Taiga REST API

Table of Contents

1. General notes . . . 13
   1.1. Authentication . . . 13
   1.2. OCC - Optimistic concurrency control . . . 18
   1.3. Pagination . . . 18
   1.4. Internationalization . . . 19
   1.5. Throttling . . . 19
   1.6. Read only fields . . . 19
2. Endpoints Summary . . . 19
   2.1. Auth . . . 19
   2.2. Applications . . . 19
   2.3. Application Tokens . . . 20
   2.4. Resolver . . . 20
   2.5. Searches . . . 20
   2.6. User storage . . . 20
   2.7. Project templates . . . 20
   2.8. Projects . . . 21
   2.9. Memberships/Invitations . . . 22
   2.10. Roles . . . 23
   2.11. Milestones . . . 23
   2.12. Epics . . . 24
   2.13. Epic status . . . 25
   2.14. Epic custom attribute . . . 26
   2.15. Epic custom attributes values . . . 26
   2.16. User stories . . . 26
   2.17. User story status . . . 28
   2.18. Points . . . 28
   2.19. User story custom attribute . . . 28
   2.20. User story custom attributes values . . . 29
   2.21. Tasks . . . 29
   2.22. Task status . . . 30
   2.23. Task custom attribute . . . 31
   2.24. Task custom attributes values . . . 31
   2.25. Issues . . . 31
   2.26. Issue status . . . 32
   2.27. Issue types . . . 33
   2.28. Priorities . . . 33

2.29. Severities . . . 33
2.30. Issue custom attribute . . . 34
2.31. Issue custom attributes values . . . 34
2.32. Wiki pages . . . 34
2.33. Wiki links . . . 35
2.34. History . . . 35
2.35. Users . . . 37
2.36. Notify policies . . . 37
2.37. Contact . . . 38
2.38. Feedback . . . 38
2.39. Export/Import . . . 38
2.40. Webhooks . . . 38
2.41. Timelines . . . 39
2.42. Locales . . . 39
2.43. Stats . . . 39
2.44. Importers . . . 39
3. Auth . . . 40
3.1. Normal login . . . 40
3.2. Github login . . . 40
3.3. Public registry . . . 41
3.4. Private registry . . . 42
4. Applications . . . 42
4.1. Get . . . 42
4.2. Get token . . . 43
5. Application tokens . . . 43
5.1. List . . . 43
5.2. Get . . . 43
5.3. Delete . . . 43
5.4. Authorize . . . 44
5.5. Validate . . . 44
6. Resolver . . . 45
6.1. Projects . . . 45
6.2. User stories . . . 45
6.3. Issues . . . 46
6.4. Tasks . . . 46
6.5. Milestones . . . 47
6.6. Wiki pages . . . 47
6.7. Multiple resolution . . . 48
6.8. By ref value . . . 48
7. Searches . . . 48
7.1. Search . . . 49

8. User storage . . . 49
8.1. List . . . 49
8.2. Create . . . 49
8.3. Get . . . 50
8.4. Edit . . . 50
8.5. Delete . . . 50
9. Project templates . . . 50
9.1. List . . . 51
9.2. Create . . . 51
9.3. Get . . . 62
9.4. Edit . . . 62
9.5. Delete . . . 63
10. Projects . . . 63
10.1. List . . . 63
10.2. Create . . . 64
10.3. Get . . . 65
10.4. Get by slug . . . 65
10.5. Edit . . . 66
10.6. Delete . . . 66
10.7. Bulk update order . . . 67
10.8. Get modules configuration . . . 67
10.9. Edit modules configuration . . . 67
10.10. Stats . . . 68
10.11. Issue stats . . . 68
10.12. Tag colors . . . 68
10.13. Create tag . . . 69
10.14. Edit tag . . . 69
10.15. Delete-tag . . . 69
10.16. Mix tags . . . 70
10.17. Like a project . . . 70
10.18. Unlike a project . . . 70
10.19. List project fans . . . 71
10.20. Watch a project . . . 71
10.21. Stop watching project . . . 71
10.22. List project watchers . . . 71
10.23. Create template . . . 72
10.24. Leave . . . 72
10.25. Change logo . . . 72
10.26. Remove logo . . . 73
10.27. Transfer validate-token . . . 73
10.28. Transfer request . . . 73

10.29. Transfer start . . . 73
10.30. Transfer accept . . . 74
10.31. Transfer reject . . . 74
10.32. Duplicate . . . 75
11. Memberships/Invitations . . . 75
11.1. List . . . 75
11.2. Create . . . 76
11.3. Bulk creation . . . 76
11.4. Get . . . 77
11.5. Edit . . . 77
11.6. Delete . . . 78
11.7. Resend invitation . . . 78
11.8. Get Invitation (by token) . . . 78
12. Roles . . . 78
12.1. List . . . 78
12.2. Create . . . 79
12.3. Get . . . 80
12.4. Edit . . . 80
12.5. Delete . . . 80
13. Milestones . . . 81
13.1. List . . . 81
13.2. Create . . . 81
13.3. Get . . . 82
13.4. Edit . . . 83
13.5. Delete . . . 83
13.6. Stats . . . 83
13.7. Watch a milestone . . . 84
13.8. Stop watching a milestone . . . 84
13.9. List milestone watchers . . . 84
14. Epics . . . 84
14.1. List . . . 84
14.2. Create . . . 85
14.3. Get . . . 86
14.4. Get by ref . . . 87
14.5. Edit . . . 87
14.6. Delete . . . 87
14.7. Bulk creation . . . 88
14.8. Filters data . . . 88
14.9. List related userstories . . . 88
14.10. Create related userstory . . . 89
14.11. Get related userstory . . . 89

14.12. Edit related userstory . . . 89
14.13. Delete related userstory . . . 90
14.14. Bulk related userstories creation . . . 90
14.15. Vote an epic . . . 90
14.16. Remove vote from an epic . . . 91
14.17. Get epic voters list . . . 91
14.18. Watch an epic . . . 91
14.19. Stop watching an epic . . . 91
14.20. List epic watchers . . . 92
14.21. List attachments . . . 92
14.22. Create attachment . . . 92
14.23. Get attachment . . . 93
14.24. Edit attachment . . . 93
14.25. Delete attachment . . . 93
15. Epic status . . . 93
15.1. List . . . 94
15.2. Create . . . 94
15.3. Get . . . 95
15.4. Edit . . . 95
15.5. Delete . . . 95
15.6. Bulk update order . . . 96
16. Epic custom attribute . . . 96
16.1. List . . . 96
16.2. Create . . . 97
16.3. Get . . . 98
16.4. Edit . . . 98
16.5. Delete . . . 98
16.6. Bulk update order . . . 99
17. Epic custom attributes values . . . 99
17.1. Get . . . 99
17.2. Edit . . . 100
18. User stories . . . 100
18.1. List . . . 100
18.2. Create . . . 101
18.3. Get . . . 103
18.4. Get by ref . . . 103
18.5. Edit . . . 103
18.6. Delete . . . 104
18.7. Bulk creation . . . 104
18.8. Bulk update backlog order . . . 105
18.9. Bulk update kanban order . . . 105

18.10. Bulk update sprint order . . . 106
18.11. Bulk update milestone . . . 107
18.12. Filters data . . . 107
18.13. Vote a user story . . . 108
18.14. Remove vote from a user story . . . 108
18.15. Get user story voters list . . . 108
18.16. Watch a user story . . . 108
18.17. Stop watching a user story . . . 109
18.18. List user story watchers . . . 109
18.19. List attachments . . . 109
18.20. Create attachment . . . 109
18.21. Get attachment . . . 110
18.22. Edit attachment . . . 110
18.23. Delete attachment . . . 111
19. User story status . . . 111
19.1. List . . . 111
19.2. Create . . . 111
19.3. Get . . . 112
19.4. Edit . . . 112
19.5. Delete . . . 113
19.6. Bulk update order . . . 113
20. Points . . . 114
20.1. List . . . 114
20.2. Create . . . 115
20.3. Get . . . 115
20.4. Edit . . . 116
20.5. Delete . . . 116
20.6. Bulk update order . . . 116
21. User story custom attribute . . . 117
21.1. List . . . 117
21.2. Create . . . 118
21.3. Get . . . 118
21.4. Edit . . . 119
21.5. Delete . . . 119
21.6. Bulk update order . . . 119
22. User story custom attributes values . . . 120
22.1. Get . . . 120
22.2. Edit . . . 120
23. Tasks . . . 121
23.1. List . . . 121
23.2. Create . . . 122

23.3. Get . . . 123
23.4. Get by ref . . . 124
23.5. Edit . . . 126
23.6. Delete . . . 127
23.7. Bulk creation . . . 127
23.8. Filters data . . . 128
23.9. Vote a task . . . 128
23.10. Remove vote from a task . . . 128
23.11. Get task voters list . . . 128
23.12. Watch a task . . . 129
23.13. Stop watching a task . . . 129
23.14. List task watchers . . . 129
23.15. List attachments . . . 129
23.16. Create attachment . . . 130
23.17. Get attachment . . . 130
23.18. Edit attachment . . . 130
23.19. Delete attachment . . . 131
24. Task status . . . 131
24.1. List . . . 131
24.2. Create . . . 132
24.3. Get . . . 132
24.4. Edit . . . 133
24.5. Delete . . . 133
24.6. Bulk update order . . . 133
25. Task custom attribute . . . 134
25.1. List . . . 134
25.2. Create . . . 135
25.3. Get . . . 135
25.4. Edit . . . 136
25.5. Delete . . . 136
25.6. Bulk update order . . . 136
26. Task custom attributes values . . . 137
26.1. Get . . . 137
26.2. Edit . . . 137
27. Issues . . . 138
27.1. List . . . 138
27.2. Create . . . 139
27.3. Get . . . 141
27.4. Get by ref . . . 141
27.5. Edit . . . 141
27.6. Delete . . . 142

27.7. Filters data . . . 142
27.8. Vote an issue . . . 142
27.9. Remove vote from an issue . . . 142
27.10. Get issue voters list . . . 143
27.11. Watch an issue . . . 143
27.12. Stop watching an issue . . . 143
27.13. List issue watchers . . . 143
27.14. List attachments . . . 144
27.15. Create attachment . . . 144
27.16. Get attachment . . . 144
27.17. Edit attachment . . . 145
27.18. Delete attachment . . . 145
28. Issue status . . . 145
28.1. List . . . 145
28.2. Create . . . 146
28.3. Get . . . 147
28.4. Edit . . . 147
28.5. Delete . . . 147
28.6. Bulk update order . . . 147
29. Issue types . . . 148
29.1. List . . . 148
29.2. Create . . . 149
29.3. Get . . . 149
29.4. Edit . . . 150
29.5. Delete . . . 150
29.6. Bulk update order . . . 150
30. Priorities . . . 151
30.1. List . . . 151
30.2. Create . . . 152
30.3. Get . . . 152
30.4. Edit . . . 153
30.5. Delete . . . 153
30.6. Bulk update order . . . 153
31. Severities . . . 154
31.1. List . . . 154
31.2. Create . . . 155
31.3. Get . . . 155
31.4. Edit . . . 156
31.5. Delete . . . 156
31.6. Bulk update order . . . 156
32. Issue custom attribute . . . 157

32.1. List . . . 157
32.2. Create . . . 158
32.3. Get . . . 158
32.4. Edit . . . 159
32.5. Delete . . . 159
32.6. Bulk update order . . . 159
33. Issue custom attributes values . . . 160
33.1. Get . . . 160
33.2. Edit . . . 160
34. Wiki pages . . . 161
34.1. List . . . 161
34.2. Create . . . 161
34.3. Get . . . 162
34.4. Get by slug . . . 162
34.5. Edit . . . 163
34.6. Delete . . . 163
34.7. Watch a wiki page . . . 163
34.8. Stop watching a wiki page . . . 163
34.9. List wiki page watchers . . . 164
34.10. List attachments . . . 164
34.11. Create attachment . . . 164
34.12. Get attachment . . . 165
34.13. Edit attachment . . . 165
34.14. Delete attachment . . . 166
35. Wiki links . . . 166
35.1. List . . . 166
35.2. Create . . . 167
35.3. Get . . . 167
35.4. Edit . . . 168
35.5. Delete . . . 168
36. History . . . 168
36.1. Get user story, task, issue or wiki page history . . . 168
36.2. Get comment versions . . . 169
36.3. Edit comment . . . 169
36.4. Delete comment . . . 169
36.5. Undelete comment . . . 170
37. Users . . . 170
37.1. List . . . 170
37.2. Get . . . 170
37.3. Me . . . 171
37.4. Get user stats . . . 171

37.5. Get watched content . . . 171
37.6. Get liked content . . . 172
37.7. Get voted content . . . 172
37.8. Edit . . . 172
37.9. Delete . . . 173
37.10. Get contacts . . . 173
37.11. Cancel . . . 173
37.12. Change avatar . . . 174
37.13. Remove avatar . . . 174
37.14. Change email . . . 174
37.15. Change password . . . 175
37.16. Password recovery . . . 175
37.17. Change password from recovery . . . 176
38. Notify policies . . . 176
38.1. List . . . 176
38.2. Get . . . 176
38.3. Edit . . . 177
39. Feedback . . . 177
39.1. Create . . . 177
40. Export/Import . . . 177
40.1. Export . . . 177
40.2. Import . . . 178
41. Webhooks . . . 178
41.1. List . . . 178
41.2. Create . . . 179
41.3. Get . . . 179
41.4. Edit . . . 180
41.5. Delete . . . 180
41.6. Test . . . 180
41.7. Logs list . . . 180
41.8. Log get . . . 181
41.9. Resend request . . . 181
42. Timelines . . . 181
42.1. List user timeline . . . 182
42.2. List profile timeline . . . 182
42.3. List project timeline . . . 182
43. Locales . . . 182
43.1. List . . . 183
44. Stats . . . 183
44.1. Get discover stats . . . 183
44.2. Get system stats . . . 183

45. Importers . . . 183
45.1. Trello . . . 183
45.2. Github . . . 185
45.3. Jira . . . 187
46. Contact . . . 189
46.1. Contact project . . . 189
47. Objects Summary . . . 190
47.1. Attachment . . . 190
47.2. Application token object . . . 191
47.3. Authorization code object . . . 191
47.4. Cyphered token object . . . 191
47.5. User detail . . . 191
47.6. User contact detail . . . 192
47.7. User authentication-detail . . . 193
47.8. User stats detail . . . 194
47.9. Search results detail . . . 195
47.10. User storage data . . . 199
48. Project templates detail . . . 199
48.1. Project list entry . . . 211
48.2. Project detail . . . 213
48.3. Project modules configuration . . . 240
48.4. Project stats detail . . . 241
48.5. Project issue stats detail . . . 243
48.6. Project tag colors data detail . . . 255
48.7. Project voter detail . . . 258
48.8. Project watcher detail . . . 258
48.9. Membership detail . . . 258
48.10. Role detail . . . 259
48.11. Milestone detail . . . 260
48.12. Milestone watcher detail . . . 266
48.13. Milestone stats detail . . . 266
48.14. Epic detail . . . 268
48.15. Epic detail (GET) . . . 270
48.16. Epic detail (LIST) . . . 272
48.17. Epic filters data detail . . . 274
48.18. Epic voter detail . . . 292
48.19. Epic watcher detail . . . 292
48.20. Epic status detail . . . 292
48.21. Epic custom attribute detail . . . 292
48.22. Epic custom attributes values detail . . . 293
48.23. Epic related user story detail . . . 293

48.24. User story detail . . . 293
48.25. User story detail (GET) . . . 296
48.26. User story detail (LIST) . . . 298
48.27. Issue filters data detail . . . 300
48.28. User story voter detail . . . 323
48.29. User story watcher detail . . . 323
48.30. User story status detail . . . 323
48.31. Point detail . . . 323
48.32. User story custom attribute detail . . . 324
48.33. User story custom attributes values detail . . . 324
48.34. Task detail . . . 324
48.35. Task detail (GET) . . . 327
48.36. Task detail (LIST) . . . 330
48.37. Task filters data detail . . . 332
48.38. Task voter detail . . . 338
48.39. Task watcher detail . . . 338
48.40. Task status detail . . . 338
48.41. Task custom attribute detail . . . 338
48.42. Task custom attributes values detail . . . 339
48.43. Issue detail . . . 339
48.44. Issue detail (GET) . . . 341
48.45. Issue detail (LIST) . . . 343
48.46. Issue filters data detail . . . 345
48.47. Issue voters detail . . . 367
48.48. Issue watchers detail . . . 367
48.49. Issue status detail . . . 367
48.50. Issue type detail . . . 368
48.51. Priority detail . . . 368
48.52. Severity detail . . . 368
48.53. Issue custom attribute detail . . . 368
48.54. Issue custom attributes values detail . . . 369
48.55. Wiki page . . . 369
48.56. Wiki page watcher detail . . . 372
48.57. Wiki link . . . 373
48.58. History entry comment . . . 373
48.59. History entry . . . 373
48.60. Notify policy . . . 374
48.61. Feedback . . . 374
48.62. Export detail for synch mode . . . 375
48.63. Export accepted response . . . 375
48.64. Import accepted response . . . 375

48.65. Webhook . . . 375
48.66. Webhook log . . . 375
48.67. Timeline entry detail . . . 376
48.68. Locale . . . 378
48.69. Watched . . . 380
48.70. Liked . . . 382
48.71. Voted . . . 383
48.72. Contact . . . 384
48.73. Discover stats . . . 385
48.74. System stats . . . 385
48.75. Importer Trello Auth Url . . . 386
48.76. Importer Trello Auth Token . . . 386
48.77. Importer Trello list users . . . 387
48.78. Importer Trello list projects . . . 387
48.79. Importer Trello import project . . . 387
48.80. Importer Github Auth Url . . . 388
48.81. Importer Github Auth Token . . . 388
48.82. Importer Github list users . . . 388
48.83. Importer Github list projects . . . 389
48.84. Importer Github import project . . . 389
48.85. Importer Jira Auth Url . . . 390
48.86. Importer Jira Auth Token . . . 390
48.87. Importer Jira list users . . . 390
48.88. Importer Jira list projects . . . 391
48.89. Importer Jira import project . . . 391
48.90. Importer Import project task accepted . . . 392
49. Contrib plugins . . . 392

1. General notes

NOTE

About Taiga instance and URLs used in this document

All API calls used in the documentation are refered to a local taiga instance API running on localhost:8000, so if you use another instance remember to change the url.

For example, if you want to perform the tests against our own instance, you should use *https://api.taiga.io/api/v1* instead of *http://localhost:8000/api/v1*.

1.1. Authentication

1.1.1. Standard token authentication

To authenticate requests an http header called "Authorization" should be added. Its format should be:

```
Authorization: Bearer ${AUTH_TOKEN}
```

This token can be received through the login API

To provide an example, the following can be used within a Bash script running on Ubuntu - customise as appropriate for your system configuration.

- Install `jq` (a command-line JSON processor):

```
$ sudo apt-get install jq
```

- Bash snippet:

```bash
#!/bin/bash
# Request username and password for connecting to Taiga
read -p "Username or email: " USERNAME
read -r -s -p "Password: " PASSWORD

DATA=$(jq --null-input \
        --arg username "$USERNAME" \
        --arg password "$PASSWORD" \
        '{ type: "normal", username: $username, password: $password }')

# Get AUTH_TOKEN
USER_AUTH_DETAIL=$( curl -X POST \
  -H "Content-Type: application/json" \
  -d "$DATA" \
  https://api.taiga.io/api/v1/auth 2>/dev/null )

AUTH_TOKEN=$( echo ${USER_AUTH_DETAIL} | jq -r '.auth_token' )

# Exit if AUTH_TOKEN is not available
if [ -z ${AUTH_TOKEN} ]; then
    echo "Error: Incorrect username and/or password supplied"
    exit 1
else
    echo "auth_token is ${AUTH_TOKEN}"
fi

# Proceed to use API calls as desired
...
```

- If unable to install `jq`, it is possible (but not recommended) to use `grep` and `cut` to extract the value of `auth_token` from the JSON user auth detail object - use the following line instead:

```
AUTH_TOKEN=$( echo ${USER_AUTH_DETAIL} | grep -Po '"auth_token":.*?[^\\]",' | cut -d\"
-f4 )
```

1.1.2. Application token authentication

This kind of tokens are designed for allowing external apps use the Taiga API, they are associated to an existing user and an Application. They can be manually created via the django ADMIN or programatically created via API.

They work in the same way than standard Taiga authentication tokens but the "Authorization" header change slightly. Its format should be:

```
Authorization: Application ${AUTH_TOKEN}
```

The process for obtaining a valid token consists in:

- Checking if there is an existing application token for the requesting user
- Requesting an authorization code for the requesting user if it doesn't exist yet
- Validating the authorization code to obtain the final token
- Decyphering the token

Checking if there is an existing application token for the requesting user

A GET request must be done to the applications resource including the application id in the url and specifying the token endpoint:

```
curl -X GET \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer {AUTH_TOKEN}" \
    https://api.taiga.io/api/v1/applications/5c8515c2-4fc4-11e5-9a5e-
68f72800aadd/token
```

The API will answer with info about the application and the token:

```
{
    "user": 4,
    "id": null,
    "application": {
        "id": "a60c3208-5234-11e5-96df-68f72800aadd",
        "name": "Testing application",
        "web": "http://taiga.io",
        "description": "Testing external app",
        "icon_url": "https://tree.taiga.io/images/beta.png"
    },
    "auth_code": null,
    "next_url": "http://tree.taiga.io/redirect?auth_code=None"
}
```

If id and auth_code are null it means there is no application token generated and you need to authorize one. If they are not null you can jump to the validation step.

Requesting an authorization code for the requesting user if it doesn't exist yet

The request should include:

- application: the application id for the requested token
- state: an unguessable random string. It is used to protect against cross-site request forgery attacks. The API will include this value when the validation process is completed so the final app can verify it matches the original one.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer
eyJ1c2VyX2F1dGhlbnRpY2F0aW9uX2lkIjo0fQ:1ZX33b:QnAN3EcuChLoRVf3CdybWEi2OEg" \
    -d '{
          "application": "a60c3208-5234-11e5-96df-68f72800aadd",
          "state": "random-state"
      }' \
    https://api.taiga.io/api/v1/application-tokens/authorize
```

The API answer will be something like:

```
{
    "auth_code": "c8bfacba-5236-11e5-b8f6-68f72800aadd",
    "state": "random-state",
    "next_url": "asd?auth_code=c8bfacba-5236-11e5-b8f6-68f72800aadd"
}
```

The obtained auth_code must be validated as described in the validation step.

Validating the authorization code to obtain the final token

Now the external app must validate the auth_code obtained in the previous steps with a request including:

- application: the application id for the requested token
- state: an unguessable random string. It is used to protect against cross-site request forgery attacks. The API will include this value when the validation process is completed so the final app can verify it matches the original one.
- auth_code: the authorization code received on previous the steps.

```
curl -X POST \
    -H "Content-Type: application/json" \
    -H "Authorization: Bearer
eyJ1c2VyX2F1dGhlbnRpY2F0aW9uX2lkIjo0fQ:1ZX33b:QnAN3EcuChLoRVf3CdybWEi2OEg" \
    -d '{
          "application": "a60c3208-5234-11e5-96df-68f72800aadd",
          "auth_code": "21ce08c4-5237-11e5-a8a3-68f72800aadd",
          "state": "random-state"
      }' \
https://api.taiga.io/api/v1/application-tokens/validate
```

The API answer will be something like:


```
{
    "cyphered_token": "eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiQTEyOEtXIn0.E-Ee1cRgG0JEd90yJu-
Dgl_vwKHTHdPy2YHRbCsMvfiJx0OvR12E8g.kGwJPnWQJecFPEae.ebQtpRNPbKh6FBS-
LSUhw1xNARl0Q5loCO4fAk00LHFqcDpAwba7LHeR3MPx9T9LfA.KM-Id_041g8OdWaseGyV8g"
}
```


Decyphering the token

The token is cyphered using JWE with A128KW as algorythm and A256GCM as encryption. Both parts (Taiga and the external application requesting the token) must know about the encryption key used in the process (in Taiga it's an attribute of the application model).

- A python snippet for decyphering the token:

```
from jwkest.jwk import SYMKey
from jwkest.jwe import JWE
key ="this-is-the-secret-key"
cyphered_token="eyJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiQTEyOEtXIn0.H5jWzzXQISSh_QPCO5mWhT0EI9
RRV45xA7vbWoxeBIjiCL3qwAmlzg.bBWVKwGTkta5y99c.ArycfFtrlmWgyZ4lwXw_JiSVmkn9YF6Xwlh8nVDk
u0BLW8kvaxNy3XRbbb17MtZ7mg.pDkpgDwffCyCy4sYNQI6zA"
sym_key = SYMKey(key=key, alg="A128KW")
token=JWE().decrypt(cyphered_token, keys=[sym_key])
print(token)
```

When decyphering it correctly you will obtain a json containing the application token that can be used in the Authorization headers

```
{
    "token": "95db1710-5238-11e5-a86e-68f72800aadd"
}
```

1.2. OCC - Optimistic concurrency control

In taiga multiple operations can be happening at the same time for an element so every modifying request should include a valid version parameter. You can think about two different users updating the same user story, there are two possible scenarios here:

- They are updating the same attributes on the element. In this situation the API will accept the first request and deny the second one because the version parameter will be considered as invalid.
- They are updating different attributes on the element. In this situation the API is smart enough for accepting both requests, the second one would have an invalid version but the changes are not affecting modified attributes so they can be applied safely

The version parameter is considered valid if it contains the current version for the element, it will be incremented automatically if the modification is successful.

1.3. Pagination

By default the API will always return paginated results and includes the following headers in the response:

- x-paginated: boolean indicating if pagination is being used for the request
- x-paginated-by: number of results per page
- x-pagination-count: total number of results
- x-pagination-current: current page
- x-pagination-next: next results
- x-pagination-prev: previous results

Disabling pagination can be accomplished by setting an extra http header:

```
x-disable-pagination: True
```

1.4. Internationalization

The API returns some content translated, you can specify the language with an extra http header:

