API Documentation

All methods require authentication and should be done by using the header with the following: “**Key**: *Authorization”* and “**Value**: *dXNlcg==”* .

Due to some database issues, only Projects supports the basic functions, in the next releases all of these issues will be fixed. A working version can be found in the following URL http://51.75.163.237:8080/ .

Project API

GET

/v1/projects/

Returns a list containing the representation of all the projects.

Representation of a project:

Name	**Type**	**Description**
Id	Int	The number of project.
Name	String	Name of the project.
short_desc	String	Short description of the project.

POST

/v1/projects/

Creates a new project.

Parameters for ProjectM

Name	**Type**	**Description**
Id	Int	The number of project.
Name	String	Name of the project.
short_desc	String	Short description of the project.

PUT

/v1/projects/:project_id

Updates an existing project. The id of the respective project must exist on the server.

Parameters for ProjectIM

Name	**Type**	**Description**
Id	Int	The number of project.
Name	String	Name of the project.
short_desc	String	Short description of the project.

DELETE

/v1/projects/:project_id

Deletes the project with the specified project_id.

Name	**Type**	**Description**
Id	Int	The id of the project to be deleted.

Available States API

GET
/v1/projects/:project_id/states/
Returns a list with the representation of all available states that belong to the project containing the project_id.

Representation of a state:

Name	**Type**	**Description**
Id	Int	The id of the State
Name	String	Name of the state.
projectID	Int	The project id.

PUT
/v1/projects/:project_id/states/
Updates the array of states that belong to the project containing the project_id.

Parameters

Name	**Type**	**Description**
states	Array of Strings	Required. An array of states to add to the project. Pass one or more states to *replace* the set of existing states. Send an empty array ([]) to clear all states from the project.

Labels API

GET
/v1/projects/:project_id/labels/
Returns a list with the representation of all available labels that belong to the project containing the project_id.

Name	Type	Description
Id	Int	The id of the Label.
text	String	Text of the label.

PUT
/v1/projects/:project_id/labels/
Updates the array of labels that belong to the project containing the project_id.

Parameters

Name	Type	Description
labels	Array of Strings	Required. An array of labels to add to the project. Pass one or more labels to *replace* the set of existing labels. Send an empty array ([]) to clear all labels from the project.

Allowed Labels API

GET
v1/projects/:project_id/labels
Returns a list with the representation of all allowed labels that belong to the project containing the project_id.

Representation of allowed labels

Name	**Type**	**Description**
Project_id	Int	The id of the project.
Label_id	[]String	An array that contains the id's of the labels.

PUT
v1/projects/:project_id/labels
The labels are updated every time that a PUT is done, replacing the previous information.

Parameters of allowed labels

Name	**Type**	**Description**
Project_id	Int	The id of the project.
Label_id	[]String	An array that contains the id's of the labels.

Transitions API

GET

/v1/projects/:project_id/transitions/

Returns a list with the representation of all available state transitions that belong to the project containing the project_id.

Representation of a transition:

Name	Type	Description
projectID	Int	The id of the project.
initialStateID	Int	Initial state of the project.
targetStateID	Int	Target state of the project.

POST

/v1/projects/:project_id/transitions/

Creates a new transition that belongs to the project containing the project_id.

Parameters of TransitionIM

Name	Type	Description
projectID	Int	The id of the project.
initialStateID	Int	Initial state of the project.
targetStateID	Int	Target state of the project.

PUT

/v1/projects/:project_id/transitions/:transition_id

Updates an existing transition that belongs to the project containing the project_id.

Parameters of TransitionIM

Name	Type	Description
projectID	Int	The id of the project.
initialStateID	Int	Initial state of the project.
targetStateID	Int	Target state of the project.

DELETE

/v1/projects/:project_id/transitions/:transition_id

Deletes the transition with the specified transition_id that belongs to the project containing the project_id.

Issues API

GET

/v1/projects/:project_id/issues/

Returns a list with the representation of all available issues that belong to the project containing the project_id.

Representation of a issue:

Name	**Type**	**Description**
Id	Int	The id of the issue.
Name	String	Name of the issue.
Short_desc	String	A short description of the issue
creationDate	Date	The date on which the issue was created.
closeDate	Date	On what date the Issue was closed.
stated	Int	The state of the issue.
labels	[]String	An array with the labels id.
projectID	Int	The id of the project.

POST

/v1/projects/:project_id/issues/

Creates a new issue that belongs to the project containing the project_id.

Parameters of IssueIM

Name	**Type**	**Description**
Id	Int	The id of the issue.
Name	String	Name of the issue.
Short_desc	String	A short description of the issue
labels	[]String	An array with the labels id.
projectID	Int	The id of the project.

PUT

/v1/projects/:project_id/issues/:issues_id

Updates an issue that belongs to the project containing the project_id. The respective issue must exist on the server.

Parameters

Name	**Type**	**Description**
Id	Int	The id of the issue.
Name	String	Name of the issue.
Short_desc	String	A short description of the issue
projectID	Int	The id of the project.

DELETE
/v1/projects/:project_id/issues/:issue_id
Deletes the issue with the specified issue_id that belong to the project containing the project_id.

Parameters

Name	**Type**	**Description**
Id	Int	The id of the issue to be deleted.

Comments API

GET

/v1/projects/:project_id/issues/:issue_id/comments

Returns a list with the representation of all available comments that belong to the issue containing the issue_id.

Representation of a comment:

Name	Type	Description
Id	Int	The id of the comment.
text	String	Text of the comment.
issueID	Int	The id of issue.

POST

/v1/projects/:project_id/issues/:issue_id/comments

Creates a new comment that belong to the issue containing the issue_id.

Parameters of CommentsIM

Name	Type	Description
Id	Int	The id of the comment.
text	String	Text of the comment.
issueID	Int	The id of issue.

PUT

/v1/projects/:project_id/issues/:issues_id/comments/:comment_id

Updates a a existing issue that belong to the issue containing the issue_id.

Parameters

Name	Type	Description
Id	Int	The id of the comment.
text	String	Text of the comment.
issueID	Int	The id of issue.

DELETE

/v1/projects/:project_id/issues/:issues_id/comments/:comment_id

Deletes the comment with the specified comment_id that belong to the issue containing the issue_id.

Parameters

Name	Type	Description
comment_id	Int	The id of the comment to be deleted.