}/pbis/{idpbis}/tasks/{idtask}/edit | Edits task of a PBI |
| POST | /projects/{idproj}/pbis/{idpbis}/tasks/{idtask}/delete | Deletes task of a PBI |
| GET | /stats/devpoints | Retrives points of developers. |
| GET | /chat/{idproj}/list | Returns the list of available chat of specified project |
| POST | /chat/{idproj}/create | Creates a new chat into indicated project |
| GET | /chat/{idproj}/msg/{idchat}/list | Returns the last 20 messages of the chat |
| POST | /chat/{idproj}/msg/{idchat}/send | Sends a new message |
| GET | /chat/{idproj}/msg/{idchat}/polling | Polling requests for new message |

Common exceptions

All common exceptions has this syntax:

```
{
  "status" : "error",
  "reason" : "a human readable string describing the error"
}
```

Common exceptions can be raised in any request indicated in this document. Other specific exception is specified in each request table.

In case of common exception, HTTP status codes is not 200 bad changes as follow:

- Request error (e.g. GET instead of POST, etc.): 405
- User not authenticated: 403
- User has no permission to do that: 402 (abuse of unused HTTP error code)
- General back-end error: 500

Data format

All Date fields are written with string data type. All Date fields must be encoding in ISO8601 compliant format.

Account

| Path: | /account/userinfo |
|-------------------------|--|
| Request: | GET |
| Description: | Return information about current logged user or other user. If <i>id</i> is not specified, this request retrives information for current user. If <i>id</i> is specified, it retrives information about that user. NOTE: for security reason it's not possible to retriive the password field (nor hashed). |
| Parameters: | None for retriive information of current user. Otherwise: <ul style="list-style-type: none"> • id : integer |
| Authentication: | Needed. |
| Response format: | <pre>{ "id" : integer, "firstname" : string, "lastname" : string, "sex" : string ("M" or "F"), "dateOfBirth" : Date, "country" : string, "timezone" : string (IANA format), "email" : string, "languages" : [string, string, ...], "programmingLanguages" : [(only for developers) string, string, ...] "type" : string ("stakeholder" or "developer") }</pre> |
| Exception 1 error code: | 200 – User not exists |

| | |
|---------------------|------------------------------|
| Exception 1 format: | { "status" : "error" } |
|---------------------|------------------------------|

| | |
|--------------------------------|--|
| Path: | /account/register |
| Request: | POST |
| Description: | Register a new user. You need to specify if the user is "stakeholder" or "developer". In the latter case, you need to specify also "programmingLanguages" field. You can get a list of available languages invoking a request on /accout/langs NOTE: user's type cannot be changed after registration. |
| Authentication: | No. ¹ |
| Parameters: | <ul style="list-style-type: none"> • firstname : string • lastname : string • sex : string ("M" or "F") • dateOfBirth : Date • country : string • timezone : string (IANA format) • email : string • password : string (plain-text) • languages : string [] • programmingLanguages : string [] (only for developers) • "type" : string ("stakeholder" or "developer") |
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – E-Mail already exists |
| Exception 1 format: | { "status" : "error", "code" : 1 } |
| Exception 2 error code: | 200 – Wrong field value |
| Exception 2 format: | { "status" : "error", "code" : 2, "field" : string (name of problematic field, e.g. "firstname" or "email") "reason" : " ... " } |

| | |
|-------------------------|---|
| Path: | /account/langs |
| Request: | GET |
| Description: | Returns the list of available programmingLanguages and "spoken" languages. This data is stored in the server and can be changed (with another program) only by administrator. |
| Parameters: | None. |
| Authentication: | No. |
| Response format: | { "languages" : [] } |

¹ It's not allowed creating new account by already authenticated user (in case, server returns a 405 http error).

| | |
|--|---|
| | <pre> string, string, ...], "programmingLanguages" : [string, string, ...], } </pre> |
|--|---|

| Path: /account/edit | |
|-------------------------|---|
| Request: | POST |
| Description: | Edits the current user profile. All parameters are optional, it is possible to specify only fields that will be changed. The old values of this parameters lost after update. |
| Authentication: | Needed. |
| Parameters: | <ul style="list-style-type: none"> • firstname : string • lastname : string • sex : string ("M" or "F") • dateOfBirth : Date • country : string • timezone : string (IANA format) • email : string • password : string (plain-text) • languages : string [] |
| Response format: | <pre> { "status" : "ok" } </pre> |
| Exception 1 error code: | 200 – E-Mail already exists |
| Exception 1 format: | <pre> { "status" : "error", "code" : 1 } </pre> |
| Exception 2 error code: | 200 – Wrong field value |
| Exception 2 format: | <pre> { "status" : "error", "code" : 2, "field" : string (name of problematic field, e.g. "firstname" or "email") "reason" : " ..." } </pre> |

| Path: /account/login | |
|----------------------|--|
| Request: | POST |
| Description: | Login by email and password. The reponse to this request sets cookie/session information (in case of login success). If the user's browser doesn't support cookies, login is not possible. |
| Authentication: | Not needed ² |
| Parameters: | <ul style="list-style-type: none"> • email : string • password : string (plain text) |