```
Accept-Language: {LanguageId}
```

The LanguageId can be chosen from the value list of available languages. You can get them using the locales API.

1.5. Throttling

If the api is configured with throttling you have to take care on responses with 429 (Too many requests) status code, that mean you reach the throttling limit.

1.6. Read only fields

All the fields ending in _extra_info (assigned_to_extra_info, is_private_extra_info, owner_extra_info, project_extra_info, status_extra_info, status_extra_info, user_story_extra_info...) are read only fields

2. Endpoints Summary

2.1. Auth

URL	Method	Functionality
/api/v1/auth	POST	Login
/api/v1/auth/register	POST	Register user

2.2. Applications

URL	Method	Functionality
/api/v1/applications/{applicationId}	GET	Get application
/api/v1/applications/{applicationId}/token	GET	Get application token

2.3. Application Tokens

URL	Method	Functionality
/api/v1/application-tokens	GET	List application tokens
/api/v1/application-tokens/{applicationTokenId}	GET	Get application token
/api/v1/application-tokens/{applicationTokenId}	DELETE	Delete application token
/api/v1/application-tokens/authorize	POST	Authorize application token
/api/v1/application-tokens/validate	POST	Validate application token

2.4. Resolver

URL	Method	Functionality
/api/v1/resolver	GET	Resolve references and slugs

2.5. Searches

URL	Method	Functionality
/api/v1/search	GET	Search in a project

2.6. User storage

URL	Method	Functionality
/api/v1/user-storage	GET	List user storage data
/api/v1/user-storage	POST	Create user storage data
/api/v1/user-storage/{key}	GET	Get user storage data
/api/v1/user-storage/{key}	PUT	Modify user storage data
/api/v1/user-storage/{key}	PATCH	Modify partially an user storage data
/api/v1/user-storage/{key}	DELETE	Delete user storage data

2.7. Project templates

URL	Method	Functionality
/api/v1/project-templates	GET	List project templates

URL	Method	Functionality
/api/v1/project-templates	POST	Create project template
/api/v1/project-templates/{projectTemplateId}	GET	Get project template
/api/v1/project-templates/{projectTemplateId}	PUT	Modify project template
/api/v1/project-templates/{projectTemplateId}	PATCH	Modify partially an project template
/api/v1/project-templates/{projectTemplateId}	DELETE	Delete an project template

2.8. Projects

URL	Method	Functionality
/api/v1/projects	GET	List projects
/api/v1/projects	POST	Create project
/api/v1/projects/{projectId}	GET	Get project
/api/v1/projects/by_slug?slug={projectSlug}	GET	Get project
/api/v1/projects/{projectId}	PUT	Modify project
/api/v1/projects/{projectId}	PATCH	Modify partially a project
/api/v1/projects/{projectId}	DELETE	Delete a project
/api/v1/projects/bulk_update_order	POST	Update projects order for logged in user
/api/v1/projects/{projectId}/modules	GET	Get project modules configuration
/api/v1/projects/{projectId}/modules	PATCH	Modify partially a project modules configuration
/api/v1/projects/{projectId}/stats	GET	Get project stats
/api/v1/projects/{projectId}/issues_stats	GET	Get project issue stats
/api/v1/projects/{projectId}/tags_colors	GET	Get project tags colors
/api/v1/projects/{projectId}/create_tag	POST	Create project tag
/api/v1/projects/{projectId}/edit_tag	POST	Edit project tag

URL	Method	Functionality
/api/v1/projects/{projectId}/delete_tag	POST	Delete project tag
/api/v1/projects/{projectId}/mix_tags	POST	Mix project tags
/api/v1/projects/{projectId}/like	POST	Like a project
/api/v1/projects/{projectId}/unlike	POST	Unlike a project
/api/v1/projects/{projectId}/fans	GET	Get project fans
/api/v1/projects/{projectId}/watch	POST	Watch a project
/api/v1/projects/{projectId}/unwatch	POST	Unwatch a project
/api/v1/projects/{projectId}/watchers	GET	Get project watchers
/api/v1/projects/{projectId}/create_template	POST	Create project template
/api/v1/projects/{projectId}/leave	POST	Leave project
/api/v1/projects/{projectId}/change_logo	POST	Change logo
/api/v1/projects/{projectId}/remove_logo	POST	Remove logo
/api/v1/projects/{projectId}/transfer_validate_token	POST	Transfer validate token
/api/v1/projects/{projectId}/transfer_request	POST	Transfer request
/api/v1/projects/{projectId}/transfer_start	POST	Transfer start
/api/v1/projects/{projectId}/transfer_accept	POST	Transfer accept
/api/v1/projects/{projectId}/transfer_reject	POST	Transfer reject
/api/v1/projects/{projectId}/duplicate	POST	Duplicate project

2.9. Memberships/Invitations

URL	Method	Functionality
/api/v1/memberships	GET	List memberships
/api/v1/memberships	POST	Create membership
/api/v1/memberships/bulk_create	POST	Create a bulk of memberships
/api/v1/memberships/{membershipId}	GET	Get membership
/api/v1/memberships/{membershipId}	PUT	Modify membership
/api/v1/memberships/{membershipId}	PATCH	Modify partially a membership
/api/v1/memberships/{membershipId}	DELETE	Delete a membership
/api/v1/memberships/{membershipId}/resend_invitation	POST	Resend invitation
/api/v1/invitations/{invitationUuid}	POST	Get invitation by anonymous user

2.10. Roles

URL	Method	Functionality
/api/v1/roles	GET	List roles
/api/v1/roles	POST	Create role
/api/v1/roles/{roleId}	GET	Get role
/api/v1/roles/{roleId}	PUT	Modify role
/api/v1/roles/{roleId}	PATCH	Modify partially a role
/api/v1/roles/{roleId}	DELETE	Delete a role

2.11. Milestones

URL	Method	Functionality
/api/v1/milestones	GET	List milestones
/api/v1/milestones	POST	Create milestone
/api/v1/milestones/{milestoneId}	GET	Get milestone
/api/v1/milestones/{milestoneId}	PUT	Modify milestone

URL	Method	Functionality
/api/v1/milestones/{milestoneId}	PATCH	Modify partially a milestone
/api/v1/milestones/{milestoneId}	DELETE	Delete a milestone
/api/v1/milestones/{milestoneId}/stats	GET	Get a milestone stats
/api/v1/milestones/{milestoneId}/watch	POST	Watch a milestone
/api/v1/milestones/{milestoneId}/unwatch	POST	Stop watching a milestone
/api/v1/milestones/{milestoneId}/watchers	GET	Get milestone watchers

2.12. Epics

URL	Method	Functionality
/api/v1/epics	GET	List epics
/api/v1/epics	POST	Create epic
/api/v1/epics/{epicId}	GET	Get epic
/api/v1/epics/by_ref?ref={epicRef}&project={projectId}	GET	Get epic
/api/v1/epics/{epicId}	PUT	Modify epic
/api/v1/epics/{epicId}	PATCH	Modify partially an epic
/api/v1/epics/{epicId}	DELETE	Delete an epic
/api/v1/epics/{epicId}/related_userstories	GET	List epic related userstories
/api/v1/epics/{epicId}/related_userstories	POST	Create epic related user story
/api/v1/epics/{epicId}/related_userstories/{userStoryId}	GET	Get epic related userstory
/api/v1/epics/{epicId}/related_userstories/{userStoryId}	PUT	Modify epic related user story
/api/v1/epics/{epicId}/related_userstories/{userStoryId}	PATCH	Modify partially an epic related user story
/api/v1/epics/{epicId}/related_userstories/{userStoryId}	DELETE	Delete an epic related user story
/api/v1/epics/{epicId}/related_userstories/bulk_create	POST	Create epic related user stories on bulk mode

URL	Method	Functionality
/api/v1/epics/bulk_create	POST	Create epics on bulk mode
/api/v1/epics/filters_data?project={projectId}	GET	Get filters data
/api/v1/epics/{epicId}/upvote	POST	Add star to an epic
/api/v1/epics/{epicId}/downvote	POST	Remove star from epic
/api/v1/epics/{epicId}/voters	GET	Get epic voters
/api/v1/epics/{epicId}/watch	POST	Watch an epic
/api/v1/epics/{epicId}/unwatch	POST	Unwatch an epic
/api/v1/epics/{epicId}/watchers	GET	Get epic watchers
/api/v1/epics/attachments	GET	List epic attachments
/api/v1/epics/attachments	POST	Create epic attachments
/api/v1/epics/attachments/{epicAttachmentId}	GET	Get epic attachments
/api/v1/epics/attachments/{epicAttachmentId}	PUT	Modify epic attachments
/api/v1/epics/attachments/{epicAttachmentId}	PATCH	Modify partially an epic attachments
/api/v1/epics/attachments/{epicAttachmentId}	DELETE	Delete an epic attachments

2.13. Epic status

URL	Method	Functionality
/api/v1/epic-statuses	GET	List epic statuses
/api/v1/epic-statuses	POST	Create epic status
/api/v1/epic-statuses/{epicStatusId}	GET	Get epic status
/api/v1/epic-statuses/{epicStatusId}	PUT	Modify epic status
/api/v1/epic-statuses/{epicStatusId}	PATCH	Modify partially an epic status
/api/v1/epic-statuses/{epicStatusId}	DELETE	Delete an epic status
/api/v1/epic-statuses/bulk_update_order	POST	Update epic statuses order in bulk mode

2.14. Epic custom attribute

URL	Method	Functionality
/api/v1/epic-custom-attributes	GET	List epic custom attributes
/api/v1/epic-custom-attributes	POST	Create epic custom attribute
/api/v1/epic-custom-attributes/{epicCustomAttributeId}	GET	Get epic custom attribute
/api/v1/epic-custom-attributes/{epicCustomAttributeId}	PUT	Modify epic custom attribute
/api/v1/epic-custom-attributes/{epicCustomAttributeId}	PATCH	Modify partially an epic custom attribute
/api/v1/epic-custom-attributes/{epicCustomAttributeId}	DELETE	Delete an epic custom attribute
/api/v1/epic-custom-attributes/bulk_update_order	POST	Update epic custom attributes order in bulk mode

2.15. Epic custom attributes values

URL	Method	Functionality
/api/v1/epics/custom-attributes-values/{epicId}	GET	Get epic custom attributes values
/api/v1/epics/custom-attributes-values/{epicId}	PUT	Modify epic custom attributes values
/api/v1/epics/custom-attributes-values/{epicId}	PATCH	Modify partially an epic custom attributes values

2.16. User stories

URL	Method	Functionality
/api/v1/userstories	GET	List user stories
/api/v1/userstories	POST	Create user story
/api/v1/userstories/{userStoryId}	GET	Get user story
/api/v1/userstories/by_ref?ref={userStoryRef}&project={userStoryId}	GET	Get user story

URL	Method	Functionality
/api/v1/userstories/{userStoryId}	PUT	Modify user story
/api/v1/userstories/{userStoryId}	PATCH	Modify partially a user story
/api/v1/userstories/{userStoryId}	DELETE	Delete a user story
/api/v1/userstories/bulk_create	POST	Create user stories un bulk mode
/api/v1/userstories/bulk_update_backlog_order	POST	Update user stories order for backlog in bulk mode
/api/v1/userstories/bulk_update_kanban_order	POST	Update user stories order for kanban in bulk mode
/api/v1/userstories/bulk_update_sprint_order	POST	Update user stories order for sprint in bulk mode
/api/v1/userstories/bulk_update_milestone	POST	Update user stories sprint in bulk mode
/api/v1/userstories/filters_data?project={projectId}	GET	Get filters data
/api/v1/userstories/{userStoryId}/upvote	POST	Add star to a user story
/api/v1/userstories/{userStoryId}/downvote	POST	Remove star from user story
/api/v1/userstories/{userStoryId}/voters	GET	Get user story voters
/api/v1/userstories/{userStoryId}/watch	POST	Watch a user story
/api/v1/userstories/{userStoryId}/unwatch	POST	Unwatch a user story
/api/v1/userstories/{userStoryId}/watchers	GET	Get user story watchers
/api/v1/userstories/attachments	GET	List user story attachments
/api/v1/userstories/attachments	POST	Create user story attachments
/api/v1/userstories/attachments/{userStoryAttachmentId}	GET	Get user story attachments
/api/v1/userstories/attachments/{userStoryAttachmentId}	PUT	Modify user story attachments
/api/v1/userstories/attachments/{userStoryAttachmentId}	PATCH	Modify partially a user story attachments

URL	Method	Functionality
/api/v1/userstories/attachments/{userStoryAttachmentId}	DELETE	Delete a user story attachments

2.17. User story status

URL	Method	Functionality
/api/v1/userstory-statuses	GET	List user story status
/api/v1/userstory-statuses	POST	Create user story status
/api/v1/userstory-statuses/{userStoryStatusId}	GET	Get user story status
/api/v1/userstory-statuses/{userStoryStatusId}	PUT	Modify user story status
/api/v1/userstory-statuses/{userStoryStatusId}	PATCH	Modify partially a user story status
/api/v1/userstory-statuses/{userStoryStatusId}	DELETE	Delete a user story status
/api/v1/userstory-statuses/bulk_update_order	POST	Update user story statuses order in bulk mode

2.18. Points

URL	Method	Functionality
/api/v1/points	GET	List points
/api/v1/points	POST	Create point
/api/v1/points/{pointId}	GET	Get point
/api/v1/points/{pointId}	PUT	Modify point
/api/v1/points/{pointId}	PATCH	Modify partially a point
/api/v1/points/{pointId}	DELETE	Delete a point
/api/v1/points/bulk_update_order	POST	Update points order in bulk mode

2.19. User story custom attribute

URL	Method	Functionality
/api/v1/userstory-custom-attributes	GET	List user story custom attributes
/api/v1/userstory-custom-attributes	POST	Create user story custom attribute

URL	Method	Functionality
/api/v1/userstory-custom-attributes/{userStoryCustomAttributeId}	GET	Get user story custom attribute
/api/v1/userstory-custom-attributes/{userStoryCustomAttributeId}	PUT	Modify user story custom attribute
/api/v1/userstory-custom-attributes/{userStoryCustomAttributeId}	PATCH	Modify partially a user story custom attribute
/api/v1/userstory-custom-attributes/{userStoryCustomAttributeId}	DELETE	Delete a user story custom attribute
/api/v1/userstory-custom-attributes/bulk_update_order	POST	Update user story custom attributes order in bulk mode

2.20. User story custom attributes values

URL	Method	Functionality
/api/v1/userstories/custom-attributes-values/{userStoryId}	GET	Get user story custom attributes values
/api/v1/userstories/custom-attributes-values/{userStoryId}	PUT	Modify user story custom attributes values
/api/v1/userstories/custom-attributes-values/{userStoryId}	PATCH	Modify partially a user story custom attributes values

2.21. Tasks

URL	Method	Functionality
/api/v1/tasks	GET	List tasks
/api/v1/tasks	POST	Create task
/api/v1/tasks/{taskId}	GET	Get task
/api/v1/tasks/by_ref?ref={taskRef}&project={projectId}	GET	Get task
/api/v1/tasks/{taskId}	PUT	Modify task
/api/v1/tasks/{taskId}	PATCH	Modify partially a task
/api/v1/tasks/{taskId}	DELETE	Delete a task
/api/v1/tasks/bulk_create	POST	Create tasks on bulk mode

URL	Method	Functionality
/api/v1/tasks/filters_data?project={projectId}	GET	Get filters data
/api/v1/tasks/{taskId}/upvote	POST	Add star to a task
/api/v1/tasks/{taskId}/downvote	POST	Remove star from task
/api/v1/tasks/{taskId}/voters	GET	Get task voters
/api/v1/tasks/{taskId}/watch	POST	Watch a task
/api/v1/tasks/{taskId}/unwatch	POST	Unwatch a task
/api/v1/tasks/{taskId}/watchers	GET	Get task watchers
/api/v1/tasks/attachments	GET	List task attachments
/api/v1/tasks/attachments	POST	Create task attachments
/api/v1/tasks/attachments/{taskAttachmentId}	GET	Get task attachments
/api/v1/tasks/attachments/{taskAttachmentId}	PUT	Modify task attachments
/api/v1/tasks/attachments/{taskAttachmentId}	PATCH	Modify partially a task attachments
/api/v1/tasks/attachments/{taskAttachmentId}	DELETE	Delete a task attachments

2.22. Task status

URL	Method	Functionality
/api/v1/task-statuses	GET	List task statuses
/api/v1/task-statuses	POST	Create task status
/api/v1/task-statuses/{taskStatusId}	GET	Get task status
/api/v1/task-statuses/{taskStatusId}	PUT	Modify task status
/api/v1/task-statuses/{taskStatusId}	PATCH	Modify partially a task status
/api/v1/task-statuses/{taskStatusId}	DELETE	Delete a task status
/api/v1/task-statuses/bulk_update_order	POST	Update task statuses order in bulk mode

2.23. Task custom attribute

URL	Method	Functionality
/api/v1/task-custom-attributes	GET	List task custom attributes
/api/v1/task-custom-attributes	POST	Create task custom attribute
/api/v1/task-custom-attributes/{taskCustomAttributeId}	GET	Get task custom attribute
/api/v1/task-custom-attributes/{taskCustomAttributeId}	PUT	Modify task custom attribute
/api/v1/task-custom-attributes/{taskCustomAttributeId}	PATCH	Modify partially a task custom attribute
/api/v1/task-custom-attributes/{taskCustomAttributeId}	DELETE	Delete a task custom attribute
/api/v1/task-custom-attributes/bulk_update_order	POST	Update task custom attributes order in bulk mode

2.24. Task custom attributes values

URL	Method	Functionality
/api/v1/tasks/custom-attributes-values/{taskId}	GET	Get task custom attributes values
/api/v1/tasks/custom-attributes-values/{taskId}	PUT	Modify task custom attributes values
/api/v1/tasks/custom-attributes-values/{taskId}	PATCH	Modify partially a task custom attributes values

2.25. Issues

URL	Method	Functionality
/api/v1/issues	GET	List issues
/api/v1/issues	POST	Create issue
/api/v1/issues/{issueId}	GET	Get issue
/api/v1/issues/by_ref?ref={issueRef}&project={projectId}	GET	Get issue
/api/v1/issues/{issueId}	PUT	Modify issue

URL	Method	Functionality
/api/v1/issues/{issueId}	PATCH	Modify partially an issue
/api/v1/issues/{issueId}	DELETE	Delete an issue
/api/v1/issues/bulk_create	POST	Create issues un bulk mode
/api/v1/issues/filters_data?project={projectId}	GET	Get filters data
/api/v1/issues/{issueId}/upvote	POST	Add a vote to an issue
/api/v1/issues/{issueId}/downvote	POST	Remove your vote to an issue
/api/v1/issues/{issueId}/voters	GET	Get issue voters list
/api/v1/issues/{issueId}/watch	POST	Watch an issue
/api/v1/issues/{issueId}/unwatch	POST	Unwatch an issue
/api/v1/issues/{issueId}/watchers	GET	Get issue watchers
/api/v1/issues/attachments	GET	List issue attachments
/api/v1/issues/attachments	POST	Create issue attachments
/api/v1/issues/attachments/{issueAttachmentId}	GET	Get issue attachments
/api/v1/issues/attachments/{issueAttachmentId}	PUT	Modify issue attachments
/api/v1/issues/attachments/{issueAttachmentId}	PATCH	Modify partially an issue attachments
/api/v1/issues/attachments/{issueAttachmentId}	DELETE	Delete an issue attachments

2.26. Issue status

URL	Method	Functionality
/api/v1/issue-statuses	GET	List issue statuses
/api/v1/issue-statuses	POST	Create issue status
/api/v1/issue-statuses/{issueStatusId}	GET	Get issue status
/api/v1/issue-statuses/{issueStatusId}	PUT	Modify issue status
/api/v1/issue-statuses/{issueStatusId}	PATCH	Modify partially a issue status
/api/v1/issue-statuses/{issueStatusId}	DELETE	Delete a issue status

URL	Method	Functionality
/api/v1/issue-statuses/bulk_update_order	POST	Update issue statuses order in bulk mode

2.27. Issue types

URL	Method	Functionality
/api/v1/issue-types	GET	List issue types
/api/v1/issue-types	POST	Create issue type
/api/v1/issue-types/{issueTypeId}	GET	Get issue type
/api/v1/issue-types/{issueTypeId}	PUT	Modify issue type
/api/v1/issue-types/{issueTypeId}	PATCH	Modify partially a issue type
/api/v1/issue-types/{issueTypeId}	DELETE	Delete a issue type
/api/v1/issue-types/bulk_update_order	POST	Update issue types order in bulk mode

2.28. Priorities

URL	Method	Functionality
/api/v1/priorities	GET	List priorities
/api/v1/priorities	POST	Create priority
/api/v1/priorities/{priorityId}	GET	Get priority
/api/v1/priorities/{priorityId}	PUT	Modify priority
/api/v1/priorities/{priorityId}	PATCH	Modify partially a priority
/api/v1/priorities/{priorityId}	DELETE	Delete a priority
/api/v1/priorities/bulk_update_order	POST	Update priorities order in bulk mode

2.29. Severities

URL	Method	Functionality
/api/v1/severities	GET	List severities
/api/v1/severities	POST	Create severity
/api/v1/severities/{severityId}	GET	Get severity

URL	Method	Functionality
/api/v1/severities/{severityId}	PUT	Modify severity
/api/v1/severities/{severityId}	PATCH	Modify partially a severity
/api/v1/severities/{severityId}	DELETE	Delete a severity
/api/v1/severities/bulk_update_order	POST	Update severities order in bulk mode

2.30. Issue custom attribute

URL	Method	Functionality
/api/v1/issue-custom-attributes	GET	List issue custom attributes
/api/v1/issue-custom-attributes	POST	Create issue custom attribute
/api/v1/issue-custom-attributes/{issueCustomAttributeId}	GET	Get issue custom attribute
/api/v1/issue-custom-attributes/{issueCustomAttributeId}	PUT	Modify issue custom attribute
/api/v1/issue-custom-attributes/{issueCustomAttributeId}	PATCH	Modify partially a issue custom attribute
/api/v1/issue-custom-attributes/{issueCustomAttributeId}	DELETE	Delete a issue custom attribute
/api/v1/issue-custom-attributes/bulk_update_order	POST	Update issue custom attributes order in bulk mode

2.31. Issue custom attributes values

URL	Method	Functionality
/api/v1/issues/custom-attributes-values/{issueId}	GET	Get issue custom attributes values
/api/v1/issues/custom-attributes-values/{issueId}	PUT	Modify issue custom attributes values
/api/v1/issues/custom-attributes-values/{issueId}	PATCH	Modify partially a issue custom attributes values

2.32. Wiki pages

URL	Method	Functionality
/api/v1/wiki	GET	List wiki pages
/api/v1/wiki	POST	Create wiki page
/api/v1/wiki/{wikiId}	GET	Get wiki page
/api/v1/wiki/by_slug?slug={wikiPageSlug}&project={projectId}	GET	Get wiki page
/api/v1/wiki/{wikiPageId}	PUT	Modify wiki page
/api/v1/wiki/{wikiPageId}	PATCH	Modify partially an wiki page
/api/v1/wiki/{wikiPageId}	DELETE	Delete an wiki page
/api/v1/wiki/{wikiPageId}/watch	POST	Watch a wiki page
/api/v1/wiki/{wikiPageId}/unwatch	POST	Stop watching a wiki page
/api/v1/wiki/{wikiPageId}/watchers	GET	Get wiki page watchers
/api/v1/wiki/attachments	GET	List wiki page attachments
/api/v1/wiki/attachments	POST	Create wiki page attachments
/api/v1/wiki/attachments/{wikiPageAttachmentId}	GET	Get wiki page attachments
/api/v1/wiki/attachments/{wikiPageAttachmentId}	PUT	Modify wiki page attachments
/api/v1/wiki/attachments/{wikiPageAttachmentId}	PATCH	Modify partially an wiki page attachments
/api/v1/wiki/attachments/{wikiPageAttachmentId}	DELETE	Delete an wiki page attachments

2.33. Wiki links

URL	Method	Functionality
/api/v1/wiki-links	GET	List wiki links
/api/v1/wiki-links	POST	Create wiki link
/api/v1/wiki-links/{wikiLinkId}	GET	Get wiki link
/api/v1/wiki-links/{wikiLinkId}	PUT	Modify wiki link
/api/v1/wiki-links/{wikiLinkId}	PATCH	Modify partially an wiki link
/api/v1/wiki-links/{wikiLinkId}	DELETE	Delete an wiki link

2.34. History

URL	Method	Functionality
/api/v1/history/userstory/{usId}	GET	Get user story history
/api/v1/history/userstory/{usId}/commentVersions?id={commentId}	GET	Get user story comment versions
/api/v1/history/userstory/{usId}/edit_comment?id={commentId}	POST	Edit user story comment
/api/v1/history/userstory/{usId}/delete_comment?id={commentId}	POST	Delete user story comment
/api/v1/history/userstory/{usId}/undelete_comment?id={commentId}	POST	Undelete user story comment
/api/v1/history/issue/{issueId}	GET	Get issue history
/api/v1/history/issue/{issueId}/commentVersions?id={commentId}	POST	Get issue comment versions
/api/v1/history/issue/{issueId}/edit_comment?id={commentId}	POST	Edit issue comment
/api/v1/history/issue/{issueId}/delete_comment?id={commentId}	POST	Delete issue comment
/api/v1/history/issue/{issueId}/undelete_comment?id={commentId}	POST	Undelete issue comment
/api/v1/history/task/<taskId>	GET	Get task history
/api/v1/history/task/{taskId}/commentVersions?id={commentId}	POST	Get task comment versions
/api/v1/history/task/{taskId}/edit_comment?id={commentId}	POST	Edit task comment
/api/v1/history/task/{taskId}/delete_comment?id={commentId}	POST	Delete task comment
/api/v1/history/task/{taskId}/undelete_comment?id={commentId}	POST	Undelete task comment
/api/v1/history/wiki/{wikiId}	GET	Get wiki history
/api/v1/history/wiki/{wikiId}/commentVersions?id={commentId}	POST	Get wiki comment versions

URL	Method	Functionality
/api/v1/history/wiki/{wikiId}/edit_comment?id={commentId}	POST	Edit wiki comment
/api/v1/history/wiki/{wikiId}/delete_comment?id={commentId}	POST	Delete wiki comment
/api/v1/history/wiki/{wikiId}/undelete_comment?id={commentId}	POST	Undelete wiki comment

2.35. Users

URL	Method	Functionality
/api/v1/users	GET	List users
/api/v1/users/{userId}	GET	Get user
/api/v1/users/me	GET	Get myself
/api/v1/users/{userId}	PUT	Modify user
/api/v1/users/{userId}	PATCH	Modify partially a user
/api/v1/users/{userId}/stats	GET	Get user stats
/api/v1/users/{userId}/watched	GET	Get user watched content
/api/v1/users/{userId}/liked	GET	Get user liked content
/api/v1/users/{userId}/voted	GET	Get user voted content
/api/v1/users/{userId}	DELETE	Delete a user
/api/v1/users/{userId}/contacts	GET	Get user contacts
/api/v1/users/cancel	POST	Cancel user
/api/v1/users/change_avatar	POST	Change avatar
/api/v1/users/remove_avatar	POST	Remove avatar
/api/v1/users/change_email	POST	Change email
/api/v1/users/change_password	POST	Change password
/api/v1/users/password_recovery	POST	Password recovery
/api/v1/users/change_password_from_recovery	POST	Change password from recovery

2.36. Notify policies

URL	Method	Functionality
/api/v1/notify-policies	GET	List notify policies

URL	Method	Functionality
/api/v1/notify-policies/{policyId}	GET	Get notify policy
/api/v1/notify-policies/{policyId}	PUT	Modify notify policy
/api/v1/notify-policies/{policyId}	PATCH	Modify partially a notify policy

2.37. Contact

URL	Method	Functionality
/api/v1/contact	POST	Contact project

2.38. Feedback

URL	Method	Functionality
/api/v1/feedback	POST	Send feedback

2.39. Export/Import

URL	Method	Functionality
/api/v1/exporter/{projectId}	GET	Export a project dump
/api/v1/importer/load_dump	POST	Import a project dump

2.40. Webhooks

URL	Method	Functionality
/api/v1/webhooks	GET	List webhooks
/api/v1/webhooks	POST	Create webhook
/api/v1/webhooks/{webhookId}	GET	Get webhook
/api/v1/webhooks/{webhookId}	PUT	Modify webhook
/api/v1/webhooks/{webhookId}	PATCH	Modify partially an webhook
/api/v1/webhooks/{webhookId}	DELETE	Delete an webhook
/api/v1/webhooks/{webhookId}/test	POST	Test webhook
/api/v1/webhooklogs	GET	List webhooks logs
/api/v1/webhooklogs/{webhookLogId}	GET	Get webhook log
/api/v1/webhooklogs/{webhookLogId}/resend	POST	Resend webhook log request

2.41. Timelines

URL	Method	Functionality
/api/v1/timeline/user/{userId}	GET	List user timeline
/api/v1/timeline/profile/{userId}	GET	List profile timeline
/api/v1/timeline/project/{projectId}	GET	List project timeline

2.42. Locales

URL	Method	Functionality
/api/v1/locales	GET	List locales

2.43. Stats

URL	Method	Functionality
/api/v1/stats/discover	GET	Get discover stats
/api/v1/stats/system	GET	Get system stats

2.44. Importers

URL	Method	Functionality
/api/v1/importers/trello/auth_url	GET	Get the authorization url
/api/v1/importers/trello/authorize	POST	Obtain the authorization token
/api/v1/importers/trello/list_projects	POST	List the Trello boards
/api/v1/importers/trello/list_users	POST	List the users related to a Trello board
/api/v1/importers/trello/import_project	POST	Import the Trello project
/api/v1/importers/github/auth_url	GET	Get the authorization url
/api/v1/importers/github/authorize	POST	Obtain the authorization token
/api/v1/importers/github/list_projects	POST	List the Github boards

URL	Method	Functionality
/api/v1/importers/github/list_users	POST	List the users related to a Github board
/api/v1/importers/github/import_project	POST	Import the Github project
/api/v1/importers/jira/auth_url	GET	Get the authorization url
/api/v1/importers/jira/authorize	POST	Obtain the authorization token
/api/v1/importers/jira/list_projects	POST	List the Jira boards
/api/v1/importers/jira/list_users	POST	List the users related to a Jira board
/api/v1/importers/jira/import_project	POST	Import the Jira project

3. Auth

3.1. Normal login

To login a user send a POST request containing the following data:

- **type** with value "normal"
- **username** (required): this field also supports the user email
- **password** (required)

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{
        "password": "password",
        "type": "normal",
        "username": "test-username"
    }' \
-s http://localhost:8000/api/v1/auth
```

When the login is successful, the HTTP response is a 200 OK and the response body is a JSON user auth detail object

3.2. Github login

To login a user via GitHub send a POST request containing the following data:

- **type** with value "github"
- **code** (required): your github authentication code

- **token**: generated when creating a project's membership (for accept invitations to projects)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -d '{
          "type": "github",
          "code": "'${GITHUB_CODE}'"
      }' \
  https://api.taiga.io/api/v1/auth
```

When the login is successful, the HTTP response is a 200 OK and the response body is a JSON user auth detail object

NOTE	*Get GitHub autorized code* To get the GitHub code you have to follow the first step *Redirect users to request GitHub access* described in GitHub Documentation for Developers - API - OAuth - Web Application Flow. Taiga needs privileges to get the user email from Github so you have to use the scope user:email.

3.3. Public registry

To register a user without invitation send a POST request containing the following data:

- **type** with value "public"
- **username** (required)
- **password** (required)
- **email** (required)
- **full_name** (required)
- **accepted_terms** (required): boolean

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{
        "accepted_terms": "true",
        "email": "test-register2@email.com",
        "full_name": "test",
        "password": "password",
        "type": "public",
        "username": "test-username2"
    }' \
-s http://localhost:8000/api/v1/auth/register
```

When the registration is successful, the HTTP response is a 201 CREATED and the response body is a

JSON user auth detail object

3.4. Private registry

To add a user into a project via invitation send a POST request containing the following data:

- **type** with value "private"
- **existing** (required): indicates if the user is member or not
- **token** (required): generated when creating a project's membership
- **username** (required)
- **password** (required)
- **email** (required only if the user doesn't exist in the platform)
- **full_name** (required only if the user doesn't exist in the platform)

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{
        "accepted_terms": "true",
        "email": "test-register@email.com",
        "existing": false,
        "full_name": "test",
        "password": "password",
        "token": "00000000-0000-0000-0000-000000000000",
        "type": "private",
        "username": "test-username"
    }' \
-s http://localhost:8000/api/v1/auth/register
```

When the registration is successful, the HTTP response is a 201 CREATED and the response body is a JSON user auth detail object

4. Applications

4.1. Get

To get an application send a GET request specifying the application id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/applications/00000000-0000-0000-0000-000000000000
```

The HTTP response is a 200 OK and the response body is a JSON application object

4.2. Get token

To get an application token send a GET request specifying the application id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/applications/00000000-0000-0000-0000-
000000000000/token
```

The HTTP response is a 200 OK and the response body is a JSON application token object

5. Application tokens

5.1. List

To list the application tokens for an authenticated user send a GET request with the following parameters:

- **application**: application id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/application-tokens
```

The HTTP response is a 200 OK and the response body is a JSON list of application token objects

5.2. Get

To get an application token send a GET request specifying the application token id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/application-tokens/1
```

The HTTP response is a 200 OK and the response body is a JSON application token object

5.3. Delete

To delete application tokens send a DELETE specifying the application token id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/application-tokens/2
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

5.4. Authorize

To request an authorization code send a POST request with the following data:

- **application**: the application id for the requested token
- **state**: an unguessable random string. It is used to protect against cross-site request forgery attacks. The API will include this value when the validation process is completed so the final app can verify it matches the original one.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "application": "00000000-0000-0000-0000-000000000000",
        "state": "random-state"
    }' \
-s http://localhost:8000/api/v1/application-tokens/authorize
```

When the creation is successful, the HTTP response is a 200 and the response body is a JSON authorization code object

5.5. Validate

To validate an authorization code send a POST request with the following data:

- **application**: the application id for the requested token
- **state**: an unguessable random string. It is used to protect against cross-site request forgery attacks. The API will include this value when the validation process is completed so the final app can verify it matches the original one.
- **auth_code**: the authorization code received on previous the steps.

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{
        "application": "00000000-0000-0000-0000-000000000000",
        "auth_code": "00000000-0000-0000-0000-000000000002",
        "state": "random-state"
    }' \
-s http://localhost:8000/api/v1/application-tokens/validate
```

When the creation is successful, the HTTP response is a 200 and the response body is a JSON cyphered token object

6. Resolver

6.1. Projects

To resolve the id of a project send a GET request with the following parameters:

- **project** (required): the project slug trying to be resolved

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0
```

The response body is a JSON object containing the project id

```
{
    "project": 1
}
```

6.2. User stories

To resolve the id of a user story send a GET request with the following parameters:

- **project** (required): the project slug trying to be resolved
- **us** (required): the user story ref trying to be resolved

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&us=1
```

The response body is a JSON object containing the project and the user story ids

```
{
    "project": 1,
    "us": 1
}
```

6.3. Issues

To resolve the id of an issue send a GET request with the following parameters:

- **project** (required): the project slug trying to be resolved
- **issue** (required): the issue ref trying to be resolved

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&issue=63
```

The response body is a JSON object containing the project and the issue ids

```
{
    "issue": 22,
    "project": 1
}
```

6.4. Tasks

To resolve the id of a task send a GET request with the following parameters:

- **project** (required): the project slug trying to be resolved
- **task** (required): the task ref trying to be resolved

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&task=2
```

The response body is a JSON object containing the project and the task ids

```
{
    "project": 1,
    "task": 1
}
```

6.5. Milestones

To resolve the id of a milestone send a GET request with the following parameters:

- **project** (required): the project slug trying to be resolved
- **milestone** (required): the milestone slug trying to be resolved

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&milestone=sprint-2020-5-8
```

The response body is a JSON object containing the project and the milestone ids

```
{
    "milestone": 1,
    "project": 1
}
```

6.6. Wiki pages

To resolve the id of a wiki page send a GET request with the following parameters:

- **project** (required): the project slug trying to be resolved
- **wikipage**(required): the wiki-page slug trying to be resolved

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&wikipage=home
```

The response body is a JSON object containing the project and the wiki page ids

```
{
    "project": 1,
    "wikipage": 1
}
```

6.7. Multiple resolution

To resolve the multiple ids you can send a GET mixing parameters from the previous examples:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&task=2\&us=1\&wikipage
=home
```

The response body is a JSON object containing the project and the task ids

```
{
    "project": 1,
    "task": 1,
    "us": 1,
    "wikipage": 1
}
```

6.8. By ref value

To resolve an object if we don't know its type we have to use `ref` GET parameter:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/resolver?project=project-0\&ref=2
```

The response body is a JSON object containing the project and the story, task or issue id.

```
{
    "project": 1,
    "task": 1
}
```

IMPORTANT	*Incompatibility between GET params* Be careful because `ref` is inconpatible with `us`, `task` and `issue` parameters in requests with multiple resolutions.

7. Searches

7.1. Search

To search send a GET request with the following get parameters:

- **project** (required): project id
- **text**: string

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/search?project=1\&text=quas
```

The HTTP response is a 200 OK and the response body is a JSON list of search results detail objects

8. User storage

8.1. List

To list user storage data send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/user-storage
```

The HTTP response is a 200 OK and the response body is a JSON list of user storage data objects

8.2. Create

To create user storage data send a POST request with the following data:

- **key**: string
- **value**: string

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "key": "favorite-forest",
        "value": "Taiga"
    }' \
-s http://localhost:8000/api/v1/user-storage
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a

JSON user storage data object

8.3. Get

To get a user storage data send a GET request specifying the user storage key in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/user-storage/favorite-forest
```

The HTTP response is a 200 OK and the response body is a JSON user storage data object

8.4. Edit

To edit user storage data send a PUT or a PATCH specifying the user storage key in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "value": "Russian Taiga"
    }' \
-s http://localhost:8000/api/v1/user-storage/favorite-forest
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON user storage data object

8.5. Delete

To delete user storage data send a DELETE specifying the user storage key in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/user-storage/favorite-forest
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

9. Project templates

9.1. List

To list project templates send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/project-templates
```

The HTTP response is a 200 OK and the response body is a JSON list of project template detail objects

9.2. Create

To create project templates send a POST request with the following data:

- **name** (required): string
- **slug**: slug
- **description** (required): string
- **default_owner_role** (required):
- **is_backlog_activated**: boolean
- **is_kanban_activated**: boolean
- **is_wiki_activated**: boolean
- **is_issues_activated**: boolean
- **videoconferences**: ("whereby-com" | "jitsi" | "talky" | "custom")
- **videoconferences_extra_data**: string
- **default_options**: a json with a list of objects with:
  - **points**: slug
  - **us_status**: slug
  - **task_status**: slug
  - **issue_status**: slug
  - **issue_type**: slug
  - **priority**: slug
  - **severity**: slug
- **us_statuses**: a json with a list of objects with:
  - **name**: string
  - **slug**: slug
  - **is_closed**: boolean

    - **color**: #rgb color
    - **wip_limit**: integer or none
    - **order**: integer
- **points**: a json with a list of objects with:
    - **name**: string
    - **value**: integer or none
    - **order**: integer
- **task_statuses**: a json with a list of objects with:
    - **name**: string
    - **slug**: slug
    - **is_closed**: boolean
    - **color**: #rgb color
    - **order**: integer
- **issue_statuses**: a json with a list of objects with:
    - **name**: string
    - **slug**: slug
    - **is_closed**: boolean
    - **color**: #rgb color
    - **order**: integer
- **issue_types**: a json with a list of objects with:
    - **name**: string
    - **color**: #rgb color
    - **order**: integer
- **priorities**: a json with a list of objects with:
    - **name**: string
    - **color**: #rgb color
    - **order**: integer
- **severities**: a json with a list of objects with:
    - **name**: string
    - **color**: #rgb color
    - **order**: integer
- **roles**: a json with a list of objects with:
    - **name**: string
    - **slug**: slug
    - **permissions**: list of permissions

- **order**: integer
- **computable**: boolean

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${ADMIN_AUTH_TOKEN}" \
-d '{
        "default_options": {
            "issue_status": "New",
            "issue_type": "Bug",
            "points": "?",
            "priority": "Normal",
            "severity": "Normal",
            "task_status": "New",
            "us_status": "New"
        },
        "default_owner_role": "product-owner",
        "description": "Sample description",
        "id": 2,
        "is_backlog_activated": false,
        "is_issues_activated": false,
        "is_kanban_activated": true,
        "is_wiki_activated": false,
        "issue_statuses": [
            {
                "color": "#999999",
                "is_closed": false,
                "name": "New",
                "order": 1
            },
            {
                "color": "#729fcf",
                "is_closed": false,
                "name": "In progress",
                "order": 2
            },
            {
                "color": "#f57900",
                "is_closed": true,
                "name": "Ready for test",
                "order": 3
            },
            {
                "color": "#4e9a06",
                "is_closed": true,
                "name": "Closed",
                "order": 4
            },
            {
                "color": "#cc0000",
```

```
                "is_closed": false,
                "name": "Needs Info",
                "order": 5
            },
            {
                "color": "#d3d7cf",
                "is_closed": true,
                "name": "Rejected",
                "order": 6
            },
            {
                "color": "#75507b",
                "is_closed": false,
                "name": "Postponed",
                "order": 7
            }
        ],
        "issue_types": [
            {
                "color": "#cc0000",
                "name": "Bug",
                "order": 1
            },
            {
                "color": "#729fcf",
                "name": "Question",
                "order": 2
            },
            {
                "color": "#4e9a06",
                "name": "Enhancement",
                "order": 3
            }
        ],
        "name": "New Template",
        "points": [
            {
                "name": "?",
                "order": 1,
                "value": null
            },
            {
                "name": "0",
                "order": 2,
                "value": 0.0
            },
            {
                "name": "1/2",
                "order": 3,
                "value": 0.5
            },
```

```
            {
                "name": "1",
                "order": 4,
                "value": 1.0
            },
            {
                "name": "2",
                "order": 5,
                "value": 2.0
            },
            {
                "name": "3",
                "order": 6,
                "value": 3.0
            },
            {
                "name": "5",
                "order": 7,
                "value": 5.0
            },
            {
                "name": "8",
                "order": 8,
                "value": 8.0
            },
            {
                "name": "10",
                "order": 9,
                "value": 10.0
            },
            {
                "name": "15",
                "order": 10,
                "value": 15.0
            },
            {
                "name": "20",
                "order": 11,
                "value": 20.0
            },
            {
                "name": "40",
                "order": 12,
                "value": 40.0
            }
        ],
        "priorities": [
            {
                "color": "#999999",
                "name": "Low",
                "order": 1
```

```
            },
            {
                "color": "#4e9a06",
                "name": "Normal",
                "order": 3
            },
            {
                "color": "#CC0000",
                "name": "High",
                "order": 5
            }
        ],
        "roles": [
            {
                "computable": true,
                "name": "UX",
                "order": 10,
                "permissions": [
                    "add_issue",
                    "modify_issue",
                    "comment_issue",
                    "delete_issue",
                    "view_issues",
                    "add_milestone",
                    "modify_milestone",
                    "delete_milestone",
                    "view_milestones",
                    "view_project",
                    "add_task",
                    "modify_task",
                    "comment_task",
                    "delete_task",
                    "view_tasks",
                    "add_us",
                    "modify_us",
                    "comment_us",
                    "delete_us",
                    "view_us",
                    "add_wiki_page",
                    "modify_wiki_page",
                    "comment_wiki_page",
                    "delete_wiki_page",
                    "view_wiki_pages",
                    "add_wiki_link",
                    "delete_wiki_link",
                    "view_wiki_links"
                ],
                "slug": "ux"
            },
            {
                "computable": true,
```

```
            "name": "Design",
            "order": 20,
            "permissions": [
                "add_issue",
                "modify_issue",
                "comment_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "comment_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "comment_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "comment_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links"
            ],
            "slug": "design"
        },
        {
            "computable": true,
            "name": "Front",
            "order": 30,
            "permissions": [
                "add_issue",
                "modify_issue",
                "comment_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
```

```
                "comment_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "comment_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "comment_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links"
            ],
            "slug": "front"
        },
        {
            "computable": true,
            "name": "Back",
            "order": 40,
            "permissions": [
                "add_issue",
                "modify_issue",
                "comment_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "comment_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "comment_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "comment_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
```

```
                "view_wiki_links"
            ],
            "slug": "back"
        },
        {
            "computable": false,
            "name": "Product Owner",
            "order": 50,
            "permissions": [
                "add_issue",
                "modify_issue",
                "comment_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "comment_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "comment_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "comment_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links"
            ],
            "slug": "product-owner"
        },
        {
            "computable": false,
            "name": "Stakeholder",
            "order": 60,
            "permissions": [
                "add_issue",
                "modify_issue",
                "comment_issue",
                "delete_issue",
                "view_issues",
                "view_milestones",
```

```
                "view_project",
                "view_tasks",
                "view_us",
                "modify_wiki_page",
                "comment_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links"
            ],
            "slug": "stakeholder"
        }
    ],
    "severities": [
        {
            "color": "#999999",
            "name": "Wishlist",
            "order": 1
        },
        {
            "color": "#729fcf",
            "name": "Minor",
            "order": 2
        },
        {
            "color": "#4e9a06",
            "name": "Normal",
            "order": 3
        },
        {
            "color": "#f57900",
            "name": "Important",
            "order": 4
        },
        {
            "color": "#CC0000",
            "name": "Critical",
            "order": 5
        }
    ],
    "slug": "new-template",
    "task_statuses": [
        {
            "color": "#999999",
            "is_closed": false,
            "name": "New",
            "order": 1
        },
        {
            "color": "#729fcf",
            "is_closed": false,
```

```
                "name": "In progress",
                "order": 2
            },
            {
                "color": "#f57900",
                "is_closed": true,
                "name": "Ready for test",
                "order": 3
            },
            {
                "color": "#4e9a06",
                "is_closed": true,
                "name": "Closed",
                "order": 4
            },
            {
                "color": "#cc0000",
                "is_closed": false,
                "name": "Needs Info",
                "order": 5
            }
        ],
        "us_statuses": [
            {
                "color": "#999999",
                "is_closed": false,
                "name": "New",
                "order": 1,
                "wip_limit": null
            },
            {
                "color": "#f57900",
                "is_closed": false,
                "name": "Ready",
                "order": 2,
                "wip_limit": null
            },
            {
                "color": "#729fcf",
                "is_closed": false,
                "name": "In progress",
                "order": 3,
                "wip_limit": null
            },
            {
                "color": "#4e9a06",
                "is_closed": false,
                "name": "Ready for test",
                "order": 4,
                "wip_limit": null
            },
```

```
            {
                "color": "#cc0000",
                "is_closed": true,
                "name": "Done",
                "order": 5,
                "wip_limit": null
            }
        ],
        "videoconferences": null,
        "videoconferences_extra_data": ""
    }' \
-s http://localhost:8000/api/v1/project-templates
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${ADMIN_AUTH_TOKEN}" \
-d '{
        "default_owner_role": "product-owner",
        "description": "Sample description",
        "name": "New simple template"
    }' \
-s http://localhost:8000/api/v1/project-templates
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON project template detail object

9.3. Get

To get a project template send a GET request specifying the project template id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/project-templates/1
```

