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 retrive project list of other user.	Federico Reghenzani
25.11.14	0.3.2	Added an exception to /account/userinfo	Federico Reghenzani
02.12.14	0.3.3	Added change programming languages to /account/edit, added dueDate to PBI	Federico Reghenzani, Mikel Vuka, Filippo Pagano

Table of Contents

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 progammingLanguages.
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 *id* is not specified, this request retrives information for current user. If *id* is specified, it retrives information about that user. NOTE: for security reason it's not possible to retrieve the password field (nor hashed).
Parameters:	None for retrive information of current user. Otherwise: • id : integer
Authentication:	Needed.

Response format:	{
	  "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")
	}

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:	• 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).

```
        string,
        string,
        ...
    ],
    "programmingLanguages" : [
        string,
        string,
        ...
    ],
}
```

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:	• 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)

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/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:	• 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:	• 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 retrive the list of all developers (use /account/userinfo to retrive 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 as 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 visibile by a user identified by 'id'. This request can be used by any type of user.
Authentication:	Needed.
Parameters:	None for retrive projects of current user. Otherwise: • id : integer

Response format:	{
	   "projects" : [
	     {
	       "id" : integer
	       "name" : string ,
	       "description" : string,
	       "manager" : integer,
	       "stakeholder" : integer,
	       "developers" : [
	         integer,
	         integer,
	         ...
	       ],
	       "creationDate" : Date
	     },
	     { ... }
	   ]
	}

Path:	/projects/create
Request:	POST
Description:	Create a new project.
Authentication:	Needed, you must be a *stakeholder* user.
Parameters:	• name : string • description : string • manager : integer

Response format:	{
	   "status" : "created"
	   "id" : integer
	}

Exception 1 error code:	200 – Project name already exists

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

Path:	/projects/{idproj}/adddeveloper

Request:	POST
Description:	Add developer to existent project
Authentication:	Needed, you must be a *manager* of that project.
Parameters:	• iddev : integer

Response format:	{
	   "status" : "ok"
	 }

Path:	/projects/{idproj}/remdeveloper
Request:	POST
Description:	Remove developer from existent project
Authentication:	Needed, you must be a *manager* of that project.
Parameters:	• iddev : integer

Response format:	{
	   "status" : "ok"
	 }

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 *manager* of that project.
Parameters:	• name : string • description : string • manager : int

Response format:	{
	   "status" : "ok"
	 }

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:	{
	   "description" : string,
	   "pbis" : [
	     {
	       "id" : integer
	       "name" : string ,
	       "description" : string,
	       "sprintlog" : integer/null,
	       "type" : string (enum),
	       "priority" : integer,
	       "creationDate" : Date

```
      },
      { … }
    ]
}
```

Exception 1 error code:	200 – Backlog not existent

Exception 1 format:	{
	    "backlog" : "no"
	}

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 *manager* of that project.
Parameters:	• description : string

Response format:	{
	    "status" : "ok"
	}

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 *manager* of that project.
Parameters:	None.

Response format:	{
	    "status" : "ok"
	}

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 *manager* of that project.
Parameters:	• name : string • description : string • type : string (enum) • priority : int • dueDate : Date

Response format:	{
	    "status" : "ok"
	}

Exception 1 error code:	200 – Wrong format of one or more fields
Exception 1 format:	{

```
                    "status" : "error",
                    "field" : string (name of problematic field, e.g. "name" or "priority")
                    "reason" : " ... "
                }
```

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 *manager* of that project.
Parameters:	• name : string • description : string • type : string (enum) • priority : int

Response format:	{
	    "status" : "ok"
	}

Path:	/projects/{idproj}/pbis/{idpbis}/delete
Request:	POST
Description:	Delete a PBI from the project
Authentication:	Needed, you must be a *manager* of that project.
Parameters:	None.

Response format:	{
	    "status" : "ok"
	}

Projects - Sprintlog

Path:	/projects/{idproj}/sprintlogs/list
Request:	GET
Description:	Return the list of sprintlogs of current project
Authentication:	Needed.
Parameters:	None

Response format:	{
	    "sprintlogs" : [
	        {
	            "id" : integer
	            "name" : string ,
	            "description" : string,
	            "startDate" : Date,
	            "endDate" : Date
	        },
	        {…}, ...
	        ]
	}

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 *manager* of that project.
Parameters:	• 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 *manager* of that project.
Parameters:	• 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 *manager* of that project.
Parameters:	• id : integer

Response format:	{
	    "status" : "ok"
	}

Exception 1 error code:	200 – PBI does not exists or assigned to other sprintlog

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

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 *manager* of that project.
Parameters:	None.

Response format:	{
	    "status" : "ok"
	}

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:	{
	    "tasks" : [
	        {
	            "id" : integer,
	            "name" : string,
	            "description" : string,
	            "points" : integer,
	            "developer" : integer,
	            "state" : string (enum),
	            "pbi" : integer
	        },
	        ...
	    ]
	}

Path:	/projects/{idproj}/pbis/{idpbis}/createtask

Request:	POST
Description:	Add a task to indicated PBI.
Authentication:	Needed, you must be a *manager* of that project.
Parameters:	• 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 *manager* 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 *manager* of that project.
Parameters:	• 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:	{
	    "stats" :
	    [
	    { "dev" : integer, "points" : integer },
	    { "dev" : integer, "points" : integer },
	    { "dev" : integer, "points" : integer },
	    ...
	    ]
	}

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 partecipating to that chat (not including current developer). 1-to-1 chat contains only 1 item in users array.
Authentication:	Needed.
Parameters:	None

Response format:	{
	    "chats" :
	    [
	    { "id" : integer, "users" : [integer, integer, ...] },
	    { "id" : integer, "users" : [integer, integer, ...] },
	    { "id" : integer, "users" : [integer, integer, ...] },
	    ...
	    ]
	}

Path:	/chat/{idproj}/create
Request:	POST
Description:	Creates a new chat into indicated project.
Authentication:	Needed.
Parameters:	• users : [integer, integer, ….]

Response format:	{
	    "status" : "ok"
	}

Exception 1 error code:	200 – Same chat already exists
Exception 1 format:	{

	"status" : "error" }

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 retrive 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:	• 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:	• message : string

Response format:	{
	   "status" : "ok"
	}