² In case the user is already authenticated, this request as no effects.

| | |
|-------------------------|------------------------------|
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – Login failed |
| Exception 1 format: | { "status" : "error" } |

| | |
|------------------|---|
| Path: | /account/logout |
| Request: | GET |
| Description: | Delete the session cookies and log out user. After this call the user should be considered deauthenticated. |
| Parameters: | None. |
| Authentication: | Not needed. ³ |
| Response format: | { "status" : "ok" } |

| | |
|-------------------------|--|
| Path: | /account/recoverpassword |
| Request: | POST |
| Description: | Sends an email to reset the password to the address stored in user profile. In email there is a newly generated random password. |
| Authentication: | No. ⁴ |
| Parameters: | <ul style="list-style-type: none"> email : string dateOfBirth : Date |
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – Wrong fields |
| Exception 1 format: | { "status" : "error" } |

| | |
|------------------|---|
| Path: | /account/listdevelopers |
| Request: | GET |
| Description: | Use this request to retrieve the list of all developers (use /account/userinfo to retrieve more information about a specific user). |
| Parameters: | None. |
| Authentication: | Needed. |
| Response format: | { "developers" : [{ "id" : integer, "firstname" : string, "lastname" : string }, { "id" : integer, "firstname" : string, "lastname" : string },] } |

³ In case the user is not authenticated, this request has no effects.

⁴ Resetting password is not allowed by already authenticated user (in case, server returns a 405 http error).

| | |
|--|---|
| | } |
|--|---|

Projects

| | |
|-------------------------|---|
| Path: | /projects/list |
| Request: | GET |
| Description: | This request returns a list of projects available (possibly empty) visible by current user or visible by a user identified by 'id'. This request can be used by any type of user. |
| Authentication: | Needed. |
| Parameters: | None for retrieve projects of current user. Otherwise: <ul style="list-style-type: none"> id : integer |
| Response format: | <pre>{ "projects" : [{ "id" : integer "name" : string , "description" : string, "manager" : integer, "stakeholder" : integer, "developers" : [integer, integer, ...], "creationDate" : Date }, { ... }] }</pre> |

| | |
|--------------------------------|--|
| Path: | /projects/create |
| Request: | POST |
| Description: | Create a new project. |
| Authentication: | Needed, you must be a <i>stakeholder</i> user. |
| Parameters: | <ul style="list-style-type: none"> name : string description : string manager : integer |
| Response format: | <pre>{ "status" : "created" "id" : integer }</pre> |
| Exception 1 error code: | 200 – Project name already exists |
| Exception 1 format: | <pre>{ "status" : "error" }</pre> |

| | |
|--------------|--|
| Path: | /projects/{idproj}/adddeveloper |
|--------------|--|

| | |
|------------------|---|
| Request: | POST |
| Description: | Add developer to existent project |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> iddev : integer |
| Response format: | <pre>{ "status" : "ok" }</pre> |

| | |
|------------------|---|
| Path: | /projects/{idproj}/remdeveloper |
| Request: | POST |
| Description: | Remove developer from existent project |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> iddev : integer |
| Response format: | <pre>{ "status" : "ok" }</pre> |

| | |
|------------------|--|
| Path: | /projects/{idproj}/edit |
| Request: | POST |
| Description: | Edit a project. All parameters are optional, it is possible to specify only fields that will be changed. |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> name : string description : string manager : int |
| Response format: | <pre>{ "status" : "ok" }</pre> |