The HTTP response is a 200 OK and the response body is a JSON project template detail object

9.4. Edit

To edit project templates send a PUT or a PATCH specifying the project template id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${ADMIN_AUTH_TOKEN}" \
-d '{
        "description": "New description"
    }' \
-s http://localhost:8000/api/v1/project-templates/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON project template detail object

9.5. Delete

To delete project template send a DELETE specifying the project template id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${ADMIN_AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/project-templates/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

10. Projects

10.1. List

To list projects send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects
```

The HTTP response is a 200 OK and the response body is a JSON list of project list entry objects

The results can be filtered using the following parameters:

- **member**: member id
- **members**: member ids
- **is_looking_for_people**: the project is looking for new members
- **is_featured**: the project has been highlighted by the instance staff
- **is_backlog_activated**: backlog is active
- **is_kanban_activated**: kanban is active

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects?member=1
```

The results can be ordered using the order_by parameter with the values:

- **memberships__user_order**: the project order specified by the user
- **total_fans**: total fans for the project
- **total_fans_last_week**: number of new fans in the last week
- **total_fans_last_month**: number of new fans in the last month
- **total_fans_last_year**: number of new fans in the last year
- **total_activity**: number of history entries for the project
- **total_activity_last_week**: number of history entries generated in the last week
- **total_activity_last_month**: number of history entries generated in the last month
- **total_activity_last_year**: number of history entries generated in the last year

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects?member=1\&order_by=memberships__user_order
```

10.2. Create

To create projects send a POST request with the following data:

- **name** (required)
- **description** (required)
- **creation_template**: base template for the project
- **is_backlog_activated**
- **is_issues_activated**
- **is_kanban_activated**
- **is_private**
- **is_wiki_activated**
- **videoconferences**: "whereby-com", "jitsi", "talky" or "custom", the third party used for meetups if enabled
- **videoconferences_extra_data**: string used for the videoconference chat url generation
- **total_milestones**
- **total_story_points**

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Beta description",
        "name": "Beta project"
    }' \
-s http://localhost:8000/api/v1/projects
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "creation_template": 1,
        "description": "Taiga",
        "is_backlog_activated": false,
        "is_issues_activated": true,
        "is_kanban_activated": true,
        "is_private": false,
        "is_wiki_activated": true,
        "name": "Beta project",
        "total_milestones": 3,
        "total_story_points": 20.0,
        "videoconferences": "jitsi",
        "videoconferences_extra_data": null
    }' \
-s http://localhost:8000/api/v1/projects
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON project detail object

10.3. Get

To get a project send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1
```

The HTTP response is a 200 OK and the response body is a JSON project detail object

10.4. Get by slug

To get a project send a GET request specifying the project slug as parameter:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/by_slug?slug=project-0
```

The HTTP response is a 200 OK and the response body is a JSON project detail object

10.5. Edit

To edit projects send a PUT or a PATCH specifying the project id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PUT \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Beta description",
        "name": "Beta project put"
    }' \
-s http://localhost:8000/api/v1/projects/1
```

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Beta project patch"
    }' \
-s http://localhost:8000/api/v1/projects/1
```

When the edit is successful, the HTTP response is a 200 OK and the response body is a JSON project detail object

10.6. Delete

To delete projects send a DELETE specifying the project id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

10.7. Bulk update order

To update the projects order for the logged in user send a POST request with a json list where each element is a json object with two attributes, the project id and the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '[
        {
            "order": 10,
            "project_id": 1
        },
        {
            "order": 15,
            "project_id": 2
        }
    ]' \
-s http://localhost:8000/api/v1/projects/bulk_update_order
```

When the update is successful, the HTTP response is a 200 OK and the response body is empty

10.8. Get modules configuration

To get a project modules configuration send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/modules
```

The HTTP response is a 200 OK and the response body is a JSON project modules configuration object

10.9. Edit modules configuration

To edit a project modules configuration send a PATCH specifying the project id in the url.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "github": {
            "secret": "new_secret"
        }
    }' \
-s http://localhost:8000/api/v1/projects/1/modules
```

When edition succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

10.10. Stats

To get a project stats send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/stats
```

The HTTP response is a 200 OK and the response body is a JSON project stats object

10.11. Issue stats

To get a project issue stats send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/issues_stats
```

The HTTP response is a 200 OK and the response body is a JSON project issue stats object

10.12. Tag colors

To get a project tag colors stats send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/tags_colors
```

The HTTP response is a 200 OK and the response body is a JSON project tag colors object

10.13. Create tag

To create tags send a POST request specifying the project id in the url with the following data:

- **tag** (required)
- **color**: HEX color

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#FC8EAC",
        "tag": "testing-tag"
    }' \
-s http://localhost:8000/api/v1/projects/1/create_tag
```

When the creation is successful, the HTTP response is a 200 OK with an empty body response

10.14. Edit tag

To edit a tag send a POST specifying the project id in the url.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#FFF8E7",
        "from_tag": "testing-tag",
        "to_tag": "testing-tag-updated"
    }' \
-s http://localhost:8000/api/v1/projects/1/edit_tag
```

When the edit is successful, the HTTP response is a 200 OK with an empty body response

10.15. Delete-tag

To delete a tag send a POST specifying the project id in the url.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "tag": "testing-tag-updated"
    }' \
-s http://localhost:8000/api/v1/projects/1/delete_tag
```

When the edit is successful, the HTTP response is a 200 OK with an empty body response

10.16. Mix tags

To mix tags send a POST specifying the project id in the url.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "from_tags": [
            "cum",
            "ut",
            "adipisci",
            "voluptatibus"
        ],
        "to_tag": "cum"
    }' \
-s http://localhost:8000/api/v1/projects/1/mix_tags
```

When the edit is successful, the HTTP response is a 200 OK with an empty body response

10.17. Like a project

To like a project send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/like
```

The HTTP response is a 200 OK with an empty body response

10.18. Unlike a project

To unlike a project send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/unlike
```

The HTTP response is a 200 OK with an empty body response

10.19. List project fans

To get the list of fans from a project send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/fans
```

The HTTP response is a 200 OK and the response body is a JSON list of project voter object

10.20. Watch a project

To watch a project send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "notify_level": 3
    }' \
-s http://localhost:8000/api/v1/projects/1/watch
```

The HTTP response is a 200 OK with an empty body response

10.21. Stop watching project

To stop watching a project send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/unwatch
```

The HTTP response is a 200 OK with an empty body response

10.22. List project watchers

To get the list of watchers from a project send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/watchers
```

The HTTP response is a 200 OK and the response body is a JSON list of project watcher object

10.23. Create template

To create a template from a selected project send a POST request specifying the project id in the url with the following parameters:

- **name** (required)
- **description** (required)

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${ADMIN_AUTH_TOKEN}" \
-d '{
        "template_description": "Beta template description",
        "template_name": "Beta template"
    }' \
-s http://localhost:8000/api/v1/projects/1/create_template
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON project template detail object

10.24. Leave

To leave a project send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/2/leave
```

The HTTP response is a 200 OK with an empty body response

10.25. Change logo

To change your project logo send a POST with the following data

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F logo=@test.png \
-s http://localhost:8000/api/v1/projects/1/change_logo
```

When the change is successful, the HTTP response is a 200 OK and the response body is a JSON project detail object

10.26. Remove logo

To remove your project logo send a POST with the following data

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/remove_logo
```

When the change is successful, the HTTP response is a 200 OK and the response body is a JSON project detail object

10.27. Transfer validate-token

To check if a transfer token for one project is valid for your user send a POST request specifying the project id in the url and containing the following data:

- **token**: valid transfer token received by email.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "token": "6:1jrHFD:8NuXY5qtgY406k-oQrs_o9KMu-s"
    }' \
-s http://localhost:8000/api/v1/projects/1/transfer_validate_token
```

The HTTP response is a 200 OK with an empty body response

10.28. Transfer request

To request to the owner the transfer of a project send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/projects/1/transfer_request
```

The HTTP response is a 200 OK with an empty body response

10.29. Transfer start

To start the transfer of one of your projects to another user send a POST request specifying the project id in the url and containing the following data:

- **user**: user id of other admin member of the project.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "user": 5
    }' \
-s http://localhost:8000/api/v1/projects/1/transfer_start
```

The HTTP response is a 200 OK with an empty body response

10.30. Transfer accept

To accept the transfer of one project to your user send a POST request specifying the project id in the url and containing the following data:

- **token**: valid transfer token received by email.
- **reason**: text included in the email response to the project owner.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "reason": "testing",
        "token": "6:1jrHFD:8NuXY5qtgY406k-oQrs_o9KMu-s"
    }' \
-s http://localhost:8000/api/v1/projects/3/transfer_accept
```

The HTTP response is a 200 OK with an empty body response

10.31. Transfer reject

To reject the transfer of one project to your user send a POST request specifying the project id in the url and containing the following data:

- **token**: valid transfer token received by email.
- **reason**: text included in the email response to the project owner.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "reason": "testing",
        "token": "6:1jrHFD:8NuXY5qtgY406k-oQrs_o9KMu-s"
    }' \
-s http://localhost:8000/api/v1/projects/1/transfer_reject
```

The HTTP response is a 200 OK with an empty body response

10.32. Duplicate

To duplicate a project (create a new one with the same status, colors, attributes...) send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "c description",
        "is_private": true,
        "name": "Dup name",
        "users": [
            {
                "id": 8
            }
        ]
    }' \
-s http://localhost:8000/api/v1/projects/3/duplicate
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON project detail object

11. Memberships/Invitations

11.1. List

To list memberships send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/memberships
```

The HTTP response is a 200 OK and the response body is a JSON list of membership detail objects

The results can be filtered using the following parameters:

- **project**: project id
- **role**: role id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/memberships?project=1
```

11.2. Create

To create memberships/invitations send a POST request with the following data:

- **project** (required)
- **role** (required): Role to the membership
- **username** (required): user username or email

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": 1,
        "role": 3,
        "username": "test-user@email-test.com"
    }' \
-s http://localhost:8000/api/v1/memberships
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON membership detail object

11.3. Bulk creation

To create multiple memberships at the same time send a POST request with the following data:

- **project_id** (required)
- **bulk_memberships** (required): a list of dicts with
  - **role_id**
  - **username**: user username or email
- **invitation_extra_text**: string

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_memberships": [
            {
                "role_id": 3,
                "username": "test@test.com"
            },
            {
                "role_id": 4,
                "username": "john@doe.com"
            }
        ],
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/memberships/bulk_create
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON list of membership detail object

11.4. Get

To get a membership send a GET request specifying the membership id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/memberships/1
```

The HTTP response is a 200 OK and the response body is a JSON membership detail object

11.5. Edit

To edit memberships send a PUT or a PATCH specifying the membership id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "role": 3
    }' \
-s http://localhost:8000/api/v1/memberships/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON

membership detail object

11.6. Delete

To delete memberships/invitations send a DELETE specifying the membership id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/memberships/2
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

11.7. Resend invitation

To resend an invitation send a POST request specifying the membership id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/memberships/1/resend_invitation
```

The HTTP response is a 200 OK and the response body is a JSON membership detail object

11.8. Get Invitation (by token)

To get an invitation send a GET request specifying the invitation id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/invitations/00000000-0000-0000-0000-000000000000
```

The HTTP response is a 200 OK and the response body is a JSON membership detail object

12. Roles

12.1. List

To list roles send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/roles
```

The HTTP response is a 200 OK and the response body is a JSON list of role detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/roles?project=1
```

12.2. Create

To create roles send a POST request with the following data:

- **name** (required)
- **order**: integer
- **project**: (required): project id
- **computable**: `true` if this role has estimations
- **permissions**: list of permmissions (strings) allowed for this role

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New role",
        "order": 10,
        "permissions": [
            "view_us",
            "view_project"
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/roles
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New role name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/roles
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON role detail object

12.3. Get

To get a role send a GET request specifying the role id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/roles/1
```

The HTTP response is a 200 OK and the response body is a JSON role detail object

12.4. Edit

To edit roles send a PUT or a PATCH specifying the role id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch name"
    }' \
-s http://localhost:8000/api/v1/roles/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON role detail object

12.5. Delete

To delete roles send a DELETE specifying the role id in the url. You can use `moveTo` as query param with the new role id for the users that have this one.

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/roles/6/?moveTo=1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

13. Milestones

13.1. List

To list milestones send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones
```

The HTTP response is a 200 OK and the response body is a JSON list of milestone detail objects

The results can be filtered using the following parameters:

- **project**: project ID
- **closed**: `true` to get only closed milestones or `false` to get only opened ones.

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones?project=1
```

NOTE

When you filter milestones by project ID (`/api/v1/milestones?project=<projectID>`) the response has two new headers:

- `Taiga-Info-Total-Opened-Milestones`: the numer of opened milestones for this project.
- `Taiga-Info-Total-Closed-Milestones`: the numer of closed milestones for this project.

13.2. Create

To create milestone send a POST request with the following data:

- **project** (required): project id

- **name** (required): string
- **estimated_start** (required): iso date (YYYY-MM-DD)
- **estimated_finish** (required): iso date (YYYY-MM-DD)
- **disponibility**: float
- **slug**: slug
- **order**: integer
- **watchers**: array of watcher id's

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "disponibility": 30,
        "estimated_finish": "2014-11-04",
        "estimated_start": "2014-10-20",
        "name": "Sprint 1",
        "order": 1,
        "project": 1,
        "slug": "sprint-1",
        "watchers": []
    }' \
-s http://localhost:8000/api/v1/milestones
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "estimated_finish": "2014-11-04",
        "estimated_start": "2014-10-20",
        "name": "Sprint 3",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/milestones
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON milestone detail object

13.3. Get

To get a milestone send a GET request specifying the milestone id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones/1
```

The HTTP response is a 200 OK and the response body is a JSON milestone detail object

13.4. Edit

To edit milestones send a PUT or a PATCH specifying the milestone id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Sprint 2"
    }' \
-s http://localhost:8000/api/v1/milestones/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON milestone detail object

13.5. Delete

To delete milestones send a DELETE specifying the milestone id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

13.6. Stats

To get the milestone stats send a GET request specifying the milestone id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones/1/stats
```

The HTTP response is a 200 OK and the response body is a JSON milestone stats detail object

13.7. Watch a milestone

To watch a milestone send a POST request specifying the milestone id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones/1/watch
```

The HTTP response is a 200 OK with an empty body response

13.8. Stop watching a milestone

To stop watching an milestone send a POST request specifying the milestone id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones/1/unwatch
```

The HTTP response is a 200 OK with an empty body response

13.9. List milestone watchers

To get the list of watchers from a milestone send a GET request specifying the milestone id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/milestones/1/watchers
```

The HTTP response is a 200 OK and the response body is a JSON list of milestone watcher object

14. Epics

14.1. List

To list epics send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics
```

The HTTP response is a 200 OK and the response body is a JSON list of epic list objects

The results can be filtered using the following parameters:

- **project**: project id
- **project__slug**: project slug
- **assigned_to**: assigned to user id
- **status__is_closed**: boolean indicating if the epic status is closed

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics?project=1
```

14.2. Create

To create epics send a POST request with the following data:

- **assigned_to**: user id
- **blocked_note**: reason why the epic is blocked
- **description**: string
- **is_blocked**: boolean
- **is_closed**: boolean
- **color**: HEX color
- **project** (required): project id
- **subject** (required)
- **tags**: array of strings
- **watchers**: array of watcher id's

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "assigned_to": null,
        "blocked_note": "blocking reason",
        "client_requirement": false,
        "color": "#ABCABC",
        "description": "New epic description",
        "epics_order": 2,
        "is_blocked": true,
        "project": 1,
        "status": 2,
        "subject": "New epic",
        "tags": [
            "service catalog",
            "customer"
        ],
        "team_requirement": false,
        "watchers": []
    }' \
-s http://localhost:8000/api/v1/epics
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": 1,
        "subject": "New epic"
    }' \
-s http://localhost:8000/api/v1/epics
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON epic detail object

14.3. Get

To get an epic send a GET request specifying the epic id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/1
```

The HTTP response is a 200 OK and the response body is a JSON epic detail (GET) object

14.4. Get by ref

To get an epic send a GET request specifying the epic reference and one of the following parameters:

- project (project id)
- project__slug

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/by_ref?ref=121\&project=3
```

The HTTP response is a 200 OK and the response body is a JSON epic detail (GET) object

14.5. Edit

To edit epics send a PUT or a PATCH specifying the epic id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject",
        "version": 1
    }' \
-s http://localhost:8000/api/v1/epics/15
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON epic detail object

14.6. Delete

To delete epics send a DELETE specifying the epic id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

14.7. Bulk creation

To create multiple epics at the same time send a POST request with the following data:

- **project_id** (required)
- **status_id** (optional)
- **bulk_epics**: epic subjects, one per line

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_epics": "EPIC 1 \n EPIC 2 \n EPIC 3",
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/epics/bulk_create
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON list of epic detail object

14.8. Filters data

To get the epic filters data send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/filters_data?project=1
```

The HTTP response is a 200 OK and the response body is a JSON epic filters data object

14.9. List related userstories

To get the list of related user stories from an epic send a GET request specifying the epic id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15/related_userstories
```

The HTTP response is a 200 OK and the response body is a JSON list of epic related user story detail objects

14.10. Create related userstory

To create an epic related user story send a POST request with the following data:

- **epic**: related epic id
- **user_story**: related user story id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "epic": 15,
        "user_story": 1
    }' \
-s http://localhost:8000/api/v1/epics/15/related_userstories
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON epic related user story detail object

14.11. Get related userstory

To get a related user story from an epic send a GET request specifying the epic and user story ids in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15/related_userstories/2
```

The HTTP response is a 200 OK and the response body is a JSON epic related user story detail object

14.12. Edit related userstory

To edit epic related user stories send a PUT or a PATCH specifying the epic and user story ids in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "order": 100
    }' \
-s http://localhost:8000/api/v1/epics/15/related_userstories/2
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON epic related user story detail object

14.13. Delete related userstory

To delete epic related user stories send a DELETE specifying the epic and the userstory ids in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15/related_userstories/2
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

14.14. Bulk related userstories creation

To create multiple related user stories at the same time send a POST request with the following data:

- **project_id** (required)
- **bulk_userstories**: user stories subjects, one per line

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_userstories": "epic 1 \n epic 2 \n epic 3",
        "project_id": 3
    }' \
-s http://localhost:8000/api/v1/epics/15/related_userstories/bulk_create
```

When the creation is successful, the HTTP response is a 201 OK and the response body is a JSON list of epic related user story detail object

14.15. Vote an epic

To vote epics send a POST request specifying the epic id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/3/upvote
```

The HTTP response is a 200 OK with an empty body response

14.16. Remove vote from an epic

To remove a vote from an epic send a POST request specifying the epic id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/3/downvote
```

When remove of the vote succeeded, the HTTP response is a 200 OK with an empty body response

14.17. Get epic voters list

To get the list of voters from an epic send a GET request specifying the epic id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/1/voters
```

The HTTP response is a 200 OK and the response body is a JSON list of epic voter object

14.18. Watch an epic

To watch an epic send a POST request specifying the epic id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15/watch
```

The HTTP response is a 200 OK with an empty body response

14.19. Stop watching an epic

To stop watching an epic send a POST request specifying the epic id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15/unwatch
```

The HTTP response is a 200 OK with an empty body response

14.20. List epic watchers

To get the list of watchers from an epic send a GET request specifying the epic id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/15/watchers
```

The HTTP response is a 200 OK and the response body is a JSON list of epic watcher object

14.21. List attachments

To list epic attachments send a GET request with the following parameters:

- **project**: project id
- **object_id**: epic id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/attachments?object_id=773\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON list of attachment detail objects

14.22. Create attachment

To create epic attachments send a POST request with the following data:

- **object_id** (required): epic id
- **project** (required): project id
- **attached_file** (required): attaching file
- **description**
- **is_deprecated**

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F attached_file=@test.png \
-F from_comment=False \
-F object_id=15 \
-F project=3 \
-s http://localhost:8000/api/v1/epics/attachments
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON attachment detail object

14.23. Get attachment

To get an epic attachment send a GET request specifying the epic attachment id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/attachments/773
```

The HTTP response is a 200 OK and the response body is a JSON attachment detail object

14.24. Edit attachment

To edit epic attachments send a PUT or a PATCH specifying the epic attachment id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Updated description"
    }' \
-s http://localhost:8000/api/v1/epics/attachments/773
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON attachment detail object

14.25. Delete attachment

To delete epic attachments send a DELETE specifying the epic attachment id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/attachments/773
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

15. Epic status

15.1. List

To list epic status send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-statuses
```

The HTTP response is a 200 OK and the response body is a JSON list of epic status detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-statuses?project=1
```

15.2. Create

To create epic statuses send a POST request with the following data:

- **color**: in hexadecimal
- **is_closed**: (true|false)
- **name** (required)
- **order**: integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "is_closed": true,
        "name": "New status",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/epic-statuses
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New status name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/epic-statuses
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON epic status detail object

15.3. Get

To get an epic status send a GET request specifying the epic status id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-statuses/1
```

The HTTP response is a 200 OK and the response body is a JSON epic status detail object

15.4. Edit

To edit epic statuses send a PUT or a PATCH specifying the epic status id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch status name"
    }' \
-s http://localhost:8000/api/v1/epic-statuses/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON epic status detail object

15.5. Delete

To delete epic satuses send a DELETE specifying the epic status id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-statuses/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

15.6. Bulk update order

To update the order of multiple epic statuses at the same time send a POST request with the following data:

- **project** (required)
- **bulk_epic_statuses**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_epic_statuses": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/epic-statuses/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

16. Epic custom attribute

16.1. List

To list epic custom attributes send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-custom-attributes
```

The HTTP response is a 200 OK and the response body is a JSON list of epic custom attribute detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-custom-attributes?project=1
```

16.2. Create

To create epic custom attributes send a POST request with the following data:

- **name**: (required) text
- **description**: text
- **order**: integer
- **project**: (required) integer, project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Duration in minutes",
        "name": "Duration 2",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/epic-custom-attributes
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 3",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/epic-custom-attributes
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON epic custom attribute detail object

16.3. Get

To get an epic custom attribute send a GET request specifying the epic custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-custom-attributes/14
```

The HTTP response is a 200 OK and the response body is a JSON epic custom attribute detail object

16.4. Edit

To edit epic custom attributes send a PUT or a PATCH specifying the epic custom attribute id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 1"
    }' \
-s http://localhost:8000/api/v1/epic-custom-attributes/14
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON epic custom attribute detail object

16.5. Delete

To delete epic custom attributes send a DELETE specifying the epic custom attribute id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epic-custom-attributes/14
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

16.6. Bulk update order

To update the order of multiple epic custom attributes at the same time send a POST request with the following data:

- **project** (required)
- **bulk_epic_custom_attributes**: list where each element is a list, the first element is the custom attribute id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_epic_custom_attributes": [
            [
                14,
                10
            ],
            [
                13,
                15
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/epic-custom-attributes/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

17. Epic custom attributes values

17.1. Get

To get an epic custom attribute value send a GET request specifying the epic custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/epics/custom-attributes-values/15
```

The HTTP response is a 200 OK and the response body is a JSON epic custom attribute detail object

17.2. Edit

To edit epic custom attributes values send a PUT or a PATCH specifying the epic id in the url. "attribute_values" must be a JSON object with pairs epic custom atribute id - value. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "attributes_values": {
            "14": "240 min"
        },
        "version": 1
    }' \
-s http://localhost:8000/api/v1/epics/custom-attributes-values/15
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON epic custom attribute detail object

18. User stories

18.1. List

To list user stories send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories
```

The HTTP response is a 200 OK and the response body is a JSON list of user story list objects

The results can be filtered using the following parameters:

- **project**: project id
- **milestone**: milestone id

- **milestone__isnull**: (true|false) if you are looking for user stories associated with a milestone or not
- **status**: status id
- **status__is_archived**: (true|false)
- **tags**: separated by ","
- **watchers**: watching user id
- **assigned_to**: assigned to user id
- **epic**: epic id
- **role**: role id
- **status__is_closed**: (true|false)
- **exclude_status**: status id
- **exclude_tags**: separated by ","
- **exclude_assigned_to**: assigned to user id
- **exclude role**: role id
- **exclude epic**: epic id

```
the "exclude_" params work excluding from the response the results with which they
match.
```

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories?project=1
```

18.2. Create

To create user stories send a POST request with the following data:

- **assigned_to**: user id
- **backlog_order**: order in the backlog
- **blocked_note**: reason why the user story is blocked
- **client_requirement**: boolean
- **description**: string
- **is_blocked**: boolean
- **is_closed**: boolean
- **kanban_order**: order in the kanban
- **milestone**: milestone id

- **points**: dictionary of points
- **project** (required): project id
- **sprint_order**: order in the milestone
- **status**: status id
- **subject** (required)
- **tags**: array of strings
- **team_requirement**: boolean
- **watchers**: array of watcher id's

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "assigned_to": null,
        "backlog_order": 2,
        "blocked_note": "blocking reason",
        "client_requirement": false,
        "description": "Implement API CALL",
        "is_blocked": false,
        "is_closed": true,
        "kanban_order": 37,
        "milestone": null,
        "points": {
            "1": 4,
            "2": 3,
            "3": 2,
            "4": 1
        },
        "project": 1,
        "sprint_order": 2,
        "status": 2,
        "subject": "Customer personal data",
        "tags": [
            "service catalog",
            "customer"
        ],
        "team_requirement": false,
        "watchers": []
    }' \
-s http://localhost:8000/api/v1/userstories
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": 1,
        "subject": "Customer personal data"
    }' \
-s http://localhost:8000/api/v1/userstories
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON user story detail object

18.3. Get

To get a user story send a GET request specifying the user story id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/1
```

The HTTP response is a 200 OK and the response body is a JSON user story detail (GET) object

18.4. Get by ref

To get a user story send a GET request specifying the user story reference and one of the following parameters:

- project (project id)
- project__slug

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/by_ref?ref=1\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON user story detail (GET) object

18.5. Edit

To edit user stories send a PUT or a PATCH specifying the user story id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject",
        "version": 1
    }' \
-s http://localhost:8000/api/v1/userstories/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON user story detail object

18.6. Delete

To delete user stories send a DELETE specifying the user story id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

18.7. Bulk creation

To create multiple user stories at the same time send a POST request with the following data:

- **project_id** (required)
- **status_id**
- **bulk_stories**: user story subjects, one per line

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_stories": "US 1 \n US 2 \n US 3",
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/userstories/bulk_create
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON list of user story detail object

18.8. Bulk update backlog order

To update the backlog order of multiple user stories at the same time send a POST request with the following data:

- **project_id** (required)
- **bulk_stories**: list where each element is a json object with two attributes, the user story id and the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_stories": [
            {
                "order": 10,
                "us_id": 1
            },
            {
                "order": 15,
                "us_id": 2
            }
        ],
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/userstories/bulk_update_backlog_order
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON list of user story detail object

18.9. Bulk update kanban order

To update the kanban order of multiple user stories at the same time send a POST request with the following data:

- **project_id** (required)
- **bulk_stories**: list where each element is a json object with two attributes, the user story id and the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_stories": [
            {
                "order": 10,
                "us_id": 1
            },
            {
                "order": 15,
                "us_id": 2
            }
        ],
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/userstories/bulk_update_kanban_order
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON list of user story detail object

18.10. Bulk update sprint order

To update the sprint order of multiple user stories at the same time send a POST request with the following data:

- **project_id** (required)
- **bulk_stories**: list where each element is a json object with two attributes, the user story id and the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_stories": [
            {
                "order": 10,
                "us_id": 1
            },
            {
                "order": 15,
                "us_id": 2
            }
        ],
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/userstories/bulk_update_sprint_order
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON list of user story detail object

18.11. Bulk update milestone

To update the sprint of multiple user stories at the same time send a POST request with the following data:

- **project_id** (required)
- **milestone_id** (required)
- **bulk_stories**: list where each element is a json object with two attributes, the user story id and the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_stories": [
            {
                "order": 10,
                "us_id": 1
            },
            {
                "order": 15,
                "us_id": 2
            }
        ],
        "milestone_id": 1,
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/userstories/bulk_update_milestone
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

18.12. Filters data

To get the user stories filters data send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/filters_data?project=1
```

The HTTP response is a 200 OK and the response body is a JSON user story filters data object

18.13. Vote a user story

To add a vote to a user story send a POST request specifying the user story id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/1/upvote
```

The HTTP response is a 200 OK with an empty body response

18.14. Remove vote from a user story

To remove a vote from a user story send a POST request specifying the user story id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/1/downvote
```

When remove of the vote succeeded, the HTTP response is a 200 OK with an empty body response

18.15. Get user story voters list

To get the list of voters from a user story send a GET request specifying the user story id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/2/voters
```

The HTTP response is a 200 OK and the response body is a JSON list of user story voter object

18.16. Watch a user story

To watch a user story send a POST request specifying the project id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/1/watch
```

The HTTP response is a 200 OK with an empty body response

18.17. Stop watching a user story

To stop watching a user story send a POST request specifying the user story id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/1/unwatch
```

The HTTP response is a 200 OK with an empty body response

18.18. List user story watchers

To get the list of watchers from a user story send a GET request specifying the user story id in the url

```
{
    "full_name": "Vanesa Torres",
    "id": 6,
    "username": "user2114747470430251528"
}
```

The HTTP response is a 200 OK and the response body is a JSON list of user story watcher object

18.19. List attachments

To list user story attachments send a GET request with the following parameters:

- **project**: project id
- **object_id**: user story id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/attachments?object_id=1\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON list of attachment detail objects

18.20. Create attachment

To create user story attachments send a POST request with the following data:

- **object_id** (required): user story id
- **project** (required): project id

- **attached_file** (required): attaching file
- **description**
- **is_deprecated**

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F attached_file=@test.png \
-F from_comment=False \
-F object_id=1 \
-F project=1 \
-s http://localhost:8000/api/v1/userstories/attachments
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON attachment detail object

18.21. Get attachment

To get a user story attachment send a GET request specifying the user story attachment id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/attachments/415
```

The HTTP response is a 200 OK and the response body is a JSON attachment detail object

18.22. Edit attachment

To edit user story attachments send a PUT or a PATCH specifying the user story attachment id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "patching description"
    }' \
-s http://localhost:8000/api/v1/userstories/attachments/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON attachment detail object

