

# 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
02.12.14	0.3.3	Added change programming languages to /account/edit, added dueDate to PBI	Federico Reghenzani, Mikel Vuka, Filippo Pagano
09.12.14	0.3.4	Edit tasks URL and removed pbi from parameters (is already in the url)	Federico Reghenzani
10.12.14	0.4	Updated chat message's format	Mikel Vuka
11.12.14	0.4.1	Behaviour specification of sprintlogs removepbi	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.
GET	/account/delete	Delete the current user account.
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/{idpbi}/edit	Edit a PBI.
POST	/projects/{idproj}/pbis/{idpbi}/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/{idpbi}/listtasks	Lists all tasks of a PBI
POST	/projects/{idproj}/pbis/{idpbi}/createtask	Creates a new task to PBI
POST	/projects/{idproj}/pbis/{idpbi}/tasks/{idtask}/edit	Edits task of a PBI
POST	/projects/{idproj}/pbis/{idpbi}/tasks/{idtask}/delete	Deletes task of a PBI
GET	/stats/devpoints	Retrives points of developers.
GET	/chat/{idproj}/getchat	Returns the list of available chat of specified project
POST	/chat/{idproj}/insertmessage	Creates a new chat into indicated project
GET	/chat/{idproj}/adduser	Returns the last 20 messages of the chat
POST	/chat/{idproj}/deleteuser	Sends a new message
GET	/chat/{idproj}/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

The requests specified in this document that have no "dedicated" exceptions, can however provide a 200 error with format specified previously.

## 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 retrieves information for current user. If <i>id</i> is specified, it retrieves information about that user. NOTE: for security reason it's not possible to retrieve the password field (nor hashed).
Parameters:	None for retrieve information of current user. Otherwise: <ul style="list-style-type: none"> <li>• <i>id</i> : integer</li> </ul>
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" }

<b>Path:</b>	<b>/account/register</b>
<b>Request:</b>	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. <sup>1</sup>
Parameters:	<ul style="list-style-type: none"> <li>• firstname : string</li> <li>• lastname : string</li> <li>• sex : string ("M" or "F")</li> <li>• dateOfBirth : Date</li> <li>• country : string</li> <li>• timezone : string (IANA format)</li> <li>• email : string</li> <li>• password : string (plain-text)</li> <li>• languages : string [ ]</li> <li>• programmingLanguages : string [ ] (only for developers)</li> <li>• "type" : string ("stakeholder" or "developer")</li> </ul>
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" : " ... " }

<b>Path:</b>	<b>/account/langs</b>
<b>Request:</b>	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.

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

Authentication:	No.
Response format:	<pre>{   "languages" : [     string,     string,     ...   ],   "programmingLanguages" : [     string,     string,     ...   ], }</pre>

<b>Path:</b>	<b>/account/edit</b>
<b>Request:</b>	<b>POST</b>
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"> <li>• firstname : string</li> <li>• lastname : string</li> <li>• sex : string ("M" or "F")</li> <li>• dateOfBirth : Date</li> <li>• country : string</li> <li>• timezone : string (IANA format)</li> <li>• email : string</li> <li>• password : string (plain-text)</li> <li>• languages : string [ ]</li> <li>• programmingLanguages : string [ ] (only for developers)</li> </ul>
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>

<b>Path:</b>	<b>/account/delete</b>
<b>Request:</b>	<b>GET</b>
Description:	Remove the current user account. Pay attention: this action cannot be undone! After call this rest api, the user is automatically logged out (also deleting cookies, like a normal logout call)



Authentication:	Needed.
Parameters:	None
Response format:	{ "status" : "ok" }

<b>Path:</b>	<b>/account/login</b>
<b>Request:</b>	POST
Description:	Login by email and password. The response 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 <sup>2</sup>
Parameters:	<ul style="list-style-type: none"> <li>email : string</li> <li>password : string (plain text)</li> </ul>
Response format:	{ "status" : "ok" }
Exception 1 error code:	200 – Login failed
Exception 1 format:	{ "status" : "error" }

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

<b>Path:</b>	<b>/account/recoverpassword</b>
<b>Request:</b>	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. <sup>4</sup>
Parameters:	<ul style="list-style-type: none"> <li>email : string</li> <li>dateOfBirth : Date</li> </ul>
Response format:	{ "status" : "ok" }
Exception 1 error code:	200 – Wrong fields

<sup>2</sup> In case the user is already authenticated, this request has no effects.

<sup>3</sup> In case the user is not authenticated, this request has no effects.

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

Exception 1 format: {  
     "status" : "error",  
     "reason" : "..."  
 }

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

## Projects

<b>Path:</b>	<b>/projects/list</b>
<b>Request:</b>	GET
<b>Description:</b>	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.
<b>Authentication:</b>	Needed.
<b>Parameters:</b>	None for retrieve projects of current user. Otherwise: <ul style="list-style-type: none"> <li>id : integer</li> </ul>
<b>Response format:</b>	{ "projects" : [ { "id" : integer "name" : string , "description" : string, "manager" : integer, "stakeholder" : integer, "creationDate" : Date "developers" : [ integer, integer, ... ], }, { ... } ] }

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

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

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

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

Path:	/projects/{idproj}/deletebacklog
Request:	POST
Description:	Delete the backlog of specified project. <b>WARNING:</b> 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