Projects - Backlog

| | |
|------------------|--|
| Path: | /projects/{idproj}/getbacklog |
| Request: | GET |
| Description: | Return the backlog (as list of PBIs) of specified project. If the project has no backlog an error is returned. |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | <pre>{ "description" : string, "pbis" : [{ "id" : integer "name" : string , "description" : string, "sprintlog" : integer/null, "type" : string (enum), "priority" : integer, "creationDate" : Date }] }</pre> |

| | |
|-------------------------|---|
| | <pre> }, { ... }] } </pre> |
| Exception 1 error code: | 200 – Backlog not existent |
| Exception 1 format: | <pre> { "backlog" : "no" } </pre> |

| | |
|-------------------------|--|
| Path: | /projects/{idproj}/createbacklog |
| Request: | POST |
| Description: | Creates a backlog for existent project. To add PBIs you need to use /projects/{idproj}/pbis/create request |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> description : string |
| Response format: | <pre> { "status" : "ok" } </pre> |

| | |
|-------------------------|--|
| Path: | /projects/{idproj}/deletebacklog |
| Request: | POST |
| Description: | Delete the backlog of specified project. WARNING: this action deletes also all PBIs, all tasks, and all sprint logs. |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | None. |
| Response format: | <pre> { "status" : "ok" } </pre> |

Projects - PBIs

| | |
|-------------------------|---|
| Path: | /projects/{idproj}/pbis/create |
| Request: | POST |
| Description: | Create a PBI in current project. This PBI will be automatically added to current project's backlog |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> name : string description : string type : string (enum) priority : int |
| Response format: | <pre> { "status" : "ok" } </pre> |
| Exception 1 error code: | 200 – Wrong format of one or more fields |
| Exception 1 format: | <pre> { "status" : "error", </pre> |

| | |
|--|--|
| | <pre> "field" : string (name of problematic field, e.g. "name" or "priority") "reason" : " ..." </pre> |
|--|--|

| | |
|-------------------------|---|
| Path: | /projects/{idproj}/pbis/{idpbis}/edit |
| Request: | POST |
| Description: | Update information of indicated pbis of selected project. All parameters are optional, it is possible to specify only fields that will be changed. |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> • name : string • description : string • type : string (enum) • priority : int |
| Response format: | <pre> { "status" : "ok" } </pre> |

| | |
|-------------------------|---|
| Path: | /projects/{idproj}/pbis/{idpbis}/delete |
| Request: | POST |
| Description: | Delete a PBI from the project |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | None. |
| Response format: | <pre> { "status" : "ok" } </pre> |

Projects - Sprintlog

| | |
|-------------------------|---|
| Path: | /projects/{idproj}/sprintlogs/list |
| Request: | GET |
| Description: | Return the list of sprintlogs of current project |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | <pre> { "sprintlogs" : [{ "id" : integer "name" : string , "description" : string, "startDate" : Date, "endDate" : Date }, {...}, ...] } </pre> |

| | |
|-----------------|---|
| Path: | /projects/{idproj}/sprintlogs/create |
| Request: | POST |

| | |
|-------------------------|--|
| Description: | Create a new sprintlogs with list of pbis (you need to have created pbis before) |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> name : string description : string startDate : Date endDate: Date idpbis : integer[] |
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – Wrong format of one or more fields |
| Exception 1 format: | { "status" : "error", "code" : 1, "field" : string (name of problematic field, e.g. "name" or "priority") "reason" : " ... " } |
| Exception 2 error code: | 200 – Sprintlog name already exists |
| Exception 2 format: | { "status" : "error", "code" : 2 } |

| | |
|------------------|---|
| Path: | /projects/{idproj}/sprintlogs/{idsprintlog}/listpbis |
| Request: | GET |
| Description: | Return the list of PBIs associated to sprintlogs of current project |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | { "pbis" : [integer, integer, integer , ...] TBD: all fields of pbis or only ids like this? } |

| | |
|-------------------------|--|
| Path: | /projects/{idproj}/sprintlogs/{idsprintlog}/addpbi |
| Request: | POST |
| Description: | Add already existent pbi to the sprintlog |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> id : integer |
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – PBI does not exists or already assigned |
| Exception 1 format: | { "status" : "error" } |

| | |
|--------------|--|
| Path: | /projects/{idproj}/sprintlogs/{idsprintlog}/removepbi |
|--------------|--|

| | |
|-------------------------|---|
| Request: | POST |
| Description: | Remove pbi from sprintlog (this method doesn't delete the pbi from project) |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> id : integer |
| Response format: | <pre>{ "status" : "ok" }</pre> |
| Exception 1 error code: | 200 – PBI does not exists or assigned to other sprintlog |
| Exception 1 format: | <pre>{ "status" : "error" }</pre> |

| | |
|------------------|---|
| Path: | /projects/{idproj}/sprintlogs/{idsprintlog}/delete |
| Request: | POST |
| Description: | Remove sprintlog from project. (NOTE: PBIs assigned are not delete, they remain in backlog) |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | None. |
| Response format: | <pre>{ "status" : "ok" }</pre> |