18.23. Delete attachment

To delete user story attachments send a DELETE specifying the user story attachment id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/attachments/458
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

19. User story status

19.1. List

To list user story status send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-statuses
```

The HTTP response is a 200 OK and the response body is a JSON list of user story status detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-statuses?project=1
```

19.2. Create

To create user story statuses send a POST request with the following data:

- **color**: in hexadecimal
- **is_closed**: (true|false)
- **name** (required)
- **order**: integer
- **project**: (required): project id

- **wip_limit**: integer representing the max number of user stories in this status

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "is_closed": true,
        "name": "New status",
        "order": 8,
        "project": 1,
        "wip_limit": 6
    }' \
-s http://localhost:8000/api/v1/userstory-statuses
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New status name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/userstory-statuses
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON user story status detail object

19.3. Get

To get a user story status send a GET request specifying the user story status id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-statuses/1
```

The HTTP response is a 200 OK and the response body is a JSON user story status detail object

19.4. Edit

To edit user story statuses send a PUT or a PATCH specifying the user story status id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch status name"
    }' \
-s http://localhost:8000/api/v1/userstory-statuses/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON user story status detail object

19.5. Delete

To delete user story satuses send a DELETE specifying the user story status id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-statuses/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

19.6. Bulk update order

To update the order of multiple user story statues at the same time send a POST request with the following data:

- **project** (required)
- **bulk_userstory_statuses**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_userstory_statuses": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/userstory-statuses/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

20. Points

20.1. List

To list points send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/points
```

The HTTP response is a 200 OK and the response body is a JSON list of point detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/points?project=1
```

20.2. Create

To create points send a POST request with the following data:

- **color**: in hexadecimal
- **name** (required)
- **order**: integer
- **value** (required): integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "name": "Huge",
        "order": 8,
        "project": 1,
        "value": 40
    }' \
-s http://localhost:8000/api/v1/points
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Very huge",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/points
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON point detail object

20.3. Get

To get a point send a GET request specifying the point id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/points/1
```

The HTTP response is a 200 OK and the response body is a JSON point detail object

20.4. Edit

To edit points send a PUT or a PATCH specifying the point id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch name"
    }' \
-s http://localhost:8000/api/v1/points/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON point detail object

20.5. Delete

To delete points send a DELETE specifying the point id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/points/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

20.6. Bulk update order

To update the order of multiple points at the same time send a POST request with the following data:

- **project** (required)
- **bulk_points**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_points": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/points/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

21. User story custom attribute

21.1. List

To list user story custom attributes send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-custom-attributes
```

The HTTP response is a 200 OK and the response body is a JSON list of user story custom attribute detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-custom-attributes?project=1
```

21.2. Create

To create user story custom attributes send a POST request with the following data:

- **name**: (required) text
- **description**: text
- **order**: integer
- **project**: (required) integer, project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Duration in minutes",
        "name": "Duration 2",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/userstory-custom-attributes
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 3",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/userstory-custom-attributes
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON user story custom attribute detail object

21.3. Get

To get a user story custom attribute send a GET request specifying the user story custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-custom-attributes/1
```

The HTTP response is a 200 OK and the response body is a JSON user story custom attribute detail object

21.4. Edit

To edit user story custom attributes send a PUT or a PATCH specifying the user story custom attribute id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 1"
    }' \
-s http://localhost:8000/api/v1/userstory-custom-attributes/1
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON user story custom attribute detail object

21.5. Delete

To delete user story custom attributes send a DELETE specifying the user story custom attribute id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstory-custom-attributes/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

21.6. Bulk update order

To update the order of multiple user story custom attributes at the same time send a POST request with the following data:

- **project** (required)
- **bulk_userstory_custom_attributes**: list where each element is a list, the first element is the custom attribute id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_userstory_custom_attributes": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/userstory-custom-attributes/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

22. User story custom attributes values

22.1. Get

To get a user story custom attribute send a GET request specifying the user story custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/userstories/custom-attributes-values/1
```

The HTTP response is a 200 OK and the response body is a JSON user story custom attribute detail object

22.2. Edit

To edit user story custom attributes values send a PUT or a PATCH specifying the user story id in the url. "attribute_values" must be a JSON object with pairs user story custom atribute id - value. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "attributes_values": {
            "3": "240 min"
        },
        "version": 1
    }' \
-s http://localhost:8000/api/v1/userstories/custom-attributes-values/1
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON user story custom attribute detail object

23. Tasks

23.1. List

To list tasks send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks
```

The HTTP response is a 200 OK and the response body is a JSON list of task list objects

The results can be filtered using the following parameters:

- **project**: project id
- **status**: status id
- **tags**: separated by ","
- **user_story**: user story id
- **role**: role id
- **owner**: owner id
- **milestone**: milestone id
- **watchers**: watching user id
- **assigned_to**: assigned to user id
- **status__is_closed**: (true|false)
- **exclude_status**: status id
- **exclude_tags**: separared by ","

- **exclude_role**: role id
- **exclude_owner**: owner id
- **exclude_assigned_to**: assigned to user id

the "exclude_" params work excluding from the response the results with which they match.

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks?project=1
```

23.2. Create

To create tasks send a POST request with the following data:

- **assigned_to**: user id
- **blocked_note**: reason why the task is blocked
- **description**: string
- **is_blocked**: boolean
- **is_closed**: boolean
- **milestone**: milestone id
- **project** (required): project id
- **user_story**: user story id
- **status**: status id
- **subject** (required)
- **tags**: array of strings
- **us_order**: order in the user story,
- **taskboard_order**: order in the taskboard,
- **is_iocaine**: boolean,
- **external_reference**: tuple of ("service", serviceId),
- **watchers**: array of watcher id’s

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "assigned_to": null,
        "blocked_note": "blocking reason",
        "description": "Implement API CALL",
        "external_reference": null,
        "is_blocked": false,
        "is_closed": true,
        "is_iocaine": false,
        "milestone": null,
        "project": 1,
        "status": 1,
        "subject": "Customer personal data",
        "tags": [
            "service catalog",
            "customer"
        ],
        "taskboard_order": 1,
        "us_order": 1,
        "user_story": 17,
        "watchers": []
    }' \
-s http://localhost:8000/api/v1/tasks
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": 1,
        "subject": "Customer personal data"
    }' \
-s http://localhost:8000/api/v1/tasks
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON task detail object

23.3. Get

To get a task send a GET request specifying the task id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1
```

The HTTP response is a 200 OK and the response body is a JSON task detail (GET) object

23.4. Get by ref

To get a task send a GET request specifying the task reference and one of the following parameters:

- project (project id)
- project__slug


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2020-07-02T11:56:21.529Z",
    "description": "Nam veritatis facere debitis vitae animi eos cum suscipit
reprehenderit.",
    "description_html": "<p>Nam veritatis facere debitis vitae animi eos cum suscipit
reprehenderit.</p>",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": "2020-05-10T05:32:33.173Z",
    "generated_user_stories": null,
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": false,
    "is_watcher": true,
    "milestone": 1,
    "milestone_slug": "sprint-2020-5-8",
    "modified_date": "2020-07-03T08:41:01.723Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 3,
            "subject": "Add tests for bulk operations"
```

```
        },
        "previous": null
    },
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Beta project patch",
        "slug": "project-0"
    },
    "ref": 2,
    "status": 3,
    "status_extra_info": {
        "color": "#ffcc00",
        "is_closed": true,
        "name": "Ready for test"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "atque",
            null
        ],
        [
            "animi",
            null
        ],
        [
            "cum",
            null
        ],
        [
            "eveniet",
            null
        ],
        [
            "cumque",
            null
        ],
        [
            "reiciendis",
```

```
            null
        ],
        [
            "architecto",
            null
        ],
        [
            "perspiciatis",
            null
        ]
    ],
    "taskboard_order": 1593690981529,
    "total_comments": 1,
    "total_voters": 6,
    "total_watchers": 3,
    "us_order": 1593690981529,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": [
            {
                "color": "#f57900",
                "id": 15,
                "project": {
                    "id": 3,
                    "name": "Project Example 2",
                    "slug": "project-2"
                },
                "ref": 121,
                "subject": "Patching subject"
            }
        ],
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": [
        8,
        3,
        6
    ]
}
```


The HTTP response is a 200 OK and the response body is a JSON task detail (GET) object

23.5. Edit

To edit tasks send a PUT or a PATCH specifying the task id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject",
        "version": 1
    }' \
-s http://localhost:8000/api/v1/tasks/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON task detail object

23.6. Delete

To delete tasks send a DELETE specifying the task id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

23.7. Bulk creation

To create multiple tasks at the same time send a POST request with the following data:

- **project_id** (required)
- **status_id**
- **sprint_id**: milestone id (optional)
- **us_id**: user story id (optional)
- **bulk_tasks**: task subjects, one per line

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_tasks": "Task 1 \n Task 2 \n Task 3",
        "milestone_id": 1,
        "project_id": 1
    }' \
-s http://localhost:8000/api/v1/tasks/bulk_create
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON list

of task detail object

23.8. Filters data

To get the task filters data send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/filters_data?project=1
```

The HTTP response is a 200 OK and the response body is a JSON task filters data object

23.9. Vote a task

To vote tasks send a POST request specifying the task id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1/upvote
```

The HTTP response is a 200 OK with an empty body response

23.10. Remove vote from a task

To remove a vote from a task send a POST request specifying the task id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1/downvote
```

When remove of the vote succeeded, the HTTP response is a 200 OK with an empty body response

23.11. Get task voters list

To get the list of voters from a task send a GET request specifying the task id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1/voters
```

The HTTP response is a 200 OK and the response body is a JSON list of task voter object

23.12. Watch a task

To watch a task send a POST request specifying the task id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1/watch
```

The HTTP response is a 200 OK with an empty body response

23.13. Stop watching a task

To stop watching a task send a POST request specifying the task id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1/unwatch
```

The HTTP response is a 200 OK with an empty body response

23.14. List task watchers

To get the list of watchers from a task send a GET request specifying the task id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/1/watchers
```

The HTTP response is a 200 OK and the response body is a JSON list of task watcher object

23.15. List attachments

To list task attachments send a GET request with the following parameters:

- **project**: project id
- **object_id**: task id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/attachments?object_id=1\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON list of attachment detail objects

23.16. Create attachment

To create task attachments send a POST request with the following data:

- **object_id** (required): task id
- **project** (required): project id
- **attached_file** (required): attaching file
- **description**
- **is_deprecated**

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F attached_file=@test.png \
-F from_comment=False \
-F object_id=1 \
-F project=1 \
-s http://localhost:8000/api/v1/tasks/attachments
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON attachment detail object

23.17. Get attachment

To get a task attachment send a GET request specifying the task attachment id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/attachments/461
```

The HTTP response is a 200 OK and the response body is a JSON attachment detail object

23.18. Edit attachment

To edit task attachments send a PUT or a PATCH specifying the task attachment id in the url. In a

PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Updated description"
    }' \
-s http://localhost:8000/api/v1/tasks/attachments/461
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON attachment detail object

23.19. Delete attachment

To delete task attachments send a DELETE specifying the task attachment id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/tasks/attachments/461
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

24. Task status

24.1. List

To list task status send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-statuses
```

The HTTP response is a 200 OK and the response body is a JSON list of task status detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-statuses?project=1
```

24.2. Create

To create task statuses send a POST request with the following data:

- **color**: in hexadecimal
- **is_closed**: (true|false)
- **name** (required)
- **order**: integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "is_closed": true,
        "name": "New status",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/task-statuses
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New status name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/task-statuses
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON task status detail object

24.3. Get

To get a task status send a GET request specifying the task status id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-statuses/1
```

The HTTP response is a 200 OK and the response body is a JSON task status detail object

24.4. Edit

To edit task statuses send a PUT or a PATCH specifying the task status id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch status name"
    }' \
-s http://localhost:8000/api/v1/task-statuses/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON task status detail object

24.5. Delete

To delete task satuses send a DELETE specifying the task status id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-statuses/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

24.6. Bulk update order

To update the order of multiple task statuses at the same time send a POST request with the following data:

- **project** (required)
- **bulk_task_statuses**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_task_statuses": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/task-statuses/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

25. Task custom attribute

25.1. List

To list task custom attributes send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-custom-attributes
```

The HTTP response is a 200 OK and the response body is a JSON list of task custom attribute detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-custom-attributes?project=1
```

25.2. Create

To create task custom attributes send a POST request with the following data:

- **name**: (required) text
- **description**: text
- **order**: integer
- **project**: (required) integer, project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Duration in minutes",
        "name": "Duration 2",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/task-custom-attributes
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 3",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/task-custom-attributes
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON task custom attribute detail object

25.3. Get

To get a task custom attribute send a GET request specifying the task custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-custom-attributes/1
```

The HTTP response is a 200 OK and the response body is a JSON task custom attribute detail object

25.4. Edit

To edit task custom attributes send a PUT or a PATCH specifying the task custom attribute id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 1"
    }' \
-s http://localhost:8000/api/v1/task-custom-attributes/1
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON task custom attribute detail object

25.5. Delete

To delete task custom attributes send a DELETE specifying the task custom attribute id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-custom-attributes/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

25.6. Bulk update order

To update the order of multiple task custom attributes at the same time send a POST request with the following data:

- **project** (required)
- **bulk_task_custom_attributes**: list where each element is a list, the first element is the custom attribute id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_task_custom_attributes": [
            [
                1,
                10
            ],
            [
                5,
                15
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/task-custom-attributes/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

26. Task custom attributes values

26.1. Get

To get a task custom attribute send a GET request specifying the task custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/task-custom-attributes/1
```

The HTTP response is a 200 OK and the response body is a JSON task custom attribute detail object

26.2. Edit

To edit task custom attributes values send a PUT or a PATCH specifying the task id in the url. "attribute_values" must be a JSON object with pairs task custom atribute id - value. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 1"
    }' \
-s http://localhost:8000/api/v1/task-custom-attributes/1
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON task custom attribute detail object

27. Issues

27.1. List

To list issues send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues
```

The HTTP response is a 200 OK and the response body is a JSON list of issue detail list objects

The results can be filtered using the following parameters:

- **project**: project id
- **status**: status id
- **severity**: severity id
- **priority**: priority id
- **owner**: owner user id
- **assigned_to**: assigned to user id
- **tags**: separated by ","
- **type**: issue type id
- **role**: role id
- **watchers**: watching user id
- **status__is_closed**: (true|false)
- **exclude_status**: status id
- **exclude_severity**: severity id
- **exclude_priority**: priority id

- **exclude_owner**: user owner id
- **exclude_assigned_to**: assigned to user id
- **exlcude_tags**: separated by ","
- **exclude_type**: issue type id
- **exclude role**: role id

the "exclude_" params work excluding from the response the results with which they match.

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues?project=1
```

The results can be ordered using the order_by parameter with the values:

- **type**
- **severity**
- **status**
- **priority**
- **created_date**
- **modified_date**
- **owner**
- **assigned_to**
- **subject**

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues?project=1\&order_by=priority
```

27.2. Create

To create issues send a POST request with the following data:

- **assigned_to**: user id
- **blocked_note**: reason why the issue is blocked
- **description**: string
- **is_blocked**: boolean
- **is_closed**: boolean
- **milestone**: milestone id

- **project** (required): project id
- **status**: status id
- **severity**: severity id
- **priority**: priority id
- **type**: type id
- **subject** (required)
- **tags**: array of strings
- **watchers**: array of watcher id's

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "assigned_to": null,
        "blocked_note": "blocking reason",
        "description": "Implement API CALL",
        "is_blocked": false,
        "is_closed": true,
        "milestone": null,
        "priority": 3,
        "project": 1,
        "severity": 2,
        "status": 3,
        "subject": "Customer personal data",
        "tags": [
            "service catalog",
            "customer"
        ],
        "type": 1,
        "watchers": []
    }' \
-s http://localhost:8000/api/v1/issues
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": 1,
        "subject": "Customer personal data"
    }' \
-s http://localhost:8000/api/v1/issues
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON issue detail object

27.3. Get

To get an issue send a GET request specifying the issue id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3
```

The HTTP response is a 200 OK and the response body is a JSON issue detail (GET) object

27.4. Get by ref

To get an issue send a GET request specifying the issue reference and one of the following parameters:

- project (project id)
- project__slug

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/by_ref?ref=63\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON issue detail (GET) object

27.5. Edit

To edit issues send a PUT or a PATCH specifying the issue id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject",
        "version": 1
    }' \
-s http://localhost:8000/api/v1/issues/3
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON issue detail object

27.6. Delete

To delete issues send a DELETE specifying the issue id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/22
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

27.7. Filters data

To get the issue filters data send a GET request specifying the project id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/filters_data?project=1
```

The HTTP response is a 200 OK and the response body is a JSON issue filters data object

27.8. Vote an issue

To vote issues send a POST specifying the issue id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3/upvote
```

When vote succeeded, the HTTP response is a 200 OK with an empty body response

27.9. Remove vote from an issue

To remove a vote from an issue send a POST specifying the issue id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3/downvote
```

When remove of the vote succeeded, the HTTP response is a 200 OK with an empty body response

27.10. Get issue voters list

To get the list of voters from an issue send a GET request specifying the issue id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3/voters
```

The HTTP response is a 200 OK and the response body is a JSON issue voters detail object

27.11. Watch an issue

To watch an issue send a POST request specifying the issue id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3/watch
```

The HTTP response is a 200 OK with an empty body response

27.12. Stop watching an issue

To stop watching an issue send a POST request specifying the issue id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3/unwatch
```

The HTTP response is a 200 OK with an empty body response

27.13. List issue watchers

To get the list of watchers from an issue send a GET request specifying the issue id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/3/watchers
```

The HTTP response is a 200 OK and the response body is a JSON list of issue watcher object

27.14. List attachments

To list issue attachments send a GET request with the following parameters:

- **project**: project id
- **object_id**: issue id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/attachments?object_id=747\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON list of attachment detail objects

27.15. Create attachment

To create issue attachments send a POST request with the following data:

- **object_id** (required): issue id
- **project** (required): project id
- **attached_file** (required): attaching file
- **description**
- **is_deprecated**

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F attached_file=@test.png \
-F from_comment=False \
-F object_id=22 \
-F project=1 \
-s http://localhost:8000/api/v1/issues/attachments
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON attachment detail object

27.16. Get attachment

To get an issue attachment send a GET request specifying the issue attachment id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/attachments/747
```

The HTTP response is a 200 OK and the response body is a JSON attachment detail object

27.17. Edit attachment

To edit issue attachments send a PUT or a PATCH specifying the issue attachment id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/attachments/747
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON attachment detail object

27.18. Delete attachment

To delete issue attachments send a DELETE specifying the issue attachment id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/attachments/747
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

28. Issue status

28.1. List

To list issue status send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-statuses
```

The HTTP response is a 200 OK and the response body is a JSON list of issue status detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-statuses?project=1
```

28.2. Create

To create issue statuses send a POST request with the following data:

- **color**: in hexadecimal
- **is_closed**: (true | false)
- **name** (required)
- **order**: integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "is_closed": true,
        "name": "New status",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-statuses
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New status name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-statuses
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON issue status detail object

28.3. Get

To get a issue status send a GET request specifying the issue status id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-statuses/1
```

The HTTP response is a 200 OK and the response body is a JSON issue status detail object

28.4. Edit

To edit issue statuses send a PUT or a PATCH specifying the issue status id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject",
        "version": 1
    }' \
-s http://localhost:8000/api/v1/issues/3
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON issue status detail object

28.5. Delete

To delete issue satuses send a DELETE specifying the issue status id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-statuses/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

28.6. Bulk update order

To update the order of multiple issue statuses at the same time send a POST request with the following data:

- **project** (required)

- **bulk_issue_statuses**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_issue_statuses": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-statuses/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

29. Issue types

29.1. List

To list issue types send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-types
```

The HTTP response is a 200 OK and the response body is a JSON list of issue type detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-types?project=1
```

29.2. Create

To create issue types send a POST request with the following data:

- **color**: in hexadecimal
- **name** (required)
- **order**: integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "name": "New type",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-types
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New type name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-types
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON issue type detail object

29.3. Get

To get an issue type send a GET request specifying the issue type id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-types/1
```

The HTTP response is a 200 OK and the response body is a JSON issue type detail object

29.4. Edit

To edit issue types send a PUT or a PATCH specifying the issue type id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch type name"
    }' \
-s http://localhost:8000/api/v1/issue-types/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON issue type detail object

29.5. Delete

To delete issue statuses send a DELETE specifying the issue type id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-types/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

29.6. Bulk update order

To update the order of multiple issue types at the same time send a POST request with the following data:

- **project** (required)
- **bulk_issue_types**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_issue_types": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-types/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

30. Priorities

30.1. List

To list priorities send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/priorities
```

The HTTP response is a 200 OK and the response body is a JSON list of priority detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/priorities?project=1
```

30.2. Create

To create priorities send a POST request with the following data:

- **color**: in hexadecimal
- **name** (required)
- **order**: integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "name": "New priority",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/priorities
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New priority name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/priorities
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON priority detail object

30.3. Get

To get a priority send a GET request specifying the priority id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/priorities/1
```

The HTTP response is a 200 OK and the response body is a JSON priority detail object

30.4. Edit

To edit priorities send a PUT or a PATCH specifying the priority id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch name"
    }' \
-s http://localhost:8000/api/v1/priorities/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON priority detail object

30.5. Delete

To delete priorities send a DELETE specifying the priority id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/priorities/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

30.6. Bulk update order

To update the order of multiple priorities at the same time send a POST request with the following data:

- **project** (required)
- **bulk_priorities**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_priorities": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/priorities/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

31. Severities

31.1. List

To list severities send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/severities
```

The HTTP response is a 200 OK and the response body is a JSON list of severity detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/severities?project=1
```

31.2. Create

To create severities send a POST request with the following data:

- **color**: in hexadecimal
- **name** (required)
- **order**: integer
- **project**: (required): project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "color": "#AAAAAA",
        "name": "New severity",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/severities
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "New severity name",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/severities
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON severity detail object

31.3. Get

To get a severity send a GET request specifying the severity id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/severities/1
```

The HTTP response is a 200 OK and the response body is a JSON severity detail object

31.4. Edit

To edit severities send a PUT or a PATCH specifying the severity id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Patch name"
    }' \
-s http://localhost:8000/api/v1/severities/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON severity detail object

31.5. Delete

To delete severities send a DELETE specifying the severity id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/severities/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

31.6. Bulk update order

To update the order of multiple severities at the same time send a POST request with the following data:

- **project** (required)
- **bulk_severities**: list where each element is a list, the first element is the status id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_severities": [
            [
                1,
                10
            ],
            [
                2,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/severities/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

32. Issue custom attribute

32.1. List

To list issue custom attributes send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-custom-attributes
```

The HTTP response is a 200 OK and the response body is a JSON list of issue custom attribute detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-custom-attributes?project=1
```

32.2. Create

To create issue custom attributes send a POST request with the following data:

- **name**: (required) text
- **description**: text
- **order**: integer
- **project**: (required) integer, project id

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Duration in minutes",
        "name": "Duration 2",
        "order": 8,
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-custom-attributes
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 3",
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-custom-attributes
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON issue custom attribute detail object

32.3. Get

To get a issue custom attribute send a GET request specifying the issue custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-custom-attributes/5
```

The HTTP response is a 200 OK and the response body is a JSON issue custom attribute detail object

32.4. Edit

To edit issue custom attributes send a PUT or a PATCH specifying the issue custom attribute id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "Duration 1"
    }' \
-s http://localhost:8000/api/v1/issue-custom-attributes/5
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON issue custom attribute detail object

32.5. Delete

To delete issue custom attributes send a DELETE specifying the issue custom attribute id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issue-custom-attributes/5
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

32.6. Bulk update order

To update the order of multiple issue custom attributes at the same time send a POST request with the following data:

- **project** (required)
- **bulk_issue_custom_attributes**: list where each element is a list, the first element is the custom attribute id and the second the new order

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_issue_custom_attributes": [
            [
                5,
                10
            ],
            [
                3,
                5
            ]
        ],
        "project": 1
    }' \
-s http://localhost:8000/api/v1/issue-custom-attributes/bulk_update_order
```

When the update is successful, the HTTP response is a 204 NO CONTENT with an empty body response

33. Issue custom attributes values

33.1. Get

To get a issue custom attribute send a GET request specifying the issue custom attribute id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/issues/custom-attributes-values/22
```

The HTTP response is a 200 OK and the response body is a JSON issue custom attribute detail object

33.2. Edit

To edit issue custom attributes values send a PUT or a PATCH specifying the issue id in the url. "attribute_values" must be a JSON object with pairs issue custom atribute id - value. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "attributes_values": {
            "5": "240 min"
        },
        "version": 1
    }' \
-s http://localhost:8000/api/v1/issues/custom-attributes-values/22
```

When the update is successful, the HTTP response is a 200 OK and the response body is a JSON issue custom attribute detail object

34. Wiki pages

34.1. List

To list wiki pages send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki
```

The HTTP response is a 200 OK and the response body is a JSON list of wiki page detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki?project=1
```

34.2. Create

To create wiki pages send a POST request with the following data:

- **project** (required): project id
- **slug** (required): slug
- **content** (required): string
- **watchers**: array of watcher id's

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "content": "Lorem ipsum dolor.",
        "project": 1,
        "slug": "new-page",
        "watchers": []
    }' \
-s http://localhost:8000/api/v1/wiki
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "content": "Lorem ipsum dolor.",
        "project": 1,
        "slug": "new-simple-page"
    }' \
-s http://localhost:8000/api/v1/wiki
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON wiki page detail object

34.3. Get

To get a wiki page send a GET request specifying the wiki page id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/5
```

The HTTP response is a 200 OK and the response body is a JSON wiki page detail object

34.4. Get by slug

To get a wiki page send a GET request specifying the wiki page slug and the project id as parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/by_slug?slug=home\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON wiki page detail object

34.5. Edit

To edit wiki pages send a PUT or a PATCH specifying the wiki page id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject",
        "version": 1
    }' \
-s http://localhost:8000/api/v1/wiki/5
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON wiki page detail object

34.6. Delete

To delete wiki page send a DELETE specifying the wiki page id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/5
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

34.7. Watch a wiki page

To watch a wiki page send a POST request specifying the wiki page id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/5/watch
```

The HTTP response is a 200 OK with an empty body response

34.8. Stop watching a wiki page

To stop watching a wiki page send a POST request specifying the wiki page id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/5/unwatch
```

The HTTP response is a 200 OK with an empty body response

34.9. List wiki page watchers

To get the list of watchers from a wiki page send a GET request specifying the wiki page id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/5/watchers
```

The HTTP response is a 200 OK and the response body is a JSON list of wiki page watcher object

34.10. List attachments

To list wiki page attachments send a GET request with the following parameters:

- **project**: project id
- **object_id**: wiki page id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/attachments?object_id=5\&project=1
```

The HTTP response is a 200 OK and the response body is a JSON list of attachment detail objects

34.11. Create attachment

To create wiki page attachments send a POST request with the following data:

- **object_id** (required): wiki page id
- **project** (required): project id
- **attached_file** (required): attaching file
- **description**
- **is_deprecated**

```
{
    "attached_file":
"attachments/6/2/d/2/9fd8261a62f1be8d66867e604b81eeab08e4e28382c23f1ca8e5e7d90c49/sample_attachment_3.txt",
    "created_date": "2020-07-02T11:58:05.759Z",
    "description": "Updated description",
    "from_comment": false,
    "id": 749,
    "is_deprecated": true,
    "modified_date": "2020-07-03T08:40:58.624Z",
    "name": "sample_attachment_3.txt",
    "object_id": 7,
    "order": 1,
    "owner": 7,
    "preview_url":
"http://localhost:8000/media/attachments/6/2/d/2/9fd8261a62f1be8d66867e604b81eeab08e4e28382c23f1ca8e5e7d90c49/sample_attachment_3.txt",
    "project": 3,
    "sha1": "da2631d805f12a1b533738a0912e9b9c2261dbef",
    "size": 1178,
    "thumbnail_card_url": null,
    "url":
"http://localhost:8000/media/attachments/6/2/d/2/9fd8261a62f1be8d66867e604b81eeab08e4e28382c23f1ca8e5e7d90c49/sample_attachment_3.txt"
}
```


When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON attachment detail object

34.12. Get attachment

To get an wiki page attachment send a GET request specifying the wiki page attachment id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/attachments/749
```

The HTTP response is a 200 OK and the response body is a JSON attachment detail object

34.13. Edit attachment

To edit wiki page attachments send a PUT or a PATCH specifying the wiki page attachment id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "Updated description"
    }' \
-s http://localhost:8000/api/v1/wiki/attachments/749
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON attachment detail object

34.14. Delete attachment

To delete wiki page attachments send a DELETE specifying the wiki page attachment id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki/attachments/749
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

35. Wiki links

35.1. List

To list wiki links send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki-links
```

The HTTP response is a 200 OK and the response body is a JSON list of wiki link detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki-links?project=1
```

35.2. Create

To create wiki links send a POST request with the following data:

- **project** (required): project id
- **title** (required): string
- **href** (required): wiki page slug
- **order**: integer

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "href": "home",
        "order": 1,
        "project": 1,
        "title": "Home page"
    }' \
-s http://localhost:8000/api/v1/wiki-links
```

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "href": "home",
        "project": 1,
        "title": "Home page"
    }' \
-s http://localhost:8000/api/v1/wiki-links
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON wiki link detail object

35.3. Get

To get a wiki link send a GET request specifying the wiki link id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki-links/1
```

The HTTP response is a 200 OK and the response body is a JSON wiki link detail object

35.4. Edit

To edit wiki links send a PUT or a PATCH specifying the wiki link id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "subject": "Patching subject"
    }' \
-s http://localhost:8000/api/v1/wiki-links/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON wiki link detail object

35.5. Delete

To delete wiki link send a DELETE specifying the wiki link id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/wiki-links/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

36. History

36.1. Get user story, task, issue or wiki page history

To get the history of a user story, task, issue or wiki page send a GET request specifying the id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/history/userstory/2
```

The HTTP response is a 200 OK and the response body is a JSON of a list of history entry detail objects

36.2. Get comment versions

To get the comment versions from the history entry of a user story, task, issue or wiki page send a GET request specifying the history entry id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/history/userstory/2/comment_versions?id=00000000-0000
-0000-0000-000000000000
```

The HTTP response is a 200 OK and the response body is a JSON of a list of history entry comment detail objects

36.3. Edit comment

To edit a history comment send a POST specifying the history entry id in the url with the following data:

- **assigned_to**: the new comment

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "comment": "comment edition"
    }' \
-s http://localhost:8000/api/v1/history/userstory/2/edit_comment?id=00000000-0000-0000
-0000-000000000000
```

When deleted succesfully, the HTTP response is a 204 NO CONTENT with an empty body response

36.4. Delete comment

To delete a history comment send a POST specifying the history entry id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/history/userstory/2/delete_comment?id=00000000-0000
-0000-0000-000000000000
```

When deleted succesfully, the HTTP response is a 204 NO CONTENT with an empty body response

36.5. Undelete comment

To undelete a history comment send a POST specifying the history entry id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/history/userstory/2/undelete_comment?id=00000000-0000
-0000-0000-000000000000
```

When deleted succesfully, the HTTP response is a 204 NO CONTENT with an empty body response

37. Users

37.1. List

To list users send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users
```

The HTTP response is a 200 OK and the response body is a JSON list of user detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users?project=1
```

37.2. Get

To get a user send a GET request specifying the user id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6
```

The HTTP response is a 200 OK and the response body is a JSON user detail object

37.3. Me

To get your own user send a GET request to the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/me
```

The HTTP response is a 200 OK and the response body is a JSON user detail object

37.4. Get user stats

To get the stats from a user send a GET request specifying the user id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6/stats
```

The HTTP response is a 200 OK and the response body is a JSON user stats object

37.5. Get watched content

To get the user watched content send a GET request specifying the user id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/1/watched
```

The HTTP response is a 200 OK and the response body is a list of JSON watched detail object

The results can be filtered using the following parameters:

- **type**: of the content. Possible values: project, userstory, task and issue
- **q**: text to search in the subject of the element

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/1/watched?type=project\&q=test
```

37.6. Get liked content

To get the user liked content send a GET request specifying the user id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6/liked
```

The HTTP response is a 200 OK and the response body is a list of JSON liked detail object

The results can be filtered using the following parameters:

- **q**: text to search in the subject of the element

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6/liked?q=test
```

37.7. Get voted content

To get the user voted content send a GET request specifying the user id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6/voted
```

The HTTP response is a 200 OK and the response body is a list of JSON voted detail object

The results can be filtered using the following parameters:

- **type**: of the content. Possible values: userstory, task and issue
- **q**: text to search in the subject of the element

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6/liked?q=test
```

37.8. Edit

To edit users send a PUT or a PATCH specifying the user id in the url. In a PATCH request you just

need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "username": "patchedusername"
    }' \
-s http://localhost:8000/api/v1/users/6
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON user detail object

37.9. Delete

To delete users send a DELETE specifying the user id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${ADMIN_AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/10
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

37.10. Get contacts

To get a user contacts send a GET request specifying the user id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/6/contacts
```

The results can be filtered using the following parameter:

- **q**: text to search in username, full name or email

The HTTP response is a 200 OK and the response body is a list of JSON contact detail object

37.11. Cancel

To cancel a user account send a POST with the following data

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "cancel_token":
"eyJ1c2VyX2NhbmNlbF9hY2NvdW50X2lkIjo2fQ:1jrHFD:5svMIhFOCpm86JDngtP1CRNPlMs"
    }' \
-s http://localhost:8000/api/v1/users/cancel
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

37.12. Change avatar

To change your user avatar send a POST with the following data

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F avatar=@test.png \
-s http://localhost:8000/api/v1/users/change_avatar
```

When the change is successful, the HTTP response is a 200 OK and the response body is a JSON user detail object

37.13. Remove avatar

To remove your user avatar send a POST with the following data

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/users/remove_avatar
```

When the change is successful, the HTTP response is a 200 OK and the response body is a JSON user detail object

37.14. Change email

To change your user email send a POST with the following data

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "email_token": "email-token"
    }' \
-s http://localhost:8000/api/v1/users/change_email
```

When the change is successful, the HTTP response is a 200 OK and the response body is a JSON user detail object

37.15. Change password

To change your user password send a POST with the following data

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "current_password": "123123",
        "password": "new-password"
    }' \
-s http://localhost:8000/api/v1/users/change_password
```

When the change succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

37.16. Password recovery

To request a user password recovery send a POST with the following data:

- **username** (required): this field also supports the user email

```
curl -X POST \
-H "Content-Type: application/json" \
-d '{
        "username": "user1"
    }' \
-s http://localhost:8000/api/v1/users/password_recovery
```

When the password recovery request succeeded, the HTTP response is a 200 OK and the response body is a JSON object:

```
{
    "detail": "Mail sended successful!"
}
```

37.17. Change password from recovery

To change a user password from a request recovery send a POST with the following data

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "password": "new-password",
        "token": "password-token"
    }' \
-s http://localhost:8000/api/v1/users/change_password_from_recovery
```

When the password change succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

38. Notify policies

38.1. List

To list the notify policies of the current user send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/notify-policies
```

The HTTP response is a 200 OK and the response body is a JSON list of notify policy detail objects

38.2. Get

To get a notify policy send a GET request specifying the notify policy id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/notify-policies/7
```

The HTTP response is a 200 OK and the response body is a JSON notify policy detail object

38.3. Edit

To edit notify policies send a PUT or a PATCH specifying the notify policy id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "notify_level": 2
    }' \
-s http://localhost:8000/api/v1/notify-policies/7
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON notify policy detail object

39. Feedback

39.1. Create

To create feedback send a POST request with the following data:

- **comment** (required)

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "comment": "Testing feedback"
    }' \
-s http://localhost:8000/api/v1/feedback
```

When created successfully, the HTTP response is a 201 Created and the response body is a JSON feedback object

40. Export/Import

40.1. Export

To get a project dump send a GET request with the project id:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/exporter/1
```

Depending on server configuration it can return two results:

- If taiga is working in synchronous mode the json file is directly generated, the result is a 200 OK and as response body a JSON of export detail for synch mode.
- If taiga is working in asynchronous mode the result is a 202 Accepted and as response body a JSON of export request accepted. The export_id can be used to build the URL to download the exported file when the file generation is complete, those urls look like: MEDIA_URL/exports/PROJECT_ID/PROJECT_SLUG-export_id.json.

40.2. Import

To load a project dump send a POST request with the following file:

- **dump** (required)

```
curl -X POST \
-H "Content-Type: multipart/form-data" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-F dump=@dump.json \
-s http://localhost:8000/api/v1/importer/load_dump
```

Depending on server configuration it can return two results:

- A 202 Accepted and as response body a JSON of import request accepted.
- A 201 Created and the response body is a JSON of project detail object

41. Webhooks

41.1. List

To list webhooks send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooks
```

The HTTP response is a 200 OK and the response body is a JSON list of webhook detail objects

The results can be filtered using the following parameters:

- **project**: project id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooks?project=1
```

41.2. Create

To create webhook send a POST request with the following data:

- **project** (required): project id
- **name** (required): string
- **url** (required): payload url
- **key** (required): secret key

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "key": "my-very-secret-key",
        "name": "My service webhook",
        "project": 1,
        "url": "http://myservice.com/webhooks"
    }' \
-s http://localhost:8000/api/v1/webhooks
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON webhook detail object

41.3. Get

To get a webhook send a GET request specifying the webhook id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooks/1
```

The HTTP response is a 200 OK and the response body is a JSON webhook detail object

41.4. Edit

To edit a webhook send a PUT or a PATCH specifying the webhook id in the url. In a PATCH request you just need to send the modified data, in a PUT one the whole object must be sent.