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

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

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

## Projects - Sprintlog

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

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

<b>Path:</b>	<b>/projects/{idproj}/sprintlogs/{idsprintlog}/listpbis</b>
<b>Request:</b>	GET
<b>Description:</b>	Return the list of PBIs associated to sprintlogs of current project
<b>Authentication:</b>	Needed.
<b>Parameters:</b>	None
<b>Response format:</b>	<pre>{   "pbis" : [ integer, integer, integer , ... ] }</pre>

<b>Path:</b>	<b>/projects/{idproj}/sprintlogs/{idsprintlog}/addpbi</b>
--------------	---

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"> <li>id : integer</li> </ul>
Response format:	<pre>{   "status" : "ok" }</pre>
Exception 1 error code:	200 – PBI does not exists or already assigned
Exception 1 format:	<pre>{   "status" : "error" }</pre>

<b>Path:</b>	<b>/projects/{idproj}/sprintlogs/{idsprintlog}/removepbi</b>
Request:	POST
Description:	Remove pbi from sprintlog (this method doesn't delete the pbi from project). Due to simplification of implementation, if the PBI does not exist, no action will be taken, and a positive answer will be sent to the client.
Authentication:	Needed, you must be a <i>manager</i> of that project.
Parameters:	<ul style="list-style-type: none"> <li>id : integer</li> </ul>
Response format:	<pre>{   "status" : "ok" }</pre>

<b>Path:</b>	<b>/projects/{idproj}/sprintlogs/{idsprintlog}/delete</b>
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

<b>Path:</b>	<b>/projects/{idproj}/pbis/{idpbi}/listtasks</b>
Request:	GET
Description:	Return the list of tasks associated to PBI indicated
Authentication:	Needed.
Parameters:	None
Response format:	<pre>{   "tasks" : [     {       "id" : integer,       "name" : string, </pre>

```

        "description" : string,
        "points" : integer,
        "developer" : integer,
        "pbi" : integer,
        "state" : string (enum),
    },
    ...
]
}

```

<b>Path:</b>	<b>/projects/{idproj}/pbis/{idpbi}/createtask</b>
<b>Request:</b>	POST
<b>Description:</b>	Add a task to indicated PBI.
<b>Authentication:</b>	Needed, you must be a <i>manager</i> of that project.
<b>Parameters:</b>	<ul style="list-style-type: none"> <li>• name : string</li> <li>• description : string</li> <li>• points : integer</li> <li>• developer : integer</li> <li>• state : string (enum)</li> </ul>
<b>Response format:</b>	<pre>{   "status" : "ok" }</pre>
<b>Exception 1 error code:</b>	200 – Wrong field value
<b>Exception 1 format:</b>	<pre>{   "status" : "error",   "field" : string (name of problematic field, e.g. "name" or "description")   "reason" : " ..." }</pre>

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

<b>Path:</b>	<b>/projects/{idproj}/tasks/{idtask}/edit</b>
<b>Request:</b>	POST
<b>Description:</b>	Edit task indicated. All parameters are optional, it is possible to specify only fields that will be changed.
<b>Authentication:</b>	Needed, you must be a <i>manager</i> of that project.
<b>Parameters:</b>	<ul style="list-style-type: none"> <li>• name : string</li> </ul>



	<ul style="list-style-type: none"> <li>• description : string</li> <li>• points : integer</li> <li>• developer : integer</li> <li>• state : string (enum)</li> </ul>
Response format:	{ "status" : "ok" }
Exception 1 error code:	200 – Task does not exists or assigned to other PBI
Exception 1 format:	{ "status" : "error" }

## Statistics

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

<b>Path:</b>	<b>/stats/projpoints</b>
<b>Request:</b>	GET
<b>Description:</b>	Returns the list of developer-points (developer specified by parameter) calculated by sum all points earned in specified project.
<b>Authentication:</b>	Needed.
<b>Parameters:</b>	<ul style="list-style-type: none"> <li>• idproj : integer</li> <li>• iddev : integer</li> </ul>
<b>Response format:</b>	{ "points" : integer }

## Chat system

<b>Path:</b>	<b>/chat/{idproj}/getchat</b>
<b>Request:</b>	GET
<b>Description:</b>	Returns the chat of the specified project.

Authentication:	Needed.
Parameters:	None
Response format:	<pre>{   "id" : integer   "project": integer   "users": [     {"id": integer, "firstname": string, "lastname": string}     ...   ]   "messages": [     {"id": integer, "user": string, "date": integer, "chat": integer, "content": string}     ...   ] }</pre>

<b>Path:</b>	<b>/chat/{idproj}/adduser</b>
Request:	POST
Description:	This request adds a user in the chat.
Authentication:	Needed, you must be manager of the project.
Parameters:	<ul style="list-style-type: none"> <li>• user: integer</li> </ul>
Response format:	"status": "ok"

<b>Path:</b>	<b>/chat/{idproj}/deleteuser</b>
Request:	POST
Description:	This request deletes a user from the chat.
Authentication:	Needed, you must be manager of the project.
Parameters:	<ul style="list-style-type: none"> <li>• user: integer</li> </ul>
Response format:	"status": "ok"

<b>Path:</b>	<b>/chat/{idproj}/polling</b>
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"> <li>• lastmessageid : integer</li> <li>• chatid: integer</li> </ul>
Response format:	<pre>{   "messages" :   [     { "id" : integer, "user" : integer, "date": integer, "chat": integer, "content": string }, </pre>

```
{ "id" : integer, "user" : integer, "date": integer, "chat": integer, "content": string },  
{ "id" : integer, "user" : integer, "date": integer, "chat": integer, "content": string },  
...  
]  
}
```

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