Projects - Tasks

| | |
|------------------|--|
| Path: | /projects/{idproj}/pbis/{idpbis}/listtasks |
| Request: | GET |
| Description: | Return the list of tasks associated to PBI indicated |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | <pre>{ "tasks" : [{ "id" : integer, "name" : string, "description" : string, "points" : integer, "developer" : integer, "state" : string (enum), "pbi" : integer }, ...] }</pre> |

| | |
|--------------|--|
| Path: | /projects/{idproj}/pbis/{idpbis}/createtask |
| Request: | POST |

| | |
|------------------|--|
| Description: | Add a task to indicated PBI. |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> • name : string • description : string • points : integer • developer : integer • state : string (enum) • pbi : integer |
| Response format: | { "status" : "ok" } |

| | |
|-------------------------|---|
| Path: | /projects/{idproj}/pbis/{idpbis}/tasks/{idtask}/delete |
| Request: | POST |
| Description: | Remove task from indicated PBI. |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | None. |
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – Task does not exists or assigned to other PBI |
| Exception 1 format: | { "status" : "error" } |

| | |
|-------------------------|--|
| Path: | /projects/{idproj}/pbis/{idpbis}/tasks/{idtask}/edit |
| Request: | POST |
| Description: | Edit task indicated. All parameters are optional, it is possible to specify only fields that will be changed. |
| Authentication: | Needed, you must be a <i>manager</i> of that project. |
| Parameters: | <ul style="list-style-type: none"> • name : string • description : string • points : integer • developer : integer • state : string (enum) • pbi : integer |
| Response format: | { "status" : "ok" } |
| Exception 1 error code: | 200 – Task does not exists or assigned to other PBI |
| Exception 1 format: | { "status" : "error" } |

Statistics

| | |
|-------------------------|---|
| Path: | /stats/devpoints |
| Request: | GET |
| Description: | Returns the list of developers-points calculated by sum all points earned in each project. |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | <pre>{ "stats" : [{ "dev" : integer, "points" : integer }, { "dev" : integer, "points" : integer }, { "dev" : integer, "points" : integer }, ...] }</pre> |

Chat system

| | |
|-------------------------|---|
| Path: | /chat/{idproj}/list |
| Request: | GET |
| Description: | Returns the list of available chat of specified project for current user. The response (possibly empty) contains an identifier of the chat and a list of users participating to that chat (not including current developer). 1-to-1 chat contains only 1 item in users array. |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | <pre>{ "chats" : [{ "id" : integer, "users" : [integer, integer, ...] }, { "id" : integer, "users" : [integer, integer, ...] }, { "id" : integer, "users" : [integer, integer, ...] }, ...] }</pre> |

| | |
|--------------------------------|---|
| Path: | /chat/{idproj}/create |
| Request: | POST |
| Description: | Creates a new chat into indicated project. |
| Authentication: | Needed. |
| Parameters: | <ul style="list-style-type: none"> users : [integer, integer,] |
| Response format: | <pre>{ "status" : "ok" }</pre> |
| Exception 1 error code: | 200 – Same chat already exists |
| Exception 1 format: | <pre>{ "status" : "error" }</pre> |

| Path: | /chat/{idproj}/msg/{idchat}/list |
|------------------|--|
| Request: | GET |
| Description: | This request returns list of last 20 messages (or less) of the chat. This should be used when user open the chat to retrieve the recent history. |
| Authentication: | Needed. |
| Parameters: | None |
| Response format: | { "messages" : [{ "id" : integer, "user" : integer, "text" : string, "date" : Date }, { "id" : integer, "user" : integer, "text" : string, "date" : Date },], { "id" : integer, "user" : integer, "text" : string, "date" : Date },], ...] } |

| Path: | /chat/{idproj}/msg/{idchat}/polling |
|------------------|--|
| Request: | GET |
| Description: | This request returns list of last messages from id indicated in parameters. You should call this method every time you want to check if there are new messages, passing the id of last message received. |
| Authentication: | Needed. |
| Parameters: | <ul style="list-style-type: none">lastid : integer |
| Response format: | { "messages" : [{ "id" : integer, "user" : integer, "text" : string, "date" : Date }, { "id" : integer, "user" : integer, "text" : string, "date" : Date },], { "id" : integer, "user" : integer, "text" : string, "date" : Date },], ...] } |

| Path: | /chat/{idproj}/msg/{idchat}/send |
|------------------|--|
| Request: | POST |
| Description: | Sends a new message to the chat. |
| Authentication: | Needed. |
| Parameters: | <ul style="list-style-type: none">message : string |
| Response format: | { "status" : "ok" } |