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "name": "My service name"
    }' \
-s http://localhost:8000/api/v1/webhooks/1
```

When the creation is successful, the HTTP response is a 200 OK and the response body is a JSON webhook detail object

41.5. Delete

To delete a webhook send a DELETE specifying the webhook id in the url

```
curl -X DELETE \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooks/1
```

When delete succeeded, the HTTP response is a 204 NO CONTENT with an empty body response

41.6. Test

To test a webhook send a POST request specifying the webhook id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooks/1/test
```

The HTTP response is a 200 OK and the response body is a JSON webhook log detail object with the resault of the test.

41.7. Logs list

To list webhook logs send a GET request to the url:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooklogs
```

The HTTP response is a 200 OK and the response body is a JSON list of webhook log detail objects

The results can be filtered using the following parameters:

- **webhook**: webhook id

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooklogs?webhook=1
```

41.8. Log get

To get a webhook log send a GET request specifying the webhook log id in the url

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooklogs/1
```

The HTTP response is a 200 OK and the response body is a JSON webhook log detail object

41.9. Resend request

To resend a request from a webhook log send a POST request specifying the webhook log id in the url

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/webhooklogs/1/resend
```

The HTTP response is a 200 OK and the response body is a JSON webhook log detail object with the resault of the resend.

42. Timelines

42.1. List user timeline

To list a user timeline send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/timeline/user/1
```

This API call returns only actions directly executed by the specified user.

The HTTP response is a 200 OK and the response body is a JSON list of timeline entry detail

42.2. List profile timeline

To list a profile timeline send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/timeline/profile/1
```

This API call returns actions executed by the specified user and related to them, watching objects, actions by related team members, belonging to projects...

The HTTP response is a 200 OK and the response body is a JSON list of timeline entry detail

42.3. List project timeline

To list a project timeline send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/timeline/project/1
```

This API call returns actions executed by different users related to the specified project.

The HTTP response is a 200 OK and the response body is a JSON list of timeline entry detail

43. Locales

43.1. List

To list the available locales send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/locales
```

The HTTP response is a 200 OK and the response body is a JSON list of locale objects

44. Stats

44.1. Get discover stats

To get the discover stats send a GET request with the following parameters:

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/stats/discover
```

The HTTP response is a 200 OK and the response body is a JSON discover stats object

44.2. Get system stats

To get the discover stats send a GET request with the following parameters:

NOTE	This API will only work if your instance has system stats enabled

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/stats/system
```

The HTTP response is a 200 OK and the response body is a JSON system stats object

45. Importers

45.1. Trello

45.1.1. Auth url

Get the url for authorize Taiga to access to your Trello account.

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/importers/trello/auth_url
```

- A 200 Ok and the response body is a JSON of importer auth url object

45.1.2. Authorize

Complete the authorization process.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "code": "00000000000000000000000000000000"
    }' \
-s http://localhost:8000/api/v1/importers/trello/authorize
```

- A 200 Ok and the response body is a JSON of importer auth token object

45.1.3. List users

List your Trello users.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": "123ABC",
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }' \
-s http://localhost:8000/api/v1/importers/trello/list_users
```

- A 200 Ok and the response body is a JSON of list of importer users object

45.1.4. List projects

List your Trello boards.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }' \
-s http://localhost:8000/api/v1/importers/trello/list_projects
```

- A 200 Ok and the response body is a JSON of list of importer projects object

45.1.5. Import project

Ask the server to import a project from Trello.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "New project description",
        "is_private": false,
        "keep_external_reference": false,
        "name": "New project name",
        "project": "123ABC",
        "template": "kanban",
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "users_bindings": {
            "user-1": "123",
            "user-2": "321"
        }
    }' \
-s http://localhost:8000/api/v1/importers/trello/import
```

- If taiga is working in synchronous mode the result is a 200 OK and as response body a JSON of imported project result.
- If taiga is working in asynchronous mode the result is a 202 Accepted and as response body a JSON of import project accepted.

45.2. Github

45.2.1. Auth url

Get the url for authorize Taiga to access to your Github account.

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/importers/github/auth_url
```

- A 200 Ok and the response body is a JSON of importer auth url object

45.2.2. Authorize

Complete the authorization process.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "code": "0000000000000000000"
    }' \
-s http://localhost:8000/api/v1/importers/github/authorize
```

- A 200 Ok and the response body is a JSON of importer auth token object

45.2.3. List users

List the Github repository users.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": "user/project",
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }' \
-s http://localhost:8000/api/v1/importers/github/list_users
```

- A 200 Ok and the response body is a JSON of list of importer users object

45.2.4. List repositories

List your Github repositories.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }' \
-s http://localhost:8000/api/v1/importers/github/list_projects
```

- A 200 Ok and the response body is a JSON of list of importer projects object

45.2.5. Import project

Ask the server to import a repository from Github.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "New project description",
        "is_private": false,
        "keep_external_reference": false,
        "name": "New project name",
        "project": "user/project",
        "template": "kanban",
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "users_bindings": {
            "user-1": "123",
            "user-2": "321"
        }
    }' \
-s http://localhost:8000/api/v1/importers/github/import
```

- If taiga is working in synchronous mode the result is a 200 OK and as response body a JSON of imported project result.
- If taiga is working in asynchronous mode the result is a 202 Accepted and as response body a JSON of import project accepted.

45.3. Jira

45.3.1. Auth url

Get the url for authorize Taiga to access to your Jira account.

```
curl -X GET \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-s http://localhost:8000/api/v1/importers/jira/auth_url?url=http://your.jira.server
```

- A 200 Ok and the response body is a JSON of importer auth url object

45.3.2. Authorize

Complete the authorization process.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{}' \
-s http://localhost:8000/api/v1/importers/jira/authorize
```

- A 200 Ok and the response body is a JSON of importer auth token object

45.3.3. List users

List the Jira project users.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "project": "12345",
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "url": "http://your.jira.server"
    }' \
-s http://localhost:8000/api/v1/importers/jira/list_users
```

- A 200 Ok and the response body is a JSON of list of importer users object

45.3.4. List projects

List your Jira projects.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "url": "http://your.jira.server"
    }' \
-s http://localhost:8000/api/v1/importers/jira/list_projects
```

- A 200 Ok and the response body is a JSON of list of importer projects object

45.3.5. Import project

Ask the server to import a project from Jira.

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "description": "New project description",
        "is_private": false,
        "keep_external_reference": false,
        "name": "New project name",
        "project": "123",
        "project_type": "kanban",
        "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "url": "http://your.jira.server",
        "users_bindings": {
            "user-1": "123",
            "user-2": "321"
        }
    }' \
-s http://localhost:8000/api/v1/importers/jira/import
```

- If taiga is working in synchronous mode the result is a 200 OK and as response body a JSON of imported project result.
- If taiga is working in asynchronous mode the result is a 202 Accepted and as response body a JSON of import project accepted.

46. Contact

46.1. Contact project

To contact the admins from a project (the project must have the contact option activated) send a POST request containing the following data:

- **project**: project id
- **comment**: comment for the admin staff

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "comment": "Comment to admins",
        "project": 3
    }' \
-s http://localhost:8000/api/v1/contact
```

When the creation is successful, the HTTP response is a 201 Created and the response body is a JSON contact project object

47. Objects Summary

47.1. Attachment


```
{
    "attached_file":
"attachments/f/7/0/0/b89c9d9aaf3de48f2486e1f5dc2f897a321bfa861cc5cb59b5936f0ebc40/sample_attachment_2.txt",
    "created_date": "2020-07-02T11:57:14.266Z",
    "description": "ipsam impedit dignissimos sed ea",
    "from_comment": false,
    "id": 415,
    "is_deprecated": false,
    "modified_date": "2020-07-02T11:57:14.266Z",
    "name": "sample_attachment_2.txt",
    "object_id": 28,
    "order": 1,
    "owner": 13,
    "preview_url":
"http://localhost:8000/media/attachments/f/7/0/0/b89c9d9aaf3de48f2486e1f5dc2f897a321bfa861cc5cb59b5936f0ebc40/sample_attachment_2.txt",
    "project": 2,
    "sha1": "62dbe72633a794717c1f4817d2d7d087b1c29c69",
    "size": 1688,
    "thumbnail_card_url": null,
    "url":
"http://localhost:8000/media/attachments/f/7/0/0/b89c9d9aaf3de48f2486e1f5dc2f897a321bfa861cc5cb59b5936f0ebc40/sample_attachment_2.txt"
}
```

```
{
    "description": "description paragraph",
    "icon_url": null,
    "id": "00000000-0000-0000-0000-000000000000",
    "name": "example application",
    "web": "http://example.com"
}
```

47.2. Application token object

```
{
    "application": {
        "description": "description paragraph",
        "icon_url": null,
        "id": "00000000-0000-0000-0000-000000000000",
        "name": "example application",
        "web": "http://example.com"
    },
    "auth_code": "bd7c35d8-2138-4fe6-8109-09b9935b7736",
    "id": 1,
    "next_url": "http://example.com?auth_code=bd7c35d8-2138-4fe6-8109-09b9935b7736",
    "user": 6
}
```

47.3. Authorization code object

```
{
    "auth_code": "bd7c35d8-2138-4fe6-8109-09b9935b7736",
    "next_url": "http://example.com?auth_code=bd7c35d8-2138-4fe6-8109-09b9935b7736",
    "state": "random-state"
}
```

47.4. Cyphered token object

```
{
    "token": "00000000-0000-0000-0000-000000000001"
}
```

47.5. User detail

```
{
    "accepted_terms": true,
    "big_photo": null,
    "bio": "",
    "color": "#40826D",
    "date_joined": "2020-07-02T11:56:19.209Z",
    "email": "user2114747470430251528@taigaio.demo",
    "full_name": "Vanesa Torres",
    "full_name_display": "Vanesa Torres",
    "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
    "id": 6,
    "is_active": true,
    "lang": "",
    "max_memberships_private_projects": null,
    "max_memberships_public_projects": null,
    "max_private_projects": null,
    "max_public_projects": null,
    "photo": null,
    "read_new_terms": false,
    "roles": [
        "Design",
        "Front",
        "Product Owner",
        "UX"
    ],
    "theme": "",
    "timezone": "",
    "total_private_projects": 4,
    "total_public_projects": 4,
    "username": "patchedusername",
    "uuid": "e9296d489b6a42d79048df2f3789b396"
}
```


47.6. User contact detail

```
{
    "big_photo": null,
    "bio": "",
    "color": "#FFCC00",
    "full_name": "Angela Perez",
    "full_name_display": "Angela Perez",
    "gravatar_id": "c9ba9d485f9a9153ebf53758feb0980c",
    "id": 11,
    "is_active": true,
    "lang": "",
    "photo": null,
    "roles": [
        "Design",
        "Front",
        "Product Owner",
        "Stakeholder",
        "UX"
    ],
    "theme": "",
    "timezone": "",
    "username": "user5"
}
```


47.7. User authentication-detail

```
{
    "accepted_terms": true,
    "auth_token":
"eyJ1c2VyX2F1dGhlbnRpY2F0aW9uX2lkIjoxNn0:1jrHFG:QR38NU7tHvPzl_s7BvgWEgnoxIc",
    "big_photo": null,
    "bio": "",
    "color": "#c9f5fe",
    "date_joined": "2020-07-03T08:40:29.738Z",
    "email": "test-register@email.com",
    "full_name": "test",
    "full_name_display": "test",
    "gravatar_id": "1ec29e4d0732b571e9a975e258a7e9b5",
    "id": 16,
    "is_active": true,
    "lang": "",
    "max_memberships_private_projects": null,
    "max_memberships_public_projects": null,
    "max_private_projects": null,
    "max_public_projects": null,
    "photo": null,
    "read_new_terms": false,
    "roles": [
        "Front"
    ],
    "theme": "",
    "timezone": "",
    "total_private_projects": 0,
    "total_public_projects": 0,
    "username": "test-username",
    "uuid": "c30015cc735e4b33b008139b58f13791"
}
```


47.8. User stats detail


```
{
    "roles": [
        "Design",
        "Front",
        "UX",
        "Product Owner"
    ],
    "total_num_closed_userstories": 2,
    "total_num_contacts": 11,
    "total_num_projects": 11
}
```

47.9. Search results detail


```
{
    "count": 31,
    "epics": [
        {
            "assigned_to": 15,
            "id": 2,
            "ref": 65,
            "status": 1,
            "subject": "Experimental: modular file types"
        },
        {
            "assigned_to": 15,
            "id": 1,
            "ref": 64,
            "status": 4,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": 6,
            "id": 6,
            "ref": 69,
            "status": 2,
            "subject": "Experimental: modular file types"
        }
    ],
    "issues": [
        {
            "assigned_to": 14,
            "id": 20,
            "ref": 61,
            "status": 7,
            "subject": "Fixing templates for Django 1.6."
        },
        {
            "assigned_to": 9,
            "id": 1,
            "ref": 42,
            "status": 5,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "assigned_to": null,
            "id": 13,
            "ref": 54,
            "status": 6,
            "subject": "Create testsuite with matrix builds"
        },
        {
```

```
            "assigned_to": 12,
            "id": 14,
            "ref": 55,
            "status": 6,
            "subject": "Create the html template"
        },
        {
            "assigned_to": null,
            "id": 3,
            "ref": 44,
            "status": 4,
            "subject": "Patching subject"
        },
        {
            "assigned_to": 12,
            "id": 10,
            "ref": 51,
            "status": 6,
            "subject": "Experimental: modular file types"
        },
        {
            "assigned_to": 8,
            "id": 12,
            "ref": 53,
            "status": 7,
            "subject": "Add setting to allow regular users to create folders at the
root level."
        },
        {
            "assigned_to": 6,
            "id": 8,
            "ref": 49,
            "status": 7,
            "subject": "Lighttpd x-sendfile support"
        },
        {
            "assigned_to": 5,
            "id": 9,
            "ref": 50,
            "status": 3,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 12,
            "id": 6,
            "ref": 47,
            "status": 2,
            "subject": "Implement the form"
        },
        {
            "assigned_to": 5,
```

```
            "id": 2,
            "ref": 43,
            "status": 5,
            "subject": "Added file copying and processing of images (resizing)"
        }
    ],
    "tasks": [
        {
            "assigned_to": 6,
            "id": 15,
            "ref": 20,
            "status": 4,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": 6,
            "id": 4,
            "ref": 5,
            "status": 2,
            "subject": "Experimental: modular file types"
        },
        {
            "assigned_to": 5,
            "id": 5,
            "ref": 7,
            "status": 1,
            "subject": "Fixing templates for Django 1.6."
        },
        {
            "assigned_to": 15,
            "id": 8,
            "ref": 11,
            "status": 5,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 7,
            "id": 7,
            "ref": 9,
            "status": 5,
            "subject": "get_actions() does not check for 'delete_selected' in actions"
        },
        {
            "assigned_to": 14,
            "id": 24,
            "ref": 33,
            "status": 3,
            "subject": "Lighttpd support"
        }
    ],
    "userstories": [
```

```
        {
            "id": 12,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 36,
            "status": 1,
            "subject": "get_actions() does not check for 'delete_selected' in
actions",
            "total_points": 31.0
        },
        {
            "id": 10,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 34,
            "status": 4,
            "subject": "Experimental: modular file types",
            "total_points": 18.0
        },
        {
            "id": 4,
            "milestone_name": "Sprint 2020-5-23",
            "milestone_slug": "sprint-2020-5-23",
            "ref": 13,
            "status": 3,
            "subject": "Support for bulk actions",
            "total_points": 110.0
        },
        {
            "id": 8,
            "milestone_name": "Sprint 2020-5-23",
            "milestone_slug": "sprint-2020-5-23",
            "ref": 30,
            "status": 1,
            "subject": "Add setting to allow regular users to create folders at the
root level.",
            "total_points": 25.0
        },
        {
            "id": 16,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 40,
            "status": 3,
            "subject": "Added file copying and processing of images (resizing)",
            "total_points": 10.0
        },
        {
            "id": 1,
            "milestone_name": "Sprint 2020-5-8",
            "milestone_slug": "sprint-2020-5-8",
```

```
            "ref": 1,
            "status": 4,
            "subject": "Patching subject",
            "total_points": 44.0
        }
    ],
    "wikipages": [
        {
            "id": 5,
            "slug": "amet"
        },
        {
            "id": 4,
            "slug": "fugit-sint"
        },
        {
            "id": 2,
            "slug": "numquam"
        },
        {
            "id": 3,
            "slug": "perspiciatis"
        },
        {
            "id": 1,
            "slug": "home"
        }
    ]
}
```


47.10. User storage data


```
{
    "created_date": "2020-07-03T08:40:53.885Z",
    "key": "favorite-forest",
    "modified_date": "2020-07-03T08:40:53.921Z",
    "value": "Russian Taiga"
}
```


48. Project templates detail


```
{
    "created_date": "2014-04-22T14:48:43.596Z",
    "default_options": {
        "epic_status": "New",
        "issue_status": "New",
        "issue_type": "Bug",
```

```
        "points": "?",
        "priority": "Normal",
        "severity": "Normal",
        "task_status": "New",
        "us_status": "New"
    },
    "default_owner_role": "product-owner",
    "description": "New description",
    "epic_statuses": [
        {
            "color": "#999999",
            "is_closed": false,
            "name": "New",
            "order": 1,
            "slug": "new"
        },
        {
            "color": "#ff8a84",
            "is_closed": false,
            "name": "Ready",
            "order": 2,
            "slug": "ready"
        },
        {
            "color": "#ff9900",
            "is_closed": false,
            "name": "In progress",
            "order": 3,
            "slug": "in-progress"
        },
        {
            "color": "#fcc000",
            "is_closed": false,
            "name": "Ready for test",
            "order": 4,
            "slug": "ready-for-test"
        },
        {
            "color": "#669900",
            "is_closed": true,
            "name": "Done",
            "order": 5,
            "slug": "done"
        }
    ],
    "id": 1,
    "is_backlog_activated": true,
    "is_contact_activated": true,
    "is_epics_activated": false,
    "is_issues_activated": true,
    "is_kanban_activated": false,
```

```
    "is_wiki_activated": true,
    "issue_statuses": [
        {
            "color": "#8C2318",
            "is_closed": false,
            "name": "New",
            "order": 1,
            "slug": "new"
        },
        {
            "color": "#5E8C6A",
            "is_closed": false,
            "name": "In progress",
            "order": 2,
            "slug": "in-progress"
        },
        {
            "color": "#88A65E",
            "is_closed": true,
            "name": "Ready for test",
            "order": 3,
            "slug": "ready-for-test"
        },
        {
            "color": "#BFB35A",
            "is_closed": true,
            "name": "Closed",
            "order": 4,
            "slug": "closed"
        },
        {
            "color": "#89BAB4",
            "is_closed": false,
            "name": "Needs Info",
            "order": 5,
            "slug": "needs-info"
        },
        {
            "color": "#CC0000",
            "is_closed": true,
            "name": "Rejected",
            "order": 6,
            "slug": "rejected"
        },
        {
            "color": "#666666",
            "is_closed": false,
            "name": "Postponed",
            "order": 7,
            "slug": "posponed"
        }
```

```
    ],
    "issue_types": [
        {
            "color": "#89BAB4",
            "name": "Bug",
            "order": 1
        },
        {
            "color": "#ba89a8",
            "name": "Question",
            "order": 2
        },
        {
            "color": "#89a8ba",
            "name": "Enhancement",
            "order": 3
        }
    ],
    "modified_date": "2020-07-03T08:41:04.047Z",
    "name": "Scrum",
    "order": 1,
    "points": [
        {
            "name": "?",
            "order": 1,
            "value": null
        },
        {
            "name": "0",
            "order": 2,
            "value": 0.0
        },
        {
            "name": "1/2",
            "order": 3,
            "value": 0.5
        },
        {
            "name": "1",
            "order": 4,
            "value": 1.0
        },
        {
            "name": "2",
            "order": 5,
            "value": 2.0
        },
        {
            "name": "3",
            "order": 6,
            "value": 3.0
```

```
        },
        {
            "name": "5",
            "order": 7,
            "value": 5.0
        },
        {
            "name": "8",
            "order": 8,
            "value": 8.0
        },
        {
            "name": "10",
            "order": 9,
            "value": 10.0
        },
        {
            "name": "13",
            "order": 10,
            "value": 13.0
        },
        {
            "name": "20",
            "order": 11,
            "value": 20.0
        },
        {
            "name": "40",
            "order": 12,
            "value": 40.0
        }
    ],
    "priorities": [
        {
            "color": "#666666",
            "name": "Low",
            "order": 1
        },
        {
            "color": "#669933",
            "name": "Normal",
            "order": 3
        },
        {
            "color": "#CC0000",
            "name": "High",
            "order": 5
        }
    ],
    "roles": [
        {
```

```
            "computable": true,
            "name": "UX",
            "order": 10,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "slug": "ux"
        },
        {
            "computable": true,
            "name": "Design",
            "order": 20,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
```

```
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "slug": "design"
        },
        {
            "computable": true,
            "name": "Front",
            "order": 30,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
```

```
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "slug": "front"
        },
        {
            "computable": true,
            "name": "Back",
            "order": 40,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
```

```
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "slug": "back"
        },
        {
            "computable": false,
            "name": "Product Owner",
            "order": 50,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "slug": "product-owner"
        },
```

```
        {
            "computable": false,
            "name": "Stakeholder",
            "order": 60,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "view_milestones",
                "view_project",
                "view_tasks",
                "view_us",
                "modify_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "slug": "stakeholder"
        }
    ],
    "severities": [
        {
            "color": "#666666",
            "name": "Wishlist",
            "order": 1
        },
        {
            "color": "#669933",
            "name": "Minor",
            "order": 2
        },
        {
            "color": "#0000FF",
            "name": "Normal",
            "order": 3
        },
        {
            "color": "#FFA500",
            "name": "Important",
            "order": 4
        },
        {
            "color": "#CC0000",
```

```
            "name": "Critical",
            "order": 5
        }
    ],
    "slug": "scrum",
    "task_statuses": [
        {
            "color": "#999999",
            "is_closed": false,
            "name": "New",
            "order": 1,
            "slug": "new"
        },
        {
            "color": "#ff9900",
            "is_closed": false,
            "name": "In progress",
            "order": 2,
            "slug": "in-progress"
        },
        {
            "color": "#ffcc00",
            "is_closed": true,
            "name": "Ready for test",
            "order": 3,
            "slug": "ready-for-test"
        },
        {
            "color": "#669900",
            "is_closed": true,
            "name": "Closed",
            "order": 4,
            "slug": "closed"
        },
        {
            "color": "#999999",
            "is_closed": false,
            "name": "Needs Info",
            "order": 5,
            "slug": "needs-info"
        }
    ],
    "us_statuses": [
        {
            "color": "#999999",
            "is_archived": false,
            "is_closed": false,
            "name": "New",
            "order": 1,
            "slug": "new",
            "wip_limit": null
```

```
        },
        {
            "color": "#ff8a84",
            "is_archived": false,
            "is_closed": false,
            "name": "Ready",
            "order": 2,
            "slug": "ready",
            "wip_limit": null
        },
        {
            "color": "#ff9900",
            "is_archived": false,
            "is_closed": false,
            "name": "In progress",
            "order": 3,
            "slug": "in-progress",
            "wip_limit": null
        },
        {
            "color": "#fcc000",
            "is_archived": false,
            "is_closed": false,
            "name": "Ready for test",
            "order": 4,
            "slug": "ready-for-test",
            "wip_limit": null
        },
        {
            "color": "#669900",
            "is_archived": false,
            "is_closed": true,
            "name": "Done",
            "order": 5,
            "slug": "done",
            "wip_limit": null
        },
        {
            "color": "#5c3566",
            "is_archived": true,
            "is_closed": true,
            "name": "Archived",
            "order": 6,
            "slug": "archived",
            "wip_limit": null
        }
    ],
    "videoconferences": null,
    "videoconferences_extra_data": ""
}
```

48.1. Project list entry


```
{
    "anon_permissions": [],
    "blocked_code": null,
    "created_date": "2020-07-03T08:40:54.649Z",
    "creation_template": 1,
    "default_epic_status": 38,
    "default_issue_status": 59,
    "default_issue_type": 25,
    "default_points": 97,
    "default_priority": 28,
    "default_severity": 45,
    "default_task_status": 43,
    "default_us_status": 49,
    "description": "Beta description",
    "i_am_admin": true,
    "i_am_member": true,
    "i_am_owner": true,
    "id": 9,
    "is_backlog_activated": true,
    "is_contact_activated": true,
    "is_epics_activated": false,
    "is_fan": false,
    "is_featured": false,
    "is_issues_activated": true,
    "is_kanban_activated": false,
    "is_looking_for_people": false,
    "is_private": true,
    "is_watcher": true,
    "is_wiki_activated": true,
    "logo_big_url": null,
    "logo_small_url": null,
    "looking_for_people_note": "",
    "members": [
        6
    ],
    "modified_date": "2020-07-03T08:40:54.874Z",
    "my_homepage": false,
    "my_permissions": [
        "add_issue",
        "delete_us",
        "delete_project",
        "modify_wiki_link",
        "delete_epic",
        "view_issues",
        "add_wiki_page",
        "comment_issue",
        "modify_epic",
        "delete_issue",
```

```
        "delete_wiki_link",
        "delete_task",
        "admin_roles",
        "view_wiki_pages",
        "modify_wiki_page",
        "delete_wiki_page",
        "delete_milestone",
        "comment_task",
        "comment_wiki_page",
        "view_project",
        "add_task",
        "view_wiki_links",
        "view_tasks",
        "add_us",
        "add_milestone",
        "modify_us",
        "modify_milestone",
        "comment_epic",
        "modify_issue",
        "admin_project_values",
        "view_milestones",
        "remove_member",
        "add_member",
        "view_epics",
        "view_us",
        "comment_us",
        "modify_task",
        "add_epic",
        "modify_project",
        "add_wiki_link"
    ],
    "name": "Beta project",
    "notify_level": 1,
    "owner": {
        "big_photo": null,
        "full_name_display": "Vanesa Torres",
        "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user2114747470430251528"
    },
    "public_permissions": [],
    "slug": "user2114747470430251528-beta-project",
    "tags": [],
    "tags_colors": {},
    "total_activity": 1,
    "total_activity_last_month": 1,
    "total_activity_last_week": 1,
    "total_activity_last_year": 1,
    "total_closed_milestones": 0,
```

```
    "total_fans": 0,
    "total_fans_last_month": 0,
    "total_fans_last_week": 0,
    "total_fans_last_year": 0,
    "total_milestones": null,
    "total_story_points": null,
    "total_watchers": 1,
    "totals_updated_datetime": "2020-07-03T08:40:54.906Z",
    "videoconferences": null,
    "videoconferences_extra_data": null
}
```


48.2. Project detail


```
{
    "anon_permissions": [],
    "blocked_code": null,
    "created_date": "2020-07-02T11:56:20.265Z",
    "creation_template": null,
    "default_epic_status": null,
    "default_issue_status": null,
    "default_issue_type": null,
    "default_points": null,
    "default_priority": null,
    "default_severity": null,
    "default_task_status": null,
    "default_us_status": null,
    "description": "Beta description",
    "epic_custom_attributes": [
        {
            "created_date": "2020-07-02T11:56:21.140022+00:00",
            "description": "aut nostrum consequatur quasi odio et",
            "extra": null,
            "id": 1,
            "modified_date": "2020-07-02T11:56:21.140039+00:00",
            "name": "odit",
            "order": 1,
            "project_id": 1,
            "type": "richtext"
        },
        {
            "created_date": "2020-07-02T11:56:21.14223+00:00",
            "description": "id distinctio sequi amet incidunt",
            "extra": null,
            "id": 2,
            "modified_date": "2020-07-02T11:56:21.142246+00:00",
            "name": "corrupti totam voluptatibus",
            "order": 1,
            "project_id": 1,
```

```
            "type": "richtext"
        },
        {
            "created_date": "2020-07-02T11:56:21.14364+00:00",
            "description": "ratione eius necessitatibus ad quibusdam",
            "extra": null,
            "id": 3,
            "modified_date": "2020-07-02T11:56:21.143654+00:00",
            "name": "quod",
            "order": 1,
            "project_id": 1,
            "type": "dropdown"
        },
        {
            "created_date": "2020-07-02T11:56:21.144998+00:00",
            "description": "ratione doloribus explicabo",
            "extra": null,
            "id": 4,
            "modified_date": "2020-07-02T11:56:21.145016+00:00",
            "name": "enim similique",
            "order": 1,
            "project_id": 1,
            "type": "date"
        },
        {
            "created_date": "2020-07-02T11:56:21.146523+00:00",
            "description": "nostrum asperiores eveniet libero consequuntur expedita
velit accusantium",
            "extra": null,
            "id": 5,
            "modified_date": "2020-07-02T11:56:21.146539+00:00",
            "name": "dolores",
            "order": 1,
            "project_id": 1,
            "type": "number"
        },
        {
            "created_date": "2020-07-03T08:40:34.707016+00:00",
            "description": "Duration in minutes",
            "extra": null,
            "id": 26,
            "modified_date": "2020-07-03T08:40:34.708806+00:00",
            "name": "Duration 2",
            "order": 8,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2020-07-03T08:40:34.746556+00:00",
            "description": "",
            "extra": null,
```

```
            "id": 27,
            "modified_date": "2020-07-03T08:40:34.748075+00:00",
            "name": "Duration 3",
            "order": 1593765634740,
            "project_id": 1,
            "type": "text"
        }
    ],
    "epic_statuses": [
        {
            "color": "#ff9900",
            "id": 3,
            "is_closed": false,
            "name": "In progress",
            "order": 3,
            "project_id": 1,
            "slug": "in-progress"
        },
        {
            "color": "#fcc000",
            "id": 4,
            "is_closed": false,
            "name": "Ready for test",
            "order": 4,
            "project_id": 1,
            "slug": "ready-for-test"
        },
        {
            "color": "#ff8a84",
            "id": 2,
            "is_closed": false,
            "name": "Ready",
            "order": 5,
            "project_id": 1,
            "slug": "ready"
        },
        {
            "color": "#669900",
            "id": 5,
            "is_closed": true,
            "name": "Done",
            "order": 5,
            "project_id": 1,
            "slug": "done"
        },
        {
            "color": "#AAAAAA",
            "id": 36,
            "is_closed": true,
            "name": "New status",
            "order": 8,
```

```
            "project_id": 1,
            "slug": "new-status"
        },
        {
            "color": "#999999",
            "id": 1,
            "is_closed": false,
            "name": "Patch status name",
            "order": 10,
            "project_id": 1,
            "slug": "patch-status-name"
        },
        {
            "color": "#999999",
            "id": 37,
            "is_closed": false,
            "name": "New status name",
            "order": 10,
            "project_id": 1,
            "slug": "new-status-name"
        }
    ],
    "epics_csv_uuid": null,
    "i_am_admin": true,
    "i_am_member": true,
    "i_am_owner": true,
    "id": 1,
    "is_backlog_activated": true,
    "is_contact_activated": true,
    "is_epics_activated": false,
    "is_fan": true,
    "is_featured": false,
    "is_issues_activated": true,
    "is_kanban_activated": false,
    "is_looking_for_people": false,
    "is_out_of_owner_limits": false,
    "is_private": true,
    "is_private_extra_info": {
        "can_be_updated": true,
        "reason": null
    },
    "is_watcher": false,
    "is_wiki_activated": true,
    "issue_custom_attributes": [
        {
            "created_date": "2020-07-02T11:56:21.164615+00:00",
            "description": "officiis repudiandae dignissimos similique consequatur
mollitia at enim ad molestias praesentium",
            "extra": null,
            "id": 1,
            "modified_date": "2020-07-02T11:56:21.164631+00:00",
```

```
            "name": "fugiat optio consequuntur",
            "order": 1,
            "project_id": 1,
            "type": "dropdown"
        },
        {
            "created_date": "2020-07-02T11:56:21.166561+00:00",
            "description": "minus quibusdam neque eveniet repellendus ex dolorum optio
ullam vitae",
            "extra": null,
            "id": 2,
            "modified_date": "2020-07-02T11:56:21.166575+00:00",
            "name": "doloremque id",
            "order": 1,
            "project_id": 1,
            "type": "checkbox"
        },
        {
            "created_date": "2020-07-02T11:56:21.168149+00:00",
            "description": "facere corrupti ipsa odit mollitia saepe officiis",
            "extra": null,
            "id": 3,
            "modified_date": "2020-07-02T11:56:21.168171+00:00",
            "name": "doloribus ducimus nulla",
            "order": 1,
            "project_id": 1,
            "type": "dropdown"
        },
        {
            "created_date": "2020-07-02T11:56:21.169672+00:00",
            "description": "fugiat porro officia deleniti quidem ipsam",
            "extra": null,
            "id": 4,
            "modified_date": "2020-07-02T11:56:21.169687+00:00",
            "name": "velit",
            "order": 1,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2020-07-02T11:56:21.171158+00:00",
            "description": "voluptate rem perspiciatis ipsum",
            "extra": null,
            "id": 5,
            "modified_date": "2020-07-02T11:56:21.171172+00:00",
            "name": "adipisci exercitationem",
            "order": 1,
            "project_id": 1,
            "type": "checkbox"
        }
    ],
```

```
    "issue_duedates": [
        {
            "by_default": true,
            "color": "#9dce0a",
            "days_to_due": null,
            "id": 1,
            "name": "Default",
            "order": 1,
            "project_id": 1
        },
        {
            "by_default": false,
            "color": "#ff9900",
            "days_to_due": 14,
            "id": 2,
            "name": "Due soon",
            "order": 2,
            "project_id": 1
        },
        {
            "by_default": false,
            "color": "#ff8a84",
            "days_to_due": 0,
            "id": 3,
            "name": "Past due",
            "order": 3,
            "project_id": 1
        }
    ],
    "issue_statuses": [
        {
            "color": "#88A65E",
            "id": 3,
            "is_closed": true,
            "name": "Ready for test",
            "order": 3,
            "project_id": 1,
            "slug": "ready-for-test"
        },
        {
            "color": "#BFB35A",
            "id": 4,
            "is_closed": true,
            "name": "Closed",
            "order": 4,
            "project_id": 1,
            "slug": "closed"
        },
        {
            "color": "#5E8C6A",
            "id": 2,
```

```
            "is_closed": false,
            "name": "In progress",
            "order": 5,
            "project_id": 1,
            "slug": "in-progress"
        },
        {
            "color": "#89BAB4",
            "id": 5,
            "is_closed": false,
            "name": "Needs Info",
            "order": 5,
            "project_id": 1,
            "slug": "needs-info"
        },
        {
            "color": "#CC0000",
            "id": 6,
            "is_closed": true,
            "name": "Rejected",
            "order": 6,
            "project_id": 1,
            "slug": "rejected"
        },
        {
            "color": "#666666",
            "id": 7,
            "is_closed": false,
            "name": "Postponed",
            "order": 7,
            "project_id": 1,
            "slug": "postponed"
        },
        {
            "color": "#AAAAAA",
            "id": 50,
            "is_closed": true,
            "name": "New status",
            "order": 8,
            "project_id": 1,
            "slug": "new-status"
        },
        {
            "color": "#999999",
            "id": 51,
            "is_closed": false,
            "name": "New status name",
            "order": 10,
            "project_id": 1,
            "slug": "new-status-name"
        },
```

```
        {
            "color": "#8C2318",
            "id": 1,
            "is_closed": false,
            "name": "Patch status name",
            "order": 10,
            "project_id": 1,
            "slug": "patch-status-name"
        }
    ],
    "issue_types": [
        {
            "color": "#89BAB4",
            "id": 1,
            "name": "Bug",
            "order": 1,
            "project_id": 1
        },
        {
            "color": "#ba89a8",
            "id": 2,
            "name": "Question",
            "order": 2,
            "project_id": 1
        },
        {
            "color": "#89a8ba",
            "id": 3,
            "name": "Enhancement",
            "order": 3,
            "project_id": 1
        }
    ],
    "issues_csv_uuid": null,
    "logo_big_url":
"http://localhost:8000/media/project/4/f/3/3/56ab780682ed9426ac722feaf310aa1409927c2ac
39702c8323196509be8/test.png.300x300_q85_crop.png",
    "logo_small_url":
"http://localhost:8000/media/project/4/f/3/3/56ab780682ed9426ac722feaf310aa1409927c2ac
39702c8323196509be8/test.png.80x80_q85_crop.png",
    "looking_for_people_note": "",
    "max_memberships": null,
    "members": [
        {
            "color": "",
            "full_name": "Administrator",
            "full_name_display": "Administrator",
            "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
            "id": 5,
            "is_active": true,
            "photo": null,
```

```
            "role": 4,
            "role_name": "Back",
            "username": "admin"
        },
        {
            "color": "#40826D",
            "full_name": "Bego\u00f1a Flores",
            "full_name_display": "Bego\u00f1a Flores",
            "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
            "id": 7,
            "is_active": true,
            "photo": null,
            "role": 1,
            "role_name": "Patch name",
            "username": "user1"
        },
        {
            "color": "#B6DA55",
            "full_name": "Catalina Fernandez",
            "full_name_display": "Catalina Fernandez",
            "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
            "id": 9,
            "is_active": true,
            "photo": null,
            "role": 4,
            "role_name": "Back",
            "username": "user3"
        },
        {
            "color": "#2099DB",
            "full_name": "Enrique Crespo",
            "full_name_display": "Enrique Crespo",
            "gravatar_id": "f31e0063c7cd6da19b6467bc48d2b14b",
            "id": 10,
            "is_active": true,
            "photo": null,
            "role": 5,
            "role_name": "Product Owner",
            "username": "user4"
        },
        {
            "color": "#71A6D2",
            "full_name": "Francisco Gil",
            "full_name_display": "Francisco Gil",
            "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
            "id": 8,
            "is_active": true,
            "photo": null,
            "role": 3,
            "role_name": "Front",
            "username": "user2"
```

```
        },
        {
            "color": "#002e33",
            "full_name": "Miguel Molina",
            "full_name_display": "Miguel Molina",
            "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
            "id": 14,
            "is_active": true,
            "photo": null,
            "role": 5,
            "role_name": "Product Owner",
            "username": "user8"
        },
        {
            "color": "#B6DA55",
            "full_name": "Mohamed Ortega",
            "full_name_display": "Mohamed Ortega",
            "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
            "id": 13,
            "is_active": true,
            "photo": null,
            "role": 5,
            "role_name": "Product Owner",
            "username": "user7"
        },
        {
            "color": "#c9f5fe",
            "full_name": "test",
            "full_name_display": "test",
            "gravatar_id": "1ec29e4d0732b571e9a975e258a7e9b5",
            "id": 16,
            "is_active": true,
            "photo": null,
            "role": 3,
            "role_name": "Front",
            "username": "test-username"
        },
        {
            "color": "#71A6D2",
            "full_name": "Vanesa Garcia",
            "full_name_display": "Vanesa Garcia",
            "gravatar_id": "74cb769a5e64d445b8550789e1553502",
            "id": 12,
            "is_active": true,
            "photo": null,
            "role": 6,
            "role_name": "Stakeholder",
            "username": "user6"
        },
        {
            "color": "#40826D",
```

```
            "full_name": "Vanesa Torres",
            "full_name_display": "Vanesa Torres",
            "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
            "id": 6,
            "is_active": true,
            "photo": null,
            "role": 2,
            "role_name": "Design",
            "username": "user2114747470430251528"
        },
        {
            "color": "#FFFF00",
            "full_name": "Virginia Castro",
            "full_name_display": "Virginia Castro",
            "gravatar_id": "69b60d39a450e863609ae3546b12b360",
            "id": 15,
            "is_active": true,
            "photo": null,
            "role": 6,
            "role_name": "Stakeholder",
            "username": "user9"
        }
    ],
    "milestones": [
        {
            "closed": false,
            "id": 1,
            "name": "Sprint 2020-5-8",
            "slug": "sprint-2020-5-8"
        },
        {
            "closed": false,
            "id": 2,
            "name": "Sprint 2020-5-23",
            "slug": "sprint-2020-5-23"
        }
    ],
    "modified_date": "2020-07-03T08:40:54.558Z",
    "my_homepage": false,
    "my_permissions": [
        "add_issue",
        "delete_us",
        "delete_project",
        "modify_wiki_link",
        "delete_epic",
        "view_issues",
        "add_wiki_page",
        "comment_issue",
        "modify_epic",
        "delete_issue",
        "delete_wiki_link",
```

```
        "delete_task",
        "admin_roles",
        "view_wiki_pages",
        "modify_wiki_page",
        "delete_wiki_page",
        "delete_milestone",
        "comment_task",
        "comment_wiki_page",
        "view_project",
        "add_task",
        "view_wiki_links",
        "view_tasks",
        "add_us",
        "add_milestone",
        "modify_us",
        "modify_milestone",
        "comment_epic",
        "modify_issue",
        "admin_project_values",
        "view_milestones",
        "remove_member",
        "add_member",
        "view_epics",
        "view_us",
        "comment_us",
        "modify_task",
        "add_epic",
        "modify_project",
        "add_wiki_link"
    ],
    "name": "Beta project patch",
    "notify_level": 3,
    "owner": {
        "big_photo": null,
        "full_name_display": "Vanesa Torres",
        "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user2114747470430251528"
    },
    "points": [
        {
            "id": 1,
            "name": "?",
            "order": 1,
            "project_id": 1,
            "value": null
        },
        {
            "id": 2,
```

```
            "name": "0",
            "order": 2,
            "project_id": 1,
            "value": 0
        },
        {
            "id": 3,
            "name": "1/2",
            "order": 3,
            "project_id": 1,
            "value": 0.5
        },
        {
            "id": 4,
            "name": "1",
            "order": 4,
            "project_id": 1,
            "value": 1
        },
        {
            "id": 5,
            "name": "2",
            "order": 5,
            "project_id": 1,
            "value": 2
        },
        {
            "id": 6,
            "name": "3",
            "order": 6,
            "project_id": 1,
            "value": 3
        },
        {
            "id": 7,
            "name": "5",
            "order": 7,
            "project_id": 1,
            "value": 5
        },
        {
            "id": 8,
            "name": "8",
            "order": 8,
            "project_id": 1,
            "value": 8
        },
        {
            "id": 9,
            "name": "10",
            "order": 9,
```

```
            "project_id": 1,
            "value": 10
        },
        {
            "id": 10,
            "name": "13",
            "order": 10,
            "project_id": 1,
            "value": 13
        },
        {
            "id": 11,
            "name": "20",
            "order": 11,
            "project_id": 1,
            "value": 20
        },
        {
            "id": 12,
            "name": "40",
            "order": 12,
            "project_id": 1,
            "value": 40
        }
    ],
    "priorities": [
        {
            "color": "#CC0000",
            "id": 3,
            "name": "High",
            "order": 5,
            "project_id": 1
        },
        {
            "color": "#669933",
            "id": 2,
            "name": "Normal",
            "order": 5,
            "project_id": 1
        },
        {
            "color": "#AAAAAA",
            "id": 25,
            "name": "New priority",
            "order": 8,
            "project_id": 1
        },
        {
            "color": "#999999",
            "id": 26,
            "name": "New priority name",
```

```
            "order": 10,
            "project_id": 1
        },
        {
            "color": "#666666",
            "id": 1,
            "name": "Patch name",
            "order": 10,
            "project_id": 1
        }
    ],
    "public_permissions": [],
    "roles": [
        {
            "computable": true,
            "id": 1,
            "name": "Patch name",
            "order": 10,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
```

```
                "comment_wiki_page"
            ],
            "project_id": 1,
            "slug": "ux"
        },
        {
            "computable": true,
            "id": 44,
            "name": "New role name",
            "order": 10,
            "permissions": [],
            "project_id": 1,
            "slug": "new-role-name"
        },
        {
            "computable": true,
            "id": 43,
            "name": "New role",
            "order": 10,
            "permissions": [
                "view_us",
                "view_project"
            ],
            "project_id": 1,
            "slug": "new-role"
        },
        {
            "computable": true,
            "id": 2,
            "name": "Design",
            "order": 20,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
```

```
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "project_id": 1,
            "slug": "design"
        },
        {
            "computable": true,
            "id": 3,
            "name": "Front",
            "order": 30,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
```

```
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "project_id": 1,
            "slug": "front"
        },
        {
            "computable": true,
            "id": 4,
            "name": "Back",
            "order": 40,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "project_id": 1,
```

```
            "slug": "back"
        },
        {
            "computable": false,
            "id": 5,
            "name": "Product Owner",
            "order": 50,
            "permissions": [
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "add_milestone",
                "modify_milestone",
                "delete_milestone",
                "view_milestones",
                "view_project",
                "add_task",
                "modify_task",
                "delete_task",
                "view_tasks",
                "add_us",
                "modify_us",
                "delete_us",
                "view_us",
                "add_wiki_page",
                "modify_wiki_page",
                "delete_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "add_epic",
                "modify_epic",
                "delete_epic",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "project_id": 1,
            "slug": "product-owner"
        },
        {
            "computable": false,
            "id": 6,
            "name": "Stakeholder",
            "order": 60,
            "permissions": [
```

```
                "add_issue",
                "modify_issue",
                "delete_issue",
                "view_issues",
                "view_milestones",
                "view_project",
                "view_tasks",
                "view_us",
                "modify_wiki_page",
                "view_wiki_pages",
                "add_wiki_link",
                "delete_wiki_link",
                "view_wiki_links",
                "view_epics",
                "comment_epic",
                "comment_us",
                "comment_task",
                "comment_issue",
                "comment_wiki_page"
            ],
            "project_id": 1,
            "slug": "stakeholder"
        }
    ],
    "severities": [
        {
            "color": "#0000FF",
            "id": 3,
            "name": "Normal",
            "order": 3,
            "project_id": 1
        },
        {
            "color": "#FFA500",
            "id": 4,
            "name": "Important",
            "order": 4,
            "project_id": 1
        },
        {
            "color": "#669933",
            "id": 2,
            "name": "Minor",
            "order": 5,
            "project_id": 1
        },
        {
            "color": "#CC0000",
            "id": 5,
            "name": "Critical",
            "order": 5,
```

```
            "project_id": 1
        },
        {
            "color": "#AAAAAA",
            "id": 41,
            "name": "New severity",
            "order": 8,
            "project_id": 1
        },
        {
            "color": "#666666",
            "id": 1,
            "name": "Patch name",
            "order": 10,
            "project_id": 1
        },
        {
            "color": "#999999",
            "id": 42,
            "name": "New severity name",
            "order": 10,
            "project_id": 1
        }
    ],
    "slug": "project-0",
    "tags": [],
    "tags_colors": {},
    "task_custom_attributes": [
        {
            "created_date": "2020-07-02T11:56:21.15656+00:00",
            "description": "a sequi saepe quibusdam culpa optio accusantium minima
obcaecati",
            "extra": null,
            "id": 1,
            "modified_date": "2020-07-02T11:56:21.156576+00:00",
            "name": "esse omnis soluta",
            "order": 1,
            "project_id": 1,
            "type": "dropdown"
        },
        {
            "created_date": "2020-07-02T11:56:21.158394+00:00",
            "description": "laudantium totam dolorem minima nemo quaerat voluptate
aliquam autem quasi distinctio inventore",
            "extra": null,
            "id": 2,
            "modified_date": "2020-07-02T11:56:21.158409+00:00",
            "name": "libero",
            "order": 1,
            "project_id": 1,
            "type": "richtext"
```

```
        },
        {
            "created_date": "2020-07-02T11:56:21.159928+00:00",
            "description": "aliquid laboriosam soluta libero quo fugit molestiae
impedit officia at",
            "extra": null,
            "id": 3,
            "modified_date": "2020-07-02T11:56:21.159942+00:00",
            "name": "soluta",
            "order": 1,
            "project_id": 1,
            "type": "date"
        },
        {
            "created_date": "2020-07-02T11:56:21.161459+00:00",
            "description": "totam autem aut fuga odit",
            "extra": null,
            "id": 4,
            "modified_date": "2020-07-02T11:56:21.161473+00:00",
            "name": "unde",
            "order": 1,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2020-07-02T11:56:21.163021+00:00",
            "description": "debitis dolorum soluta mollitia aliquid sapiente nesciunt
molestias cum deserunt corporis officiis",
            "extra": null,
            "id": 5,
            "modified_date": "2020-07-02T11:56:21.16304+00:00",
            "name": "obcaecati quasi impedit",
            "order": 1,
            "project_id": 1,
            "type": "richtext"
        }
    ],
    "task_duedates": [
        {
            "by_default": true,
            "color": "#9dce0a",
            "days_to_due": null,
            "id": 1,
            "name": "Default",
            "order": 1,
            "project_id": 1
        },
        {
            "by_default": false,
            "color": "#ff9900",
            "days_to_due": 14,
```

```
            "id": 2,
            "name": "Due soon",
            "order": 2,
            "project_id": 1
        },
        {
            "by_default": false,
            "color": "#ff8a84",
            "days_to_due": 0,
            "id": 3,
            "name": "Past due",
            "order": 3,
            "project_id": 1
        }
    ],
    "task_statuses": [
        {
            "color": "#ffcc00",
            "id": 3,
            "is_closed": true,
            "name": "Ready for test",
            "order": 3,
            "project_id": 1,
            "slug": "ready-for-test"
        },
        {
            "color": "#669900",
            "id": 4,
            "is_closed": true,
            "name": "Closed",
            "order": 4,
            "project_id": 1,
            "slug": "closed"
        },
        {
            "color": "#ff9900",
            "id": 2,
            "is_closed": false,
            "name": "In progress",
            "order": 5,
            "project_id": 1,
            "slug": "in-progress"
        },
        {
            "color": "#999999",
            "id": 5,
            "is_closed": false,
            "name": "Needs Info",
            "order": 5,
            "project_id": 1,
            "slug": "needs-info"
```

```
        },
        {
            "color": "#AAAAAA",
            "id": 41,
            "is_closed": true,
            "name": "New status",
            "order": 8,
            "project_id": 1,
            "slug": "new-status"
        },
        {
            "color": "#999999",
            "id": 1,
            "is_closed": false,
            "name": "Patch status name",
            "order": 10,
            "project_id": 1,
            "slug": "patch-status-name"
        },
        {
            "color": "#999999",
            "id": 42,
            "is_closed": false,
            "name": "New status name",
            "order": 10,
            "project_id": 1,
            "slug": "new-status-name"
        }
    ],
    "tasks_csv_uuid": null,
    "total_activity": 189,
    "total_activity_last_month": 189,
    "total_activity_last_week": 189,
    "total_activity_last_year": 189,
    "total_closed_milestones": 0,
    "total_fans": 10,
    "total_fans_last_month": 10,
    "total_fans_last_week": 10,
    "total_fans_last_year": 10,
    "total_memberships": 16,
    "total_milestones": 6,
    "total_story_points": 313.0,
    "total_watchers": 15,
    "totals_updated_datetime": "2020-07-03T08:40:54.577Z",
    "transfer_token": "6:1jrHFD:8NuXY5qtgY406k-oQrs_o9KMu-s",
    "us_duedates": [
        {
            "by_default": true,
            "color": "#9dce0a",
            "days_to_due": null,
            "id": 1,
```

```
            "name": "Default",
            "order": 1,
            "project_id": 1
        },
        {
            "by_default": false,
            "color": "#ff9900",
            "days_to_due": 14,
            "id": 2,
            "name": "Due soon",
            "order": 2,
            "project_id": 1
        },
        {
            "by_default": false,
            "color": "#ff8a84",
            "days_to_due": 0,
            "id": 3,
            "name": "Past due",
            "order": 3,
            "project_id": 1
        }
    ],
    "us_statuses": [
        {
            "color": "#999999",
            "id": 1,
            "is_archived": false,
            "is_closed": false,
            "name": "New",
            "order": 1,
            "project_id": 1,
            "slug": "new",
            "wip_limit": null
        },
        {
            "color": "#ff8a84",
            "id": 2,
            "is_archived": false,
            "is_closed": false,
            "name": "Ready",
            "order": 2,
            "project_id": 1,
            "slug": "ready",
            "wip_limit": null
        },
        {
            "color": "#ff9900",
            "id": 3,
            "is_archived": false,
            "is_closed": false,
```

```
            "name": "In progress",
            "order": 3,
            "project_id": 1,
            "slug": "in-progress",
            "wip_limit": null
        },
        {
            "color": "#fcc000",
            "id": 4,
            "is_archived": false,
            "is_closed": false,
            "name": "Ready for test",
            "order": 4,
            "project_id": 1,
            "slug": "ready-for-test",
            "wip_limit": null
        },
        {
            "color": "#669900",
            "id": 5,
            "is_archived": false,
            "is_closed": true,
            "name": "Done",
            "order": 5,
            "project_id": 1,
            "slug": "done",
            "wip_limit": null
        },
        {
            "color": "#5c3566",
            "id": 6,
            "is_archived": true,
            "is_closed": true,
            "name": "Archived",
            "order": 6,
            "project_id": 1,
            "slug": "archived",
            "wip_limit": null
        }
    ],
    "userstories_csv_uuid": null,
    "userstory_custom_attributes": [
        {
            "created_date": "2020-07-02T11:56:21.153216+00:00",
            "description": "vitae error dignissimos ipsa minus nostrum",
            "extra": null,
            "id": 4,
            "modified_date": "2020-07-02T11:56:21.153229+00:00",
            "name": "obcaecati quaerat",
            "order": 1,
            "project_id": 1,
```

```
            "type": "richtext"
        },
        {
            "created_date": "2020-07-02T11:56:21.154934+00:00",
            "description": "necessitatibus velit aliquam exercitationem debitis
laboriosam",
            "extra": null,
            "id": 5,
            "modified_date": "2020-07-02T11:56:21.154948+00:00",
            "name": "facilis temporibus",
            "order": 1,
            "project_id": 1,
            "type": "date"
        },
        {
            "created_date": "2020-07-02T11:56:21.151681+00:00",
            "description": "ex a nihil porro placeat",
            "extra": null,
            "id": 3,
            "modified_date": "2020-07-02T11:56:21.151698+00:00",
            "name": "eveniet",
            "order": 1,
            "project_id": 1,
            "type": "checkbox"
        },
        {
            "created_date": "2020-07-02T11:56:21.150223+00:00",
            "description": "inventore ab iusto optio tempora hic",
            "extra": null,
            "id": 2,
            "modified_date": "2020-07-02T11:56:21.150238+00:00",
            "name": "maiores harum ipsa",
            "order": 5,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2020-07-03T08:40:39.219974+00:00",
            "description": "Duration in minutes",
            "extra": null,
            "id": 26,
            "modified_date": "2020-07-03T08:40:39.22255+00:00",
            "name": "Duration 2",
            "order": 8,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2020-07-02T11:56:21.148166+00:00",
            "description": "vel omnis culpa quisquam nulla",
            "extra": null,
```

```
            "id": 1,
            "modified_date": "2020-07-03T08:40:39.162601+00:00",
            "name": "Duration 1",
            "order": 10,
            "project_id": 1,
            "type": "richtext"
        },
        {
            "created_date": "2020-07-03T08:40:39.283413+00:00",
            "description": "",
            "extra": null,
            "id": 27,
            "modified_date": "2020-07-03T08:40:39.285351+00:00",
            "name": "Duration 3",
            "order": 1593765639275,
            "project_id": 1,
            "type": "text"
        }
    ],
    "videoconferences": null,
    "videoconferences_extra_data": null
}
```


48.3. Project modules configuration

```
{
    "bitbucket": {
        "secret": "f805de8e75fc407d8c8ed16cd40f9ba6",
        "valid_origin_ips": [
            "131.103.20.165",
            "131.103.20.166",
            "104.192.143.192/28",
            "104.192.143.208/28"
        ],
        "webhooks_url": "http://localhost:8000/api/v1/bitbucket-
hook?project=1&key=f805de8e75fc407d8c8ed16cd40f9ba6"
    },
    "github": {
        "secret": "7ad6a6302b614eb5b7a53855dc6b0552",
        "webhooks_url": "http://localhost:8000/api/v1/github-hook?project=1"
    },
    "gitlab": {
        "secret": "89b8c12250074e8a83c7cab674462aa2",
        "valid_origin_ips": [],
        "webhooks_url": "http://localhost:8000/api/v1/gitlab-
hook?project=1&key=89b8c12250074e8a83c7cab674462aa2"
    },
    "gogs": {
        "secret": "3bccc05f59dd41c9944023a1bf4a3f9c",
        "webhooks_url": "http://localhost:8000/api/v1/gogs-hook?project=1"
    }
}
```


48.4. Project stats detail


```
{
    "assigned_points": 364.5,
    "assigned_points_per_role": {
        "1": 110.5,
        "2": 104.5,
        "3": 70.5,
        "4": 79.0
    },
    "closed_points": 44.0,
    "closed_points_per_role": {
        "1": 21.0,
        "2": 1.5,
        "3": 3.5,
        "4": 18.0
    },
    "defined_points": 628.5,
    "defined_points_per_role": {
        "1": 162.5,
```

```
        "2": 213.5,
        "3": 82.5,
        "4": 170.0
    },
    "milestones": [
        {
            "client-increment": 0,
            "evolution": 313.0,
            "name": "Sprint 2020-5-8",
            "optimal": 313.0,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": 313.0,
            "name": "Sprint 2020-5-23",
            "optimal": 260.8333333333333,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": 269.0,
            "name": "Future sprint",
            "optimal": 208.66666666666669,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Future sprint",
            "optimal": 156.5,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Future sprint",
            "optimal": 104.33333333333334,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Future sprint",
            "optimal": 52.166666666666686,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Project End",
```

```
            "optimal": 2.1316282072803006e-14,
            "team-increment": 0
        }
    ],
    "name": "Project Example 0",
    "speed": 0,
    "total_milestones": 6,
    "total_points": 313.0
}
```


48.5. Project issue stats detail


```
{
    "closed_issues": 10,
    "issues_per_assigned_to": {
        "0": {
            "color": "black",
            "count": 8,
            "id": 0,
            "name": "Unassigned",
            "username": "Unassigned"
        },
        "10": {
            "color": "#2099DB",
            "count": 1,
            "id": 10,
            "name": "Enrique Crespo",
            "username": "user4"
        },
        "12": {
            "color": "#71A6D2",
            "count": 3,
            "id": 12,
            "name": "Vanesa Garcia",
            "username": "user6"
        },
        "13": {
            "color": "#B6DA55",
            "count": 1,
            "id": 13,
            "name": "Mohamed Ortega",
            "username": "user7"
        },
        "14": {
            "color": "#002e33",
            "count": 1,
            "id": 14,
            "name": "Miguel Molina",
            "username": "user8"
```

```
        },
        "5": {
            "color": "",
            "count": 5,
            "id": 5,
            "name": "Administrator",
            "username": "admin"
        },
        "6": {
            "color": "#40826D",
            "count": 3,
            "id": 6,
            "name": "Vanesa Torres",
            "username": "user2114747470430251528"
        },
        "7": {
            "color": "#40826D",
            "count": 1,
            "id": 7,
            "name": "Bego\u00f1a Flores",
            "username": "user1"
        },
        "8": {
            "color": "#71A6D2",
            "count": 2,
            "id": 8,
            "name": "Francisco Gil",
            "username": "user2"
        },
        "9": {
            "color": "#B6DA55",
            "count": 2,
            "id": 9,
            "name": "Catalina Fernandez",
            "username": "user3"
        }
    },
    "issues_per_owner": {
        "10": {
            "color": "#2099DB",
            "count": 1,
            "id": 10,
            "name": "Enrique Crespo",
            "username": "user4"
        },
        "12": {
            "color": "#71A6D2",
            "count": 2,
            "id": 12,
            "name": "Vanesa Garcia",
            "username": "user6"
```

```
        },
        "13": {
            "color": "#B6DA55",
            "count": 4,
            "id": 13,
            "name": "Mohamed Ortega",
            "username": "user7"
        },
        "14": {
            "color": "#002e33",
            "count": 2,
            "id": 14,
            "name": "Miguel Molina",
            "username": "user8"
        },
        "15": {
            "color": "#FFFF00",
            "count": 3,
            "id": 15,
            "name": "Virginia Castro",
            "username": "user9"
        },
        "5": {
            "color": "",
            "count": 2,
            "id": 5,
            "name": "Administrator",
            "username": "admin"
        },
        "6": {
            "color": "#40826D",
            "count": 6,
            "id": 6,
            "name": "Vanesa Torres",
            "username": "user2114747470430251528"
        },
        "7": {
            "color": "#40826D",
            "count": 4,
            "id": 7,
            "name": "Bego\u00f1a Flores",
            "username": "user1"
        },
        "9": {
            "color": "#B6DA55",
            "count": 3,
            "id": 9,
            "name": "Catalina Fernandez",
            "username": "user3"
        }
    },
```

```
    "issues_per_priority": {
        "1": {
            "color": "#666666",
            "count": 9,
            "id": 1,
            "name": "Patch name"
        },
        "2": {
            "color": "#669933",
            "count": 13,
            "id": 2,
            "name": "Normal"
        },
        "3": {
            "color": "#CC0000",
            "count": 5,
            "id": 3,
            "name": "High"
        }
    },
    "issues_per_severity": {
        "1": {
            "color": "#666666",
            "count": 5,
            "id": 1,
            "name": "Patch name"
        },
        "2": {
            "color": "#669933",
            "count": 4,
            "id": 2,
            "name": "Minor"
        },
        "3": {
            "color": "#0000FF",
            "count": 9,
            "id": 3,
            "name": "Normal"
        },
        "4": {
            "color": "#FFA500",
            "count": 5,
            "id": 4,
            "name": "Important"
        },
        "5": {
            "color": "#CC0000",
            "count": 4,
            "id": 5,
            "name": "Critical"
        }
```

```
    },
    "issues_per_status": {
        "1": {
            "color": "#8C2318",
            "count": 7,
            "id": 1,
            "name": "Patch status name"
        },
        "2": {
            "color": "#5E8C6A",
            "count": 2,
            "id": 2,
            "name": "In progress"
        },
        "3": {
            "color": "#88A65E",
            "count": 4,
            "id": 3,
            "name": "Ready for test"
        },
        "4": {
            "color": "#BFB35A",
            "count": 3,
            "id": 4,
            "name": "Closed"
        },
        "5": {
            "color": "#89BAB4",
            "count": 3,
            "id": 5,
            "name": "Needs Info"
        },
        "6": {
            "color": "#CC0000",
            "count": 3,
            "id": 6,
            "name": "Rejected"
        },
        "7": {
            "color": "#666666",
            "count": 5,
            "id": 7,
            "name": "Postponed"
        }
    },
    "issues_per_type": {
        "1": {
            "color": "#89BAB4",
            "count": 12,
            "id": 1,
            "name": "Bug"
```

```
        },
        "2": {
            "color": "#ba89a8",
            "count": 7,
            "id": 2,
            "name": "Question"
        },
        "3": {
            "color": "#89a8ba",
            "count": 8,
            "id": 3,
            "name": "Enhancement"
        }
    },
    "last_four_weeks_days": {
        "by_open_closed": {
            "closed": [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                9,
                1
            ],
            "open": [
                0,
                0,
                0,
                0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                22,
                5
            ]
        },
        "by_priority": {
            "1": {
                "color": "#666666",
                "data": [
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
```

```
                0,
                0,
                0,
                0,
                0,
                9,
                5
            ],
            "id": 1,
            "name": "Patch name"
        },
        "2": {
            "color": "#669933",
            "data": [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                9,
                9
            ],
            "id": 2,
            "name": "Normal"
        },
        "3": {
            "color": "#CC0000",
            "data": [
                0,
                0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                4,
                4
            ],
            "id": 3,
            "name": "High"
        }
    },
    "by_severity": {
        "1": {
            "color": "#666666",
            "data": [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                5,
                2
            ],
            "id": 1,
            "name": "Patch name"
        },
        "2": {
            "color": "#669933",
            "data": [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                3,
                4
            ],
            "id": 2,
            "name": "Minor"
        },
```

```
            "3": {
                "color": "#0000FF",
                "data": [
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    5,
                    9
                ],
                "id": 3,
                "name": "Normal"
            },
            "4": {
                "color": "#FFA500",
                "data": [
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
                    0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                5,
                0
            ],
            "id": 4,
            "name": "Important"
        },
        "5": {
            "color": "#CC0000",
            "data": [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                4,
                3
            ],
```

```
                "id": 5,
                "name": "Critical"
            }
        },
        "by_status": {}
    },
    "opened_issues": 17,
    "total_issues": 27
}
```


48.6. Project tag colors data detail


```
{
    "ab": "#da2361",
    "accusamus": "#801cf7",
    "accusantium": null,
    "ad": null,
    "adipisci": null,
    "alias": "#cdb6fd",
    "aliquam": null,
    "aliquid": "#f01df5",
    "amet": "#db04fb",
    "animi": null,
    "aperiam": null,
    "architecto": null,
    "asperiores": null,
    "assumenda": null,
    "at": null,
    "atque": "#713547",
    "autem": null,
    "blanditiis": "#65026b",
    "commodi": "#3b70df",
    "consectetur": "#97176f",
    "consequatur": "#3ad7db",
    "consequuntur": "#ce24ec",
    "corporis": "#ed9c91",
    "corrupti": "#432493",
    "culpa": null,
    "cum": null,
    "cumque": null,
    "cupiditate": "#144bba",
    "delectus": null,
    "deleniti": "#6188db",
    "deserunt": "#e7b695",
    "dicta": "#939b44",
    "dignissimos": "#79b3c9",
    "distinctio": "#1f8960",
    "dolor": "#641bd9",
    "dolore": null,
```

```
    "dolorem": "#604860",
    "doloremque": null,
    "dolores": null,
    "doloribus": null,
    "dolorum": "#db7ec2",
    "ea": null,
    "eaque": null,
    "eius": "#860b86",
    "eligendi": "#5d8273",
    "enim": null,
    "error": null,
    "esse": "#d77661",
    "et": null,
    "eum": null,
    "eveniet": "#5d26b5",
    "ex": null,
    "excepturi": null,
    "exercitationem": null,
    "expedita": "#740c41",
    "explicabo": null,
    "facere": "#113f4a",
    "facilis": "#0f6b6b",
    "fuga": null,
    "fugiat": "#1c563a",
    "fugit": "#9345df",
    "harum": null,
    "hic": "#f75f0b",
    "id": null,
    "illo": null,
    "illum": null,
    "impedit": null,
    "in": "#af10ef",
    "incidunt": "#3099ec",
    "inventore": null,
    "ipsam": null,
    "ipsum": "#da3ba4",
    "iste": "#491b3a",
    "itaque": null,
    "iure": null,
    "iusto": "#3a10e8",
    "laborum": "#67eac4",
    "laudantium": "#9e3f1f",
    "libero": null,
    "magni": null,
    "maiores": null,
    "maxime": "#1acc29",
    "minima": "#f0048e",
    "minus": "#59b653",
    "modi": "#494e30",
    "mollitia": "#002e7f",
    "nam": "#ce4004",
```

```
    "natus": null,
    "necessitatibus": "#84e3b6",
    "nemo": "#e81498",
    "neque": "#150607",
    "nesciunt": "#4c8404",
    "nihil": "#98a352",
    "nisi": "#ef7fdc",
    "non": "#37031f",
    "nulla": null,
    "numquam": null,
    "obcaecati": null,
    "odio": null,
    "odit": null,
    "officia": "#c4f027",
    "officiis": null,
    "omnis": null,
    "optio": null,
    "pariatur": "#7b0e4e",
    "perferendis": "#999645",
    "perspiciatis": "#afb825",
    "placeat": null,
    "porro": null,
    "possimus": null,
    "praesentium": "#0cd131",
    "provident": null,
    "quae": "#d91a8b",
    "quaerat": "#0b4425",
    "quam": null,
    "quas": "#6e3390",
    "quasi": null,
    "qui": "#61f611",
    "quia": "#f53074",
    "quibusdam": "#c49ac2",
    "quis": null,
    "quisquam": "#ebca0b",
    "quo": null,
    "quod": "#0e5b24",
    "quos": null,
    "ratione": "#570ce3",
    "reiciendis": "#560ff6",
    "rem": "#688119",
    "repellat": "#807389",
    "repellendus": null,
    "reprehenderit": null,
    "repudiandae": "#3a2b71",
    "rerum": null,
    "saepe": null,
    "sed": "#c15b7b",
    "sequi": null,
    "sint": "#3b2404",
    "sit": "#abdcde",
```

```
    "soluta": "#1398ab",
    "sunt": null,
    "suscipit": "#38abf3",
    "tempora": null,
    "tempore": "#ae2670",
    "temporibus": null,
    "totam": "#560a5d",
    "ullam": "#98ad13",
    "unde": "#da2470",
    "ut": null,
    "vel": "#91e065",
    "velit": null,
    "veniam": null,
    "vero": null,
    "vitae": "#d9fe5e",
    "voluptate": null,
    "voluptates": null,
    "voluptatibus": null,
    "voluptatum": "#02d22f"
}
```


48.7. Project voter detail


```
{
    "full_name": "GitHub",
    "id": 2,
    "username": "github-ecf32e2347fd4afda1b4914414a965cf"
}
```


48.8. Project watcher detail


```
{
    "full_name": "Administrator",
    "id": 5,
    "username": "admin"
}
```


48.9. Membership detail

```
{
    "color": "#002e33",
    "created_at": "2020-07-02T11:56:20.444Z",
    "email": "user8@taigaio.demo",
    "full_name": "Miguel Molina",
    "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
    "id": 1,
    "invitation_extra_text": null,
    "invited_by": null,
    "is_admin": true,
    "is_owner": false,
    "is_user_active": true,
    "photo": null,
    "project": 1,
    "project_name": "Project Example 0",
    "project_slug": "project-0",
    "role": 5,
    "role_name": "Product Owner",
    "user": 14,
    "user_email": "user8@taigaio.demo",
    "user_order": 6
}
```


48.10. Role detail

```
{
    "computable": true,
    "id": 1,
    "members_count": 2,
    "name": "Patch name",
    "order": 10,
    "permissions": [
        "add_issue",
        "modify_issue",
        "delete_issue",
        "view_issues",
        "add_milestone",
        "modify_milestone",
        "delete_milestone",
        "view_milestones",
        "view_project",
        "add_task",
        "modify_task",
        "delete_task",
        "view_tasks",
        "add_us",
        "modify_us",
        "delete_us",
        "view_us",
        "add_wiki_page",
        "modify_wiki_page",
        "delete_wiki_page",
        "view_wiki_pages",
        "add_wiki_link",
        "delete_wiki_link",
        "view_wiki_links",
        "view_epics",
        "add_epic",
        "modify_epic",
        "delete_epic",
        "comment_epic",
        "comment_us",
        "comment_task",
        "comment_issue",
        "comment_wiki_page"
    ],
    "project": 1,
    "slug": "ux"
}
```


48.11. Milestone detail


```
{
```

```
    "closed": false,
    "closed_points": null,
    "created_date": "2020-05-08T11:56:21.173Z",
    "disponibility": 0.0,
    "estimated_finish": "2020-05-23",
    "estimated_start": "2020-05-08",
    "id": 1,
    "modified_date": "2020-07-03T08:41:05.099Z",
    "name": "Sprint 2",
    "order": 10,
    "owner": 6,
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Beta project patch",
        "slug": "project-0"
    },
    "slug": "sprint-2020-5-8",
    "total_points": 124.0,
    "user_stories": [
        {
            "assigned_to": 9,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Catalina Fernandez",
                "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
                "id": 9,
                "is_active": true,
                "photo": null,
                "username": "user3"
            },
            "backlog_order": 10,
            "blocked_note": "",
            "client_requirement": false,
            "created_date": "2020-07-02T11:56:21.217Z",
            "due_date": null,
            "due_date_reason": "",
            "due_date_status": "not_set",
            "epics": [
                {
                    "color": "#f57900",
                    "id": 15,
                    "project": {
                        "id": 3,
                        "name": "Project Example 2",
                        "slug": "project-2"
                    },
                    "ref": 121,
                    "subject": "Patching subject"
                }
```

```
            ],
            "external_reference": null,
            "finish_date": null,
            "id": 1,
            "is_blocked": false,
            "is_closed": false,
            "kanban_order": 10,
            "milestone": 1,
            "modified_date": "2020-07-03T08:40:36.879Z",
            "points": {
                "1": 12,
                "2": 2,
                "3": 5,
                "4": 5
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 1,
            "sprint_order": 10,
            "status": 4,
            "status_extra_info": {
                "color": "#fcc000",
                "is_closed": false,
                "name": "Ready for test"
            },
            "subject": "Patching subject",
            "team_requirement": false,
            "total_points": 44.0,
            "version": 2
        },
        {
            "assigned_to": 15,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Virginia Castro",
                "gravatar_id": "69b60d39a450e863609ae3546b12b360",
                "id": 15,
                "is_active": true,
                "photo": null,
                "username": "user9"
            },
            "backlog_order": 15,
            "blocked_note": "",
            "client_requirement": false,
            "created_date": "2020-07-02T11:56:22.518Z",
            "due_date": null,
```

```
            "due_date_reason": "",
            "due_date_status": "not_set",
            "epics": [
                {
                    "color": "#888a85",
                    "id": 5,
                    "project": {
                        "id": 1,
                        "name": "Beta project patch",
                        "slug": "project-0"
                    },
                    "ref": 68,
                    "subject": "Migrate to Python 3 and milk a beautiful cow"
                },
                {
                    "color": "#f57900",
                    "id": 15,
                    "project": {
                        "id": 3,
                        "name": "Project Example 2",
                        "slug": "project-2"
                    },
                    "ref": 121,
                    "subject": "Patching subject"
                }
            ],
            "external_reference": null,
            "finish_date": null,
            "id": 2,
            "is_blocked": false,
            "is_closed": false,
            "kanban_order": 15,
            "milestone": 1,
            "modified_date": "2020-07-02T11:56:22.697Z",
            "points": {
                "1": 11,
                "2": 4,
                "3": 7,
                "4": 3
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 6,
            "sprint_order": 15,
            "status": 2,
            "status_extra_info": {
```

```
                "color": "#ff8a84",
                "is_closed": false,
                "name": "Ready"
            },
            "subject": "Lighttpd x-sendfile support",
            "team_requirement": false,
            "total_points": 26.5,
            "version": 1
        },
        {
            "assigned_to": 7,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Bego\u00f1a Flores",
                "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
                "id": 7,
                "is_active": true,
                "photo": null,
                "username": "user1"
            },
            "backlog_order": 1593690983608,
            "blocked_note": "",
            "client_requirement": false,
            "created_date": "2020-07-02T11:56:23.608Z",
            "due_date": null,
            "due_date_reason": "",
            "due_date_status": "not_set",
            "epics": [
                {
                    "color": "#3465a4",
                    "id": 1,
                    "project": {
                        "id": 1,
                        "name": "Beta project patch",
                        "slug": "project-0"
                    },
                    "ref": 64,
                    "subject": "Added file copying and processing of images
(resizing)"
                },
                {
                    "color": "#ad7fa8",
                    "id": 2,
                    "project": {
                        "id": 1,
                        "name": "Beta project patch",
                        "slug": "project-0"
                    },
                    "ref": 65,
                    "subject": "Experimental: modular file types"
                },
```

```
                {
                    "color": "#888a85",
                    "id": 5,
                    "project": {
                        "id": 1,
                        "name": "Beta project patch",
                        "slug": "project-0"
                    },
                    "ref": 68,
                    "subject": "Migrate to Python 3 and milk a beautiful cow"
                }
            ],
            "external_reference": null,
            "finish_date": null,
            "id": 3,
            "is_blocked": false,
            "is_closed": false,
            "kanban_order": 1593690983608,
            "milestone": 1,
            "modified_date": "2020-07-02T11:56:23.834Z",
            "points": {
                "1": 6,
                "2": 12,
                "3": 9,
                "4": 3
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 10,
            "sprint_order": 1593690983609,
            "status": 4,
            "status_extra_info": {
                "color": "#fcc000",
                "is_closed": false,
                "name": "Ready for test"
            },
            "subject": "get_actions() does not check for 'delete_selected' in
actions",
            "team_requirement": false,
            "total_points": 53.5,
            "version": 1
        }
    ]
}
```

48.12. Milestone watcher detail


```
{
    "full_name": "Vanesa Torres",
    "id": 6,
    "username": "user2114747470430251528"
}
```


48.13. Milestone stats detail


```
{
    "completed_points": [],
    "completed_tasks": 4,
    "completed_userstories": 0,
    "days": [
        {
            "day": "2020-05-08",
            "name": 8,
            "open_points": 124.0,
            "optimal_points": 124.0
        },
        {
            "day": "2020-05-09",
            "name": 9,
            "open_points": 124.0,
            "optimal_points": 115.73333333333333
        },
        {
            "day": "2020-05-10",
            "name": 10,
            "open_points": 104.16666666666666,
            "optimal_points": 107.46666666666667
        },
        {
            "day": "2020-05-11",
            "name": 11,
            "open_points": 104.16666666666666,
            "optimal_points": 99.2
        },
        {
            "day": "2020-05-12",
            "name": 12,
            "open_points": 93.16666666666666,
            "optimal_points": 90.93333333333334
        },
        {
            "day": "2020-05-13",
            "name": 13,
```

```
            "open_points": 93.16666666666666,
            "optimal_points": 82.66666666666667
        },
        {
            "day": "2020-05-14",
            "name": 14,
            "open_points": 93.16666666666666,
            "optimal_points": 74.4
        },
        {
            "day": "2020-05-15",
            "name": 15,
            "open_points": 93.16666666666666,
            "optimal_points": 66.13333333333334
        },
        {
            "day": "2020-05-16",
            "name": 16,
            "open_points": 93.16666666666666,
            "optimal_points": 57.866666666666674
        },
        {
            "day": "2020-05-17",
            "name": 17,
            "open_points": 93.16666666666666,
            "optimal_points": 49.60000000000001
        },
        {
            "day": "2020-05-18",
            "name": 18,
            "open_points": 93.16666666666666,
            "optimal_points": 41.33333333333334
        },
        {
            "day": "2020-05-19",
            "name": 19,
            "open_points": 93.16666666666666,
            "optimal_points": 33.06666666666668
        },
        {
            "day": "2020-05-20",
            "name": 20,
            "open_points": 93.16666666666666,
            "optimal_points": 24.80000000000001
        },
        {
            "day": "2020-05-21",
            "name": 21,
            "open_points": 82.16666666666666,
            "optimal_points": 16.533333333333346
        },
```

```
        {
            "day": "2020-05-22",
            "name": 22,
            "open_points": 82.16666666666666,
            "optimal_points": 8.266666666666678
        },
        {
            "day": "2020-05-23",
            "name": 23,
            "open_points": 82.16666666666666,
            "optimal_points": 1.0658141036401503e-14
        }
    ],
    "estimated_finish": "2020-05-23",
    "estimated_start": "2020-05-08",
    "iocaine_doses": 0,
    "name": "Sprint 2020-5-8",
    "total_points": {
        "1": 63.0,
        "2": 41.0,
        "3": 17.0,
        "4": 3.0
    },
    "total_tasks": 12,
    "total_userstories": 3
}
```


48.14. Epic detail


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "color": "#3465a4",
    "comment": "",
    "created_date": "2020-07-02T11:56:44.275Z",
    "description": "Voluptatem nihil unde ipsam at ipsum praesentium in labore, quo
nobis asperiores ipsam. Obcaecati doloribus voluptatem sint mollitia ea deserunt
```

```
totam, odit dicta repellat doloremque voluptate necessitatibus ipsa, molestiae
deserunt itaque aliquam tenetur consectetur officia consequatur repellat quisquam
nulla rerum? Iste suscipit quas incidunt cumque enim consectetur illo eum, nostrum
modi voluptates doloremque illo nemo, nesciunt culpa pariatur dolor sapiente nobis
repellendus itaque molestiae accusamus adipisci, dolorum quod tempore accusantium
saepe eius placeat iure ullam, dolorum aliquid doloribus animi tenetur optio?",
    "description_html": "<p>Voluptatem nihil unde ipsam at ipsum praesentium in
labore, quo nobis asperiores ipsam. Obcaecati doloribus voluptatem sint mollitia ea
deserunt totam, odit dicta repellat doloremque voluptate necessitatibus ipsa,
molestiae deserunt itaque aliquam tenetur consectetur officia consequatur repellat
quisquam nulla rerum? Iste suscipit quas incidunt cumque enim consectetur illo eum,
nostrum modi voluptates doloremque illo nemo, nesciunt culpa pariatur dolor sapiente
nobis repellendus itaque molestiae accusamus adipisci, dolorum quod tempore
accusantium saepe eius placeat iure ullam, dolorum aliquid doloribus animi tenetur
optio?</p>",
    "epics_order": 1593691004275,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": true,
    "is_watcher": false,
    "modified_date": "2020-07-02T11:56:44.453Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 65,
            "subject": "Experimental: modular file types"
        },
        "previous": {
            "id": 27,
            "ref": 71,
            "subject": "New epic"
        }
    },
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
```

```
    "ref": 64,
    "status": 4,
    "status_extra_info": {
        "color": "#fcc000",
        "is_closed": false,
        "name": "Ready for test"
    },
    "subject": "Added file copying and processing of images (resizing)",
    "tags": [
        [
            "eligendi",
            "#5d8273"
        ]
    ],
    "team_requirement": false,
    "total_voters": 3,
    "total_watchers": 2,
    "user_stories_counts": {
        "progress": 1.5,
        "total": 8
    },
    "version": 1,
    "watchers": [
        12,
        14
    ]
}
```


48.15. Epic detail (GET)


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "color": "#3465a4",
    "comment": "",
    "created_date": "2020-07-02T11:56:44.275Z",
    "description": "Voluptatem nihil unde ipsam at ipsum praesentium in labore, quo
```

```
nobis asperiores ipsam. Obcaecati doloribus voluptatem sint mollitia ea deserunt
totam, odit dicta repellat doloremque voluptate necessitatibus ipsa, molestiae
deserunt itaque aliquam tenetur consectetur officia consequatur repellat quisquam
nulla rerum? Iste suscipit quas incidunt cumque enim consectetur illo eum, nostrum
modi voluptates doloremque illo nemo, nesciunt culpa pariatur dolor sapiente nobis
repellendus itaque molestiae accusamus adipisci, dolorum quod tempore accusantium
saepe eius placeat iure ullam, dolorum aliquid doloribus animi tenetur optio?",
    "description_html": "<p>Voluptatem nihil unde ipsam at ipsum praesentium in
labore, quo nobis asperiores ipsam. Obcaecati doloribus voluptatem sint mollitia ea
deserunt totam, odit dicta repellat doloremque voluptate necessitatibus ipsa,
molestiae deserunt itaque aliquam tenetur consectetur officia consequatur repellat
quisquam nulla rerum? Iste suscipit quas incidunt cumque enim consectetur illo eum,
nostrum modi voluptates doloremque illo nemo, nesciunt culpa pariatur dolor sapiente
nobis repellendus itaque molestiae accusamus adipisci, dolorum quod tempore
accusantium saepe eius placeat iure ullam, dolorum aliquid doloribus animi tenetur
optio?</p>",
    "epics_order": 1593691004275,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": true,
    "is_watcher": false,
    "modified_date": "2020-07-02T11:56:44.453Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 65,
            "subject": "Experimental: modular file types"
        },
        "previous": {
            "id": 27,
            "ref": 71,
            "subject": "New epic"
        }
    },
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
```

```
    },
    "ref": 64,
    "status": 4,
    "status_extra_info": {
        "color": "#fcc000",
        "is_closed": false,
        "name": "Ready for test"
    },
    "subject": "Added file copying and processing of images (resizing)",
    "tags": [
        [
            "eligendi",
            "#5d8273"
        ]
    ],
    "team_requirement": false,
    "total_voters": 3,
    "total_watchers": 2,
    "user_stories_counts": {
        "progress": 1.5,
        "total": 8
    },
    "version": 1,
    "watchers": [
        12,
        14
    ]
}
```


48.16. Epic detail (LIST)


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "client_requirement": false,
    "color": "#3465a4",
    "created_date": "2020-07-02T11:56:44.275Z",
    "epics_order": 1593691004275,
    "id": 1,
```

```
    "is_blocked": false,
    "is_closed": false,
    "is_voter": true,
    "is_watcher": false,
    "modified_date": "2020-07-02T11:56:44.453Z",
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 64,
    "status": 4,
    "status_extra_info": {
        "color": "#fcc000",
        "is_closed": false,
        "name": "Ready for test"
    },
    "subject": "Added file copying and processing of images (resizing)",
    "tags": [
        [
            "eligendi",
            "#5d8273"
        ]
    ],
    "team_requirement": false,
    "total_voters": 3,
    "total_watchers": 2,
    "user_stories_counts": {
        "progress": 1.5,
        "total": 8
    },
    "version": 1,
    "watchers": [
        12,
        14
    ]
}
```

48.17. Epic filters data detail


```
{
    "assigned_to": [
        {
            "count": 5,
            "full_name": "",
            "id": null
        },
        {
            "count": 1,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 0,
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
            "count": 0,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 1,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 0,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 0,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 0,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 2,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
```

```
            "count": 1,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 2,
            "full_name": "Virginia Castro",
            "id": 15
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "owners": [
        {
            "count": 1,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 1,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 1,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 3,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 5,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 1,
            "full_name": "Virginia Castro",
            "id": 15
        }
    ],
    "statuses": [
        {
            "color": "#999999",
            "count": 5,
```

```
            "id": 1,
            "name": "New",
            "order": 1
        },
        {
            "color": "#ff8a84",
            "count": 4,
            "id": 2,
            "name": "Ready",
            "order": 2
        },
        {
            "color": "#ff9900",
            "count": 1,
            "id": 3,
            "name": "In progress",
            "order": 3
        },
        {
            "color": "#fcc000",
            "count": 2,
            "id": 4,
            "name": "Ready for test",
            "order": 4
        },
        {
            "color": "#669900",
            "count": 0,
            "id": 5,
            "name": "Done",
            "order": 5
        }
    ],
    "tags": [
        {
            "color": "#da2361",
            "count": 0,
            "name": "ab"
        },
        {
            "color": "#801cf7",
            "count": 0,
            "name": "accusamus"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusantium"
        },
        {
            "color": null,
```

```
            "count": 0,
            "name": "ad"
        },
        {
            "color": "#cdb6fd",
            "count": 0,
            "name": "alias"
        },
        {
            "color": null,
            "count": 1,
            "name": "aliquam"
        },
        {
            "color": "#f01df5",
            "count": 0,
            "name": "aliquid"
        },
        {
            "color": "#db04fb",
            "count": 0,
            "name": "amet"
        },
        {
            "color": null,
            "count": 0,
            "name": "animi"
        },
        {
            "color": null,
            "count": 0,
            "name": "aperiam"
        },
        {
            "color": null,
            "count": 0,
            "name": "architecto"
        },
        {
            "color": null,
            "count": 0,
            "name": "asperiores"
        },
        {
            "color": null,
            "count": 0,
            "name": "assumenda"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "at"
        },
        {
            "color": "#713547",
            "count": 0,
            "name": "atque"
        },
        {
            "color": null,
            "count": 0,
            "name": "autem"
        },
        {
            "color": "#65026b",
            "count": 0,
            "name": "blanditiis"
        },
        {
            "color": "#3b70df",
            "count": 0,
            "name": "commodi"
        },
        {
            "color": "#97176f",
            "count": 0,
            "name": "consectetur"
        },
        {
            "color": "#3ad7db",
            "count": 0,
            "name": "consequatur"
        },
        {
            "color": "#ce24ec",
            "count": 0,
            "name": "consequuntur"
        },
        {
            "color": "#ed9c91",
            "count": 0,
            "name": "corporis"
        },
        {
            "color": "#432493",
            "count": 0,
            "name": "corrupti"
        },
        {
            "color": null,
            "count": 1,
            "name": "culpa"
```

```
        },
        {
            "color": null,
            "count": 0,
            "name": "cum"
        },
        {
            "color": null,
            "count": 0,
            "name": "cumque"
        },
        {
            "color": "#144bba",
            "count": 0,
            "name": "cupiditate"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": null,
            "count": 0,
            "name": "delectus"
        },
        {
            "color": "#6188db",
            "count": 0,
            "name": "deleniti"
        },
        {
            "color": "#e7b695",
            "count": 0,
            "name": "deserunt"
        },
        {
            "color": "#939b44",
            "count": 1,
            "name": "dicta"
        },
        {
            "color": "#79b3c9",
            "count": 0,
            "name": "dignissimos"
        },
        {
            "color": "#1f8960",
            "count": 0,
            "name": "distinctio"
        },
```

```
        {
            "color": "#641bd9",
            "count": 0,
            "name": "dolor"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolore"
        },
        {
            "color": "#604860",
            "count": 0,
            "name": "dolorem"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloremque"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolores"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloribus"
        },
        {
            "color": "#db7ec2",
            "count": 0,
            "name": "dolorum"
        },
        {
            "color": null,
            "count": 1,
            "name": "ea"
        },
        {
            "color": null,
            "count": 0,
            "name": "eaque"
        },
        {
            "color": "#860b86",
            "count": 0,
            "name": "eius"
        },
        {
```

```
            "color": "#5d8273",
            "count": 1,
            "name": "eligendi"
        },
        {
            "color": null,
            "count": 1,
            "name": "enim"
        },
        {
            "color": null,
            "count": 0,
            "name": "error"
        },
        {
            "color": "#d77661",
            "count": 0,
            "name": "esse"
        },
        {
            "color": null,
            "count": 0,
            "name": "et"
        },
        {
            "color": null,
            "count": 0,
            "name": "eum"
        },
        {
            "color": "#5d26b5",
            "count": 0,
            "name": "eveniet"
        },
        {
            "color": null,
            "count": 0,
            "name": "ex"
        },
        {
            "color": null,
            "count": 0,
            "name": "excepturi"
        },
        {
            "color": null,
            "count": 1,
            "name": "exercitationem"
        },
        {
            "color": "#740c41",
```

```
            "count": 0,
            "name": "expedita"
        },
        {
            "color": null,
            "count": 0,
            "name": "explicabo"
        },
        {
            "color": "#113f4a",
            "count": 0,
            "name": "facere"
        },
        {
            "color": "#0f6b6b",
            "count": 1,
            "name": "facilis"
        },
        {
            "color": null,
            "count": 0,
            "name": "fuga"
        },
        {
            "color": "#1c563a",
            "count": 0,
            "name": "fugiat"
        },
        {
            "color": "#9345df",
            "count": 0,
            "name": "fugit"
        },
        {
            "color": null,
            "count": 0,
            "name": "harum"
        },
        {
            "color": "#f75f0b",
            "count": 0,
            "name": "hic"
        },
        {
            "color": null,
            "count": 0,
            "name": "id"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "illo"
        },
        {
            "color": null,
            "count": 0,
            "name": "illum"
        },
        {
            "color": null,
            "count": 0,
            "name": "impedit"
        },
        {
            "color": "#af10ef",
            "count": 0,
            "name": "in"
        },
        {
            "color": "#3099ec",
            "count": 0,
            "name": "incidunt"
        },
        {
            "color": null,
            "count": 0,
            "name": "inventore"
        },
        {
            "color": null,
            "count": 0,
            "name": "ipsam"
        },
        {
            "color": "#da3ba4",
            "count": 1,
            "name": "ipsum"
        },
        {
            "color": "#491b3a",
            "count": 0,
            "name": "iste"
        },
        {
            "color": null,
            "count": 0,
            "name": "itaque"
        },
        {
            "color": null,
            "count": 0,
            "name": "iure"
```

```
        },
        {
            "color": "#3a10e8",
            "count": 0,
            "name": "iusto"
        },
        {
            "color": "#67eac4",
            "count": 1,
            "name": "laborum"
        },
        {
            "color": "#9e3f1f",
            "count": 0,
            "name": "laudantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "libero"
        },
        {
            "color": null,
            "count": 0,
            "name": "magni"
        },
        {
            "color": null,
            "count": 0,
            "name": "maiores"
        },
        {
            "color": "#1acc29",
            "count": 0,
            "name": "maxime"
        },
        {
            "color": "#f0048e",
            "count": 0,
            "name": "minima"
        },
        {
            "color": "#59b653",
            "count": 0,
            "name": "minus"
        },
        {
            "color": "#494e30",
            "count": 0,
            "name": "modi"
        },
```

```
        {
            "color": "#002e7f",
            "count": 0,
            "name": "mollitia"
        },
        {
            "color": "#ce4004",
            "count": 0,
            "name": "nam"
        },
        {
            "color": null,
            "count": 0,
            "name": "natus"
        },
        {
            "color": "#84e3b6",
            "count": 0,
            "name": "necessitatibus"
        },
        {
            "color": "#e81498",
            "count": 0,
            "name": "nemo"
        },
        {
            "color": "#150607",
            "count": 0,
            "name": "neque"
        },
        {
            "color": "#4c8404",
            "count": 0,
            "name": "nesciunt"
        },
        {
            "color": "#98a352",
            "count": 0,
            "name": "nihil"
        },
        {
            "color": "#ef7fdc",
            "count": 0,
            "name": "nisi"
        },
        {
            "color": "#37031f",
            "count": 0,
            "name": "non"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "nulla"
        },
        {
            "color": null,
            "count": 0,
            "name": "numquam"
        },
        {
            "color": null,
            "count": 0,
            "name": "obcaecati"
        },
        {
            "color": null,
            "count": 0,
            "name": "odio"
        },
        {
            "color": null,
            "count": 0,
            "name": "odit"
        },
        {
            "color": "#c4f027",
            "count": 0,
            "name": "officia"
        },
        {
            "color": null,
            "count": 0,
            "name": "officiis"
        },
        {
            "color": null,
            "count": 0,
            "name": "omnis"
        },
        {
            "color": null,
            "count": 0,
            "name": "optio"
        },
        {
            "color": "#7b0e4e",
            "count": 1,
            "name": "pariatur"
        },
        {
            "color": "#999645",
```

```
            "count": 0,
            "name": "perferendis"
        },
        {
            "color": "#afb825",
            "count": 0,
            "name": "perspiciatis"
        },
        {
            "color": null,
            "count": 0,
            "name": "placeat"
        },
        {
            "color": null,
            "count": 0,
            "name": "porro"
        },
        {
            "color": null,
            "count": 1,
            "name": "possimus"
        },
        {
            "color": "#0cd131",
            "count": 0,
            "name": "praesentium"
        },
        {
            "color": null,
            "count": 1,
            "name": "provident"
        },
        {
            "color": "#d91a8b",
            "count": 0,
            "name": "quae"
        },
        {
            "color": "#0b4425",
            "count": 0,
            "name": "quaerat"
        },
        {
            "color": null,
            "count": 0,
            "name": "quam"
        },
        {
            "color": "#6e3390",
            "count": 0,
```

```
            "name": "quas"
        },
        {
            "color": null,
            "count": 0,
            "name": "quasi"
        },
        {
            "color": "#61f611",
            "count": 0,
            "name": "qui"
        },
        {
            "color": "#f53074",
            "count": 0,
            "name": "quia"
        },
        {
            "color": "#c49ac2",
            "count": 0,
            "name": "quibusdam"
        },
        {
            "color": null,
            "count": 0,
            "name": "quis"
        },
        {
            "color": "#ebca0b",
            "count": 0,
            "name": "quisquam"
        },
        {
            "color": null,
            "count": 0,
            "name": "quo"
        },
        {
            "color": "#0e5b24",
            "count": 0,
            "name": "quod"
        },
        {
            "color": null,
            "count": 0,
            "name": "quos"
        },
        {
            "color": "#570ce3",
            "count": 0,
            "name": "ratione"
```

```
        },
        {
            "color": "#560ff6",
            "count": 0,
            "name": "reiciendis"
        },
        {
            "color": "#688119",
            "count": 0,
            "name": "rem"
        },
        {
            "color": "#807389",
            "count": 1,
            "name": "repellat"
        },
        {
            "color": null,
            "count": 0,
            "name": "repellendus"
        },
        {
            "color": null,
            "count": 0,
            "name": "reprehenderit"
        },
        {
            "color": "#3a2b71",
            "count": 0,
            "name": "repudiandae"
        },
        {
            "color": null,
            "count": 0,
            "name": "rerum"
        },
        {
            "color": null,
            "count": 0,
            "name": "saepe"
        },
        {
            "color": "#c15b7b",
            "count": 0,
            "name": "sed"
        },
        {
            "color": null,
            "count": 0,
            "name": "sequi"
        },
```

```
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
        {
            "color": "#3b2404",
            "count": 0,
            "name": "sint"
        },
        {
            "color": "#abdcde",
            "count": 1,
            "name": "sit"
        },
        {
            "color": "#1398ab",
            "count": 0,
            "name": "soluta"
        },
        {
            "color": null,
            "count": 0,
            "name": "sunt"
        },
        {
            "color": "#38abf3",
            "count": 0,
            "name": "suscipit"
        },
        {
            "color": null,
            "count": 1,
            "name": "tempora"
        },
        {
            "color": "#ae2670",
            "count": 0,
            "name": "tempore"
        },
        {
            "color": null,
            "count": 0,
            "name": "temporibus"
        },
        {
            "color": "#560a5d",
            "count": 1,
            "name": "totam"
        },
        {
```

```
            "color": "#98ad13",
            "count": 0,
            "name": "ullam"
        },
        {
            "color": "#da2470",
            "count": 0,
            "name": "unde"
        },
        {
            "color": "#91e065",
            "count": 0,
            "name": "vel"
        },
        {
            "color": null,
            "count": 0,
            "name": "velit"
        },
        {
            "color": null,
            "count": 0,
            "name": "veniam"
        },
        {
            "color": null,
            "count": 0,
            "name": "vero"
        },
        {
            "color": "#d9fe5e",
            "count": 0,
            "name": "vitae"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptate"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptates"
        },
        {
            "color": "#02d22f",
            "count": 0,
            "name": "voluptatum"
        }
    ]
}
```

48.18. Epic voter detail

```
{
    "full_name": "Bego\u00f1a Flores",
    "id": 7,
    "username": "user1"
}
```

48.19. Epic watcher detail

```
{
    "full_name": "BitBucket",
    "id": 1,
    "username": "bitbucket-39e1bd53be764862bb63edb881e16f8a"
}
```

48.20. Epic status detail

```
{
    "color": "#999999",
    "id": 1,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name"
}
```

48.21. Epic custom attribute detail

```
{
    "created_date": "2020-07-02T11:57:22.124Z",
    "description": "nesciunt consectetur culpa ullam harum fugit veritatis eius
dolorem assumenda",
    "extra": null,
    "id": 14,
    "modified_date": "2020-07-03T08:40:34.667Z",
    "name": "Duration 1",
    "order": 1,
    "project": 3,
    "type": "url"
}
```

48.22. Epic custom attributes values detail


```
{
    "attributes_values": {
        "14": "240 min"
    },
    "epic": 15,
    "version": 2
}
```


48.23. Epic related user story detail


```
{
    "epic": 15,
    "order": 100,
    "user_story": 2
}
```


48.24. User story detail


```
{
    "assigned_to": 9,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Catalina Fernandez",
        "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
        "id": 9,
        "is_active": true,
        "photo": null,
        "username": "user3"
    },
    "assigned_users": [
        9
    ],
    "attachments": [],
    "backlog_order": 1593690981217,
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "comment": "",
    "created_date": "2020-07-02T11:56:21.217Z",
    "description": "Expedita quos dolore adipisci animi harum hic ut sed fugiat, harum
voluptates iste obcaecati totam eos deleniti, dolores quo nemo quibusdam? Dolorum
expedita veniam eveniet numquam officiis laudantium? Voluptate voluptas molestiae
necessitatibus id facilis animi odit dolores enim, maxime cupiditate perspiciatis modi
odit eveniet ullam. Consequuntur doloremque sit deserunt earum corrupti architecto
```

```
eveniet quis amet obcaecati, non ex amet rerum aut facere rem vel delectus fugit
tempora nobis, ut debitis ipsam alias iure tenetur, quae voluptatem illum tenetur
porro voluptas doloribus, obcaecati corporis porro neque itaque temporibus nulla unde
quas tempora.",
    "description_html": "<p>Expedita quos dolore adipisci animi harum hic ut sed
fugiat, harum voluptates iste obcaecati totam eos deleniti, dolores quo nemo
quibusdam? Dolorum expedita veniam eveniet numquam officiis laudantium? Voluptate
voluptas molestiae necessitatibus id facilis animi odit dolores enim, maxime
cupiditate perspiciatis modi odit eveniet ullam. Consequuntur doloremque sit deserunt
earum corrupti architecto eveniet quis amet obcaecati, non ex amet rerum aut facere
rem vel delectus fugit tempora nobis, ut debitis ipsam alias iure tenetur, quae
voluptatem illum tenetur porro voluptas doloribus, obcaecati corporis porro neque
itaque temporibus nulla unde quas tempora.</p>",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "epic_order": null,
    "epics": [
        {
            "color": "#f57900",
            "id": 15,
            "project": {
                "id": 3,
                "name": "Project Example 2",
                "slug": "project-2"
            },
            "ref": 121,
            "subject": "Patching subject"
        }
    ],
    "external_reference": null,
    "finish_date": null,
    "generated_from_issue": null,
    "generated_from_task": null,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "kanban_order": 1593690981217,
    "milestone": 1,
    "milestone_name": "Sprint 2020-5-8",
    "milestone_slug": "sprint-2020-5-8",
    "modified_date": "2020-07-03T08:40:36.879Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 6,
            "subject": "Lighttpd x-sendfile support"
        },
        "previous": {
```

```
            "id": 142,
            "ref": 79,
            "subject": "Customer personal data"
        }
    },
    "origin_issue": null,
    "origin_task": null,
    "owner": 6,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Vanesa Torres",
        "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user2114747470430251528"
    },
    "points": {
        "1": 12,
        "2": 2,
        "3": 5,
        "4": 5
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 1,
    "sprint_order": 10,
    "status": 4,
    "status_extra_info": {
        "color": "#fcc000",
        "is_closed": false,
        "name": "Ready for test"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "cum",
            null
        ]
    ],
    "tasks": [],
    "team_requirement": false,
    "total_attachments": 4,
    "total_comments": 1,
    "total_points": 44.0,
    "total_voters": 1,
```

```
    "total_watchers": 0,
    "tribe_gig": null,
    "version": 2,
    "watchers": []
}
```


48.25. User story detail (GET)


```
{
    "assigned_to": 9,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Catalina Fernandez",
        "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
        "id": 9,
        "is_active": true,
        "photo": null,
        "username": "user3"
    },
    "assigned_users": [
        9
    ],
    "attachments": [],
    "backlog_order": 1593690981217,
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "comment": "",
    "created_date": "2020-07-02T11:56:21.217Z",
    "description": "Expedita quos dolore adipisci animi harum hic ut sed fugiat, harum
voluptates iste obcaecati totam eos deleniti, dolores quo nemo quibusdam? Dolorum
expedita veniam eveniet numquam officiis laudantium? Voluptate voluptas molestiae
necessitatibus id facilis animi odit dolores enim, maxime cupiditate perspiciatis modi
odit eveniet ullam. Consequuntur doloremque sit deserunt earum corrupti architecto
eveniet quis amet obcaecati, non ex amet rerum aut facere rem vel delectus fugit
tempora nobis, ut debitis ipsam alias iure tenetur, quae voluptatem illum tenetur
porro voluptas doloribus, obcaecati corporis porro neque itaque temporibus nulla unde
quas tempora.",
    "description_html": "<p>Expedita quos dolore adipisci animi harum hic ut sed
fugiat, harum voluptates iste obcaecati totam eos deleniti, dolores quo nemo
quibusdam? Dolorum expedita veniam eveniet numquam officiis laudantium? Voluptate
voluptas molestiae necessitatibus id facilis animi odit dolores enim, maxime
cupiditate perspiciatis modi odit eveniet ullam. Consequuntur doloremque sit deserunt
earum corrupti architecto eveniet quis amet obcaecati, non ex amet rerum aut facere
rem vel delectus fugit tempora nobis, ut debitis ipsam alias iure tenetur, quae
voluptatem illum tenetur porro voluptas doloribus, obcaecati corporis porro neque
itaque temporibus nulla unde quas tempora.</p>",
    "due_date": null,
    "due_date_reason": "",
```

```
    "due_date_status": "not_set",
    "epic_order": null,
    "epics": [
        {
            "color": "#f57900",
            "id": 15,
            "project": {
                "id": 3,
                "name": "Project Example 2",
                "slug": "project-2"
            },
            "ref": 121,
            "subject": "Patching subject"
        }
    ],
    "external_reference": null,
    "finish_date": null,
    "generated_from_issue": null,
    "generated_from_task": null,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "kanban_order": 1593690981217,
    "milestone": 1,
    "milestone_name": "Sprint 2020-5-8",
    "milestone_slug": "sprint-2020-5-8",
    "modified_date": "2020-07-03T08:40:36.879Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 6,
            "subject": "Lighttpd x-sendfile support"
        },
        "previous": {
            "id": 142,
            "ref": 79,
            "subject": "Customer personal data"
        }
    },
    "origin_issue": null,
    "origin_task": null,
    "owner": 6,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Vanesa Torres",
        "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
        "id": 6,
        "is_active": true,
        "photo": null,
```

```
        "username": "user2114747470430251528"
    },
    "points": {
        "1": 12,
        "2": 2,
        "3": 5,
        "4": 5
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 1,
    "sprint_order": 10,
    "status": 4,
    "status_extra_info": {
        "color": "#fcc000",
        "is_closed": false,
        "name": "Ready for test"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "cum",
            null
        ]
    ],
    "tasks": [],
    "team_requirement": false,
    "total_attachments": 4,
    "total_comments": 1,
    "total_points": 44.0,
    "total_voters": 1,
    "total_watchers": 0,
    "tribe_gig": null,
    "version": 2,
    "watchers": []
}
```


NOTE neighbors is a read only field

48.26. User story detail (LIST)


```
{
    "assigned_to": null,
    "assigned_to_extra_info": null,
```

```
    "assigned_users": [],
    "attachments": [],
    "backlog_order": 2,
    "blocked_note": "",
    "client_requirement": false,
    "comment": "",
    "created_date": "2020-07-03T08:40:37.245Z",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "epic_order": null,
    "epics": null,
    "external_reference": null,
    "finish_date": null,
    "generated_from_issue": null,
    "generated_from_task": null,
    "id": 142,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "kanban_order": 37,
    "milestone": null,
    "milestone_name": null,
    "milestone_slug": null,
    "modified_date": "2020-07-03T08:40:37.249Z",
    "origin_issue": null,
    "origin_task": null,
    "owner": 6,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Vanesa Torres",
        "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user2114747470430251528"
    },
    "points": {
        "1": 4,
        "2": 3,
        "3": 2,
        "4": 1,
        "43": 1,
        "44": 1
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
```

```
        "slug": "project-0"
    },
    "ref": 79,
    "sprint_order": 2,
    "status": 2,
    "status_extra_info": {
        "color": "#ff8a84",
        "is_closed": false,
        "name": "Ready"
    },
    "subject": "Customer personal data",
    "tags": [
        [
            "service catalog",
            null
        ],
        [
            "customer",
            null
        ]
    ],
    "tasks": [],
    "team_requirement": false,
    "total_attachments": 0,
    "total_comments": 0,
    "total_points": 1.5,
    "total_voters": 0,
    "total_watchers": 0,
    "tribe_gig": null,
    "version": 1,
    "watchers": []
}
```


48.27. Issue filters data detail


```
{
    "assigned_to": [
        {
            "count": 14,
            "full_name": "",
            "id": null
        },
        {
            "count": 0,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 2,
```

```
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 0,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 0,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 1,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 1,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 3,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 2,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 2,
            "full_name": "Virginia Castro",
            "id": 15
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "assigned_users": [
        {
```

```
            "count": 14,
            "full_name": "",
            "id": null
        },
        {
            "count": 0,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 2,
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 0,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 0,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 1,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 1,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 3,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 2,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 2,
```

```
            "full_name": "Virginia Castro",
            "id": 15
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "epics": [
        {
            "count": 8,
            "id": null,
            "order": 0,
            "ref": null,
            "subject": null
        },
        {
            "count": 0,
            "id": 27,
            "order": 2,
            "ref": 71,
            "subject": "New epic"
        },
        {
            "count": 8,
            "id": 1,
            "order": 1593691004275,
            "ref": 64,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "count": 3,
            "id": 2,
            "order": 1593691004514,
            "ref": 65,
            "subject": "Experimental: modular file types"
        },
        {
            "count": 2,
            "id": 3,
            "order": 1593691004798,
            "ref": 66,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "count": 0,
            "id": 4,
            "order": 1593691005001,
            "ref": 67,
            "subject": "Feature/improved image admin"
```

```
        },
        {
            "count": 9,
            "id": 5,
            "order": 1593691005198,
            "ref": 68,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "count": 5,
            "id": 6,
            "order": 1593691005461,
            "ref": 69,
            "subject": "Experimental: modular file types"
        },
        {
            "count": 0,
            "id": 7,
            "order": 1593691005723,
            "ref": 70,
            "subject": "Add tests for bulk operations"
        },
        {
            "count": 0,
            "id": 28,
            "order": 1593765630980,
            "ref": 72,
            "subject": "New epic"
        },
        {
            "count": 0,
            "id": 29,
            "order": 1593765631974,
            "ref": 73,
            "subject": "EPIC 1"
        },
        {
            "count": 0,
            "id": 30,
            "order": 1593765631974,
            "ref": 74,
            "subject": "EPIC 2"
        },
        {
            "count": 0,
            "id": 31,
            "order": 1593765631974,
            "ref": 75,
            "subject": "EPIC 3"
        }
    ],
```

```
    "owners": [
        {
            "count": 2,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 1,
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 3,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 1,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 3,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 1,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 8,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 2,
            "full_name": "Virginia Castro",
            "id": 15
        }
    ],
    "roles": [
        {
            "color": null,
            "count": 2,
```

```
            "id": 1,
            "name": "Patch name",
            "order": 10
        },
        {
            "color": null,
            "count": 0,
            "id": 44,
            "name": "New role name",
            "order": 10
        },
        {
            "color": null,
            "count": 0,
            "id": 43,
            "name": "New role",
            "order": 10
        },
        {
            "color": null,
            "count": 2,
            "id": 2,
            "name": "Design",
            "order": 20
        },
        {
            "color": null,
            "count": 0,
            "id": 3,
            "name": "Front",
            "order": 30
        },
        {
            "color": null,
            "count": 1,
            "id": 4,
            "name": "Back",
            "order": 40
        },
        {
            "color": null,
            "count": 2,
            "id": 5,
            "name": "Product Owner",
            "order": 50
        },
        {
            "color": null,
            "count": 5,
            "id": 6,
            "name": "Stakeholder",
```

```
            "order": 60
        }
    ],
    "statuses": [
        {
            "color": "#999999",
            "count": 8,
            "id": 1,
            "name": "New",
            "order": 1
        },
        {
            "color": "#ff8a84",
            "count": 4,
            "id": 2,
            "name": "Ready",
            "order": 2
        },
        {
            "color": "#ff9900",
            "count": 5,
            "id": 3,
            "name": "In progress",
            "order": 3
        },
        {
            "color": "#fcc000",
            "count": 5,
            "id": 4,
            "name": "Ready for test",
            "order": 4
        },
        {
            "color": "#669900",
            "count": 0,
            "id": 5,
            "name": "Done",
            "order": 5
        },
        {
            "color": "#5c3566",
            "count": 0,
            "id": 6,
            "name": "Archived",
            "order": 6
        }
    ],
    "tags": [
        {
            "color": "#da2361",
            "count": 0,
```

```
            "name": "ab"
        },
        {
            "color": "#801cf7",
            "count": 0,
            "name": "accusamus"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusantium"
        },
        {
            "color": null,
            "count": 1,
            "name": "ad"
        },
        {
            "color": "#cdb6fd",
            "count": 0,
            "name": "alias"
        },
        {
            "color": null,
            "count": 0,
            "name": "aliquam"
        },
        {
            "color": "#f01df5",
            "count": 0,
            "name": "aliquid"
        },
        {
            "color": "#db04fb",
            "count": 0,
            "name": "amet"
        },
        {
            "color": null,
            "count": 0,
            "name": "animi"
        },
        {
            "color": null,
            "count": 0,
            "name": "aperiam"
        },
        {
            "color": null,
            "count": 0,
            "name": "architecto"
```

```
        },
        {
            "color": null,
            "count": 0,
            "name": "asperiores"
        },
        {
            "color": null,
            "count": 1,
            "name": "assumenda"
        },
        {
            "color": null,
            "count": 0,
            "name": "at"
        },
        {
            "color": "#713547",
            "count": 1,
            "name": "atque"
        },
        {
            "color": null,
            "count": 1,
            "name": "autem"
        },
        {
            "color": "#65026b",
            "count": 1,
            "name": "blanditiis"
        },
        {
            "color": "#3b70df",
            "count": 0,
            "name": "commodi"
        },
        {
            "color": "#97176f",
            "count": 0,
            "name": "consectetur"
        },
        {
            "color": "#3ad7db",
            "count": 0,
            "name": "consequatur"
        },
        {
            "color": "#ce24ec",
            "count": 0,
            "name": "consequuntur"
        },
```

```
        {
            "color": "#ed9c91",
            "count": 0,
            "name": "corporis"
        },
        {
            "color": "#432493",
            "count": 0,
            "name": "corrupti"
        },
        {
            "color": null,
            "count": 0,
            "name": "culpa"
        },
        {
            "color": null,
            "count": 2,
            "name": "cum"
        },
        {
            "color": null,
            "count": 0,
            "name": "cumque"
        },
        {
            "color": "#144bba",
            "count": 0,
            "name": "cupiditate"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": null,
            "count": 0,
            "name": "delectus"
        },
        {
            "color": "#6188db",
            "count": 0,
            "name": "deleniti"
        },
        {
            "color": "#e7b695",
            "count": 1,
            "name": "deserunt"
        },
        {
```

```
            "color": "#939b44",
            "count": 0,
            "name": "dicta"
        },
        {
            "color": "#79b3c9",
            "count": 0,
            "name": "dignissimos"
        },
        {
            "color": "#1f8960",
            "count": 0,
            "name": "distinctio"
        },
        {
            "color": "#641bd9",
            "count": 0,
            "name": "dolor"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolore"
        },
        {
            "color": "#604860",
            "count": 0,
            "name": "dolorem"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloremque"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolores"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloribus"
        },
        {
            "color": "#db7ec2",
            "count": 0,
            "name": "dolorum"
        },
        {
            "color": null,
```

```
            "count": 1,
            "name": "ea"
        },
        {
            "color": null,
            "count": 0,
            "name": "eaque"
        },
        {
            "color": "#860b86",
            "count": 0,
            "name": "eius"
        },
        {
            "color": "#5d8273",
            "count": 0,
            "name": "eligendi"
        },
        {
            "color": null,
            "count": 0,
            "name": "enim"
        },
        {
            "color": null,
            "count": 0,
            "name": "error"
        },
        {
            "color": "#d77661",
            "count": 1,
            "name": "esse"
        },
        {
            "color": null,
            "count": 0,
            "name": "et"
        },
        {
            "color": null,
            "count": 0,
            "name": "eum"
        },
        {
            "color": "#5d26b5",
            "count": 1,
            "name": "eveniet"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "ex"
        },
        {
            "color": null,
            "count": 0,
            "name": "excepturi"
        },
        {
            "color": null,
            "count": 0,
            "name": "exercitationem"
        },
        {
            "color": "#740c41",
            "count": 0,
            "name": "expedita"
        },
        {
            "color": null,
            "count": 0,
            "name": "explicabo"
        },
        {
            "color": "#113f4a",
            "count": 0,
            "name": "facere"
        },
        {
            "color": "#0f6b6b",
            "count": 0,
            "name": "facilis"
        },
        {
            "color": null,
            "count": 1,
            "name": "fuga"
        },
        {
            "color": "#1c563a",
            "count": 0,
            "name": "fugiat"
        },
        {
            "color": "#9345df",
            "count": 0,
            "name": "fugit"
        },
        {
            "color": null,
            "count": 0,
            "name": "harum"
```

```
        },
        {
            "color": "#f75f0b",
            "count": 0,
            "name": "hic"
        },
        {
            "color": null,
            "count": 0,
            "name": "id"
        },
        {
            "color": null,
            "count": 0,
            "name": "illo"
        },
        {
            "color": null,
            "count": 0,
            "name": "illum"
        },
        {
            "color": null,
            "count": 0,
            "name": "impedit"
        },
        {
            "color": "#af10ef",
            "count": 0,
            "name": "in"
        },
        {
            "color": "#3099ec",
            "count": 0,
            "name": "incidunt"
        },
        {
            "color": null,
            "count": 1,
            "name": "inventore"
        },
        {
            "color": null,
            "count": 0,
            "name": "ipsam"
        },
        {
            "color": "#da3ba4",
            "count": 0,
            "name": "ipsum"
        },
```

```
        {
            "color": "#491b3a",
            "count": 0,
            "name": "iste"
        },
        {
            "color": null,
            "count": 0,
            "name": "itaque"
        },
        {
            "color": null,
            "count": 0,
            "name": "iure"
        },
        {
            "color": "#3a10e8",
            "count": 0,
            "name": "iusto"
        },
        {
            "color": "#67eac4",
            "count": 0,
            "name": "laborum"
        },
        {
            "color": "#9e3f1f",
            "count": 0,
            "name": "laudantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "libero"
        },
        {
            "color": null,
            "count": 0,
            "name": "magni"
        },
        {
            "color": null,
            "count": 1,
            "name": "maiores"
        },
        {
            "color": "#1acc29",
            "count": 0,
            "name": "maxime"
        },
        {
```

```
            "color": "#f0048e",
            "count": 0,
            "name": "minima"
        },
        {
            "color": "#59b653",
            "count": 0,
            "name": "minus"
        },
        {
            "color": "#494e30",
            "count": 1,
            "name": "modi"
        },
        {
            "color": "#002e7f",
            "count": 1,
            "name": "mollitia"
        },
        {
            "color": "#ce4004",
            "count": 0,
            "name": "nam"
        },
        {
            "color": null,
            "count": 0,
            "name": "natus"
        },
        {
            "color": "#84e3b6",
            "count": 1,
            "name": "necessitatibus"
        },
        {
            "color": "#e81498",
            "count": 0,
            "name": "nemo"
        },
        {
            "color": "#150607",
            "count": 1,
            "name": "neque"
        },
        {
            "color": "#4c8404",
            "count": 1,
            "name": "nesciunt"
        },
        {
            "color": "#98a352",
```

```
            "count": 1,
            "name": "nihil"
        },
        {
            "color": "#ef7fdc",
            "count": 0,
            "name": "nisi"
        },
        {
            "color": "#37031f",
            "count": 1,
            "name": "non"
        },
        {
            "color": null,
            "count": 0,
            "name": "nulla"
        },
        {
            "color": null,
            "count": 1,
            "name": "numquam"
        },
        {
            "color": null,
            "count": 0,
            "name": "obcaecati"
        },
        {
            "color": null,
            "count": 0,
            "name": "odio"
        },
        {
            "color": null,
            "count": 0,
            "name": "odit"
        },
        {
            "color": "#c4f027",
            "count": 0,
            "name": "officia"
        },
        {
            "color": null,
            "count": 0,
            "name": "officiis"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "omnis"
        },
        {
            "color": null,
            "count": 0,
            "name": "optio"
        },
        {
            "color": "#7b0e4e",
            "count": 1,
            "name": "pariatur"
        },
        {
            "color": "#999645",
            "count": 0,
            "name": "perferendis"
        },
        {
            "color": "#afb825",
            "count": 0,
            "name": "perspiciatis"
        },
        {
            "color": null,
            "count": 0,
            "name": "placeat"
        },
        {
            "color": null,
            "count": 0,
            "name": "porro"
        },
        {
            "color": null,
            "count": 0,
            "name": "possimus"
        },
        {
            "color": "#0cd131",
            "count": 0,
            "name": "praesentium"
        },
        {
            "color": null,
            "count": 0,
            "name": "provident"
        },
        {
            "color": "#d91a8b",
            "count": 0,
            "name": "quae"
```

```
        },
        {
            "color": "#0b4425",
            "count": 0,
            "name": "quaerat"
        },
        {
            "color": null,
            "count": 0,
            "name": "quam"
        },
        {
            "color": "#6e3390",
            "count": 0,
            "name": "quas"
        },
        {
            "color": null,
            "count": 0,
            "name": "quasi"
        },
        {
            "color": "#61f611",
            "count": 0,
            "name": "qui"
        },
        {
            "color": "#f53074",
            "count": 0,
            "name": "quia"
        },
        {
            "color": "#c49ac2",
            "count": 0,
            "name": "quibusdam"
        },
        {
            "color": null,
            "count": 0,
            "name": "quis"
        },
        {
            "color": "#ebca0b",
            "count": 0,
            "name": "quisquam"
        },
        {
            "color": null,
            "count": 0,
            "name": "quo"
        },
```

```
        {
            "color": "#0e5b24",
            "count": 0,
            "name": "quod"
        },
        {
            "color": null,
            "count": 0,
            "name": "quos"
        },
        {
            "color": "#570ce3",
            "count": 0,
            "name": "ratione"
        },
        {
            "color": "#560ff6",
            "count": 0,
            "name": "reiciendis"
        },
        {
            "color": "#688119",
            "count": 0,
            "name": "rem"
        },
        {
            "color": "#807389",
            "count": 0,
            "name": "repellat"
        },
        {
            "color": null,
            "count": 0,
            "name": "repellendus"
        },
        {
            "color": null,
            "count": 1,
            "name": "reprehenderit"
        },
        {
            "color": "#3a2b71",
            "count": 0,
            "name": "repudiandae"
        },
        {
            "color": null,
            "count": 0,
            "name": "rerum"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "saepe"
        },
        {
            "color": "#c15b7b",
            "count": 0,
            "name": "sed"
        },
        {
            "color": null,
            "count": 0,
            "name": "sequi"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
        {
            "color": "#3b2404",
            "count": 1,
            "name": "sint"
        },
        {
            "color": "#abdcde",
            "count": 0,
            "name": "sit"
        },
        {
            "color": "#1398ab",
            "count": 0,
            "name": "soluta"
        },
        {
            "color": null,
            "count": 0,
            "name": "sunt"
        },
        {
            "color": "#38abf3",
            "count": 1,
            "name": "suscipit"
        },
        {
            "color": null,
            "count": 0,
            "name": "tempora"
        },
        {
            "color": "#ae2670",
```

```
            "count": 0,
            "name": "tempore"
        },
        {
            "color": null,
            "count": 0,
            "name": "temporibus"
        },
        {
            "color": "#560a5d",
            "count": 0,
            "name": "totam"
        },
        {
            "color": "#98ad13",
            "count": 0,
            "name": "ullam"
        },
        {
            "color": "#da2470",
            "count": 1,
            "name": "unde"
        },
        {
            "color": "#91e065",
            "count": 1,
            "name": "vel"
        },
        {
            "color": null,
            "count": 0,
            "name": "velit"
        },
        {
            "color": null,
            "count": 0,
            "name": "veniam"
        },
        {
            "color": null,
            "count": 0,
            "name": "vero"
        },
        {
            "color": "#d9fe5e",
            "count": 0,
            "name": "vitae"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "voluptate"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptates"
        },
        {
            "color": "#02d22f",
            "count": 1,
            "name": "voluptatum"
        }
    ]
}
```


48.28. User story voter detail

inlcude::generated/user-stories-get-voters-output.adoc[]

48.29. User story watcher detail

inlcude::generated/user-stories-get-watchers-output.adoc[]

48.30. User story status detail


```
{
    "color": "#999999",
    "id": 1,
    "is_archived": false,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name",
    "wip_limit": null
}
```


48.31. Point detail

```
{
    "id": 1,
    "name": "Patch name",
    "order": 1,
    "project": 1,
    "value": null
}
```


48.32. User story custom attribute detail


```
{
    "created_date": "2020-07-02T11:56:21.148Z",
    "description": "vel omnis culpa quisquam nulla",
    "extra": null,
    "id": 1,
    "modified_date": "2020-07-03T08:40:39.162Z",
    "name": "Duration 1",
    "order": 1,
    "project": 1,
    "type": "richtext"
}
```


48.33. User story custom attributes values detail


```
{
    "attributes_values": {
        "3": "240 min"
    },
    "user_story": 1,
    "version": 2
}
```


48.34. Task detail


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
```

```
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2020-07-02T11:56:21.529Z",
    "description": "Nam veritatis facere debitis vitae animi eos cum suscipit
reprehenderit.",
    "description_html": "<p>Nam veritatis facere debitis vitae animi eos cum suscipit
reprehenderit.</p>",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": "2020-05-10T05:32:33.173Z",
    "generated_user_stories": null,
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": true,
    "is_watcher": false,
    "milestone": 1,
    "milestone_slug": "sprint-2020-5-8",
    "modified_date": "2020-07-03T08:41:01.723Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 3,
            "subject": "Add tests for bulk operations"
        },
        "previous": null
    },
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Beta project patch",
        "slug": "project-0"
    },
    "ref": 2,
```

```
    "status": 3,
    "status_extra_info": {
        "color": "#ffcc00",
        "is_closed": true,
        "name": "Ready for test"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "atque",
            null
        ],
        [
            "animi",
            null
        ],
        [
            "cum",
            null
        ],
        [
            "eveniet",
            null
        ],
        [
            "cumque",
            null
        ],
        [
            "reiciendis",
            null
        ],
        [
            "architecto",
            null
        ],
        [
            "perspiciatis",
            null
        ]
    ],
    "taskboard_order": 1593690981529,
    "total_comments": 1,
    "total_voters": 7,
    "total_watchers": 2,
    "us_order": 1593690981529,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": [
            {
                "color": "#f57900",
```

```
                "id": 15,
                "project": {
                    "id": 3,
                    "name": "Project Example 2",
                    "slug": "project-2"
                },
                "ref": 121,
                "subject": "Patching subject"
            }
        ],
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": [
        8,
        3
    ]
}
```


48.35. Task detail (GET)


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2020-07-02T11:56:21.529Z",
    "description": "Nam veritatis facere debitis vitae animi eos cum suscipit
reprehenderit.",
    "description_html": "<p>Nam veritatis facere debitis vitae animi eos cum suscipit
reprehenderit.</p>",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": "2020-05-10T05:32:33.173Z",
    "generated_user_stories": null,
```

```
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": true,
    "is_watcher": false,
    "milestone": 1,
    "milestone_slug": "sprint-2020-5-8",
    "modified_date": "2020-07-03T08:41:01.723Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 3,
            "subject": "Add tests for bulk operations"
        },
        "previous": null
    },
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Beta project patch",
        "slug": "project-0"
    },
    "ref": 2,
    "status": 3,
    "status_extra_info": {
        "color": "#ffcc00",
        "is_closed": true,
        "name": "Ready for test"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "atque",
            null
        ],
        [
            "animi",
            null
        ],
```

```
        [
            "cum",
            null
        ],
        [
            "eveniet",
            null
        ],
        [
            "cumque",
            null
        ],
        [
            "reiciendis",
            null
        ],
        [
            "architecto",
            null
        ],
        [
            "perspiciatis",
            null
        ]
    ],
    "taskboard_order": 1593690981529,
    "total_comments": 1,
    "total_voters": 7,
    "total_watchers": 2,
    "us_order": 1593690981529,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": [
            {
                "color": "#f57900",
                "id": 15,
                "project": {
                    "id": 3,
                    "name": "Project Example 2",
                    "slug": "project-2"
                },
                "ref": 121,
                "subject": "Patching subject"
            }
        ],
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": [
```

```
        8,
        3
    ]
}
```

48.36. Task detail (LIST)

```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Virginia Castro",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "created_date": "2020-07-02T11:56:21.529Z",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": "2020-05-10T05:32:33.173Z",
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": true,
    "is_watcher": true,
    "milestone": 1,
    "milestone_slug": "sprint-2020-5-8",
    "modified_date": "2020-07-03T08:41:01.723Z",
    "owner": 14,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Miguel Molina",
        "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
        "id": 14,
        "is_active": true,
        "photo": null,
        "username": "user8"
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
```

```
        "name": "Beta project patch",
        "slug": "project-0"
    },
    "ref": 2,
    "status": 3,
    "status_extra_info": {
        "color": "#ffcc00",
        "is_closed": true,
        "name": "Ready for test"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "atque",
            null
        ],
        [
            "animi",
            null
        ],
        [
            "cum",
            null
        ],
        [
            "eveniet",
            null
        ],
        [
            "cumque",
            null
        ],
        [
            "reiciendis",
            null
        ],
        [
            "architecto",
            null
        ],
        [
            "perspiciatis",
            null
        ]
    ],
    "taskboard_order": 1593690981529,
    "total_comments": 1,
    "total_voters": 7,
    "total_watchers": 3,
    "us_order": 1593690981529,
    "user_story": 1,
```

```
    "user_story_extra_info": {
        "epics": [
            {
                "color": "#f57900",
                "id": 15,
                "project": {
                    "id": 3,
                    "name": "Project Example 2",
                    "slug": "project-2"
                },
                "ref": 121,
                "subject": "Patching subject"
            }
        ],
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": [
        8,
        3,
        6
    ]
}
```


48.37. Task filters data detail


```
{
    "assigned_to": [
        {
            "count": 5,
            "full_name": "",
            "id": null
        },
        {
            "count": 2,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 2,
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Catalina Fernandez",
            "id": 9
```

```
        },
        {
            "count": 1,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 1,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 3,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 4,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 2,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 5,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 3,
            "full_name": "Virginia Castro",
            "id": 15
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "owners": [
        {
            "count": 3,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 3,
            "full_name": "Bego\u00f1a Flores",
```

```
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 3,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 3,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 4,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 3,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 1,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 6,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 2,
            "full_name": "Virginia Castro",
            "id": 15
        }
    ],
    "roles": [
        {
            "color": null,
            "count": 2,
            "id": 1,
            "name": "Patch name",
            "order": 10
        },
```

```
        {
            "color": null,
            "count": 0,
            "id": 44,
            "name": "New role name",
            "order": 10
        },
        {
            "color": null,
            "count": 0,
            "id": 43,
            "name": "New role",
            "order": 10
        },
        {
            "color": null,
            "count": 5,
            "id": 2,
            "name": "Design",
            "order": 20
        },
        {
            "color": null,
            "count": 1,
            "id": 3,
            "name": "Front",
            "order": 30
        },
        {
            "color": null,
            "count": 3,
            "id": 4,
            "name": "Back",
            "order": 40
        },
        {
            "color": null,
            "count": 8,
            "id": 5,
            "name": "Product Owner",
            "order": 50
        },
        {
            "color": null,
            "count": 5,
            "id": 6,
            "name": "Stakeholder",
            "order": 60
        }
    ],
    "statuses": [
```

```
        {
            "color": "#ffcc00",
            "count": 7,
            "id": 3,
            "name": "Ready for test",
            "order": 3
        },
        {
            "color": "#669900",
            "count": 4,
            "id": 4,
            "name": "Closed",
            "order": 4
        },
        {
            "color": "#ff9900",
            "count": 4,
            "id": 2,
            "name": "In progress",
            "order": 5
        },
        {
            "color": "#999999",
            "count": 6,
            "id": 5,
            "name": "Needs Info",
            "order": 5
        },
        {
            "color": "#AAAAAA",
            "count": 0,
            "id": 41,
            "name": "New status",
            "order": 8
        },
        {
            "color": "#999999",
            "count": 4,
            "id": 1,
            "name": "Patch status name",
            "order": 10
        },
        {
            "color": "#999999",
            "count": 0,
            "id": 42,
            "name": "New status name",
            "order": 10
        }
    ],
    "tags": [
```

```
        {
            "color": null,
            "count": 2,
            "name": "animi"
        },
        {
            "color": null,
            "count": 1,
            "name": "architecto"
        },
        {
            "color": null,
            "count": 1,
            "name": "atque"
        },
        {
            "color": null,
            "count": 3,
            "name": "cum"
        },
        {
            "color": null,
            "count": 1,
            "name": "cumque"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": null,
            "count": 1,
            "name": "eveniet"
        },
        {
            "color": null,
            "count": 1,
            "name": "perspiciatis"
        },
        {
            "color": null,
            "count": 3,
            "name": "reiciendis"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        }
    ]
```

```
}
```

48.38. Task voter detail

```
{
    "full_name": "Administrator",
    "id": 5,
    "username": "admin"
}
```

48.39. Task watcher detail

```
{
    "full_name": "GitLab",
    "id": 3,
    "username": "gitlab-da51373c16de4169856606e6e6dd9b21"
}
```

48.40. Task status detail

```
{
    "color": "#999999",
    "id": 1,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name"
}
```

48.41. Task custom attribute detail

```
{
    "created_date": "2020-07-02T11:56:21.156Z",
    "description": "a sequi saepe quibusdam culpa optio accusantium minima obcaecati",
    "extra": null,
    "id": 1,
    "modified_date": "2020-07-03T08:41:04.666Z",
    "name": "Duration 1",
    "order": 1,
    "project": 1,
    "type": "dropdown"
}
```


48.42. Task custom attributes values detail


```
{
    "attributes_values": {
        "1": "240 min"
    },
    "task": 1,
    "version": 2
}
```


48.43. Issue detail


```
{
    "assigned_to": null,
    "assigned_to_extra_info": null,
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2020-07-02T11:56:33.680Z",
    "description": "Non laborum officia quae veritatis vel ab voluptas esse beatae
doloribus ipsa, recusandae possimus debitis tempore iusto aspernatur ad quo,
repudiandae error praesentium, perspiciatis modi aliquam quisquam quos debitis
reiciendis excepturi? Ex aliquam laboriosam sequi minus dolore nisi dolorem quidem
aliquid, inventore doloribus non illum nulla minus recusandae tempore error aut
praesentium, perferendis recusandae possimus accusamus vitae illo, neque quod cumque
temporibus modi rerum eum praesentium ea ex? Laudantium illum possimus veniam impedit
qui amet aliquam, quos vero est eligendi asperiores quasi reprehenderit tempora
doloremque? Amet laudantium iusto reprehenderit in eveniet voluptatem expedita
cupiditate odio explicabo quasi, eveniet quis et est cumque eum nobis nihil aut quasi,
soluta consequuntur minima quam quod exercitationem iste illum culpa nemo, aliquid
magni nulla accusamus esse libero at.",
    "description_html": "<p>Non laborum officia quae veritatis vel ab voluptas esse
beatae doloribus ipsa, recusandae possimus debitis tempore iusto aspernatur ad quo,
```

repudiandae error praesentium, perspiciatis modi aliquam quisquam quos debitis reiciendis excepturi? Ex aliquam laboriosam sequi minus dolore nisi dolorem quidem aliquid, inventore doloribus non illum nulla minus recusandae tempore error aut praesentium, perferendis recusandae possimus accusamus vitae illo, neque quod cumque temporibus modi rerum eum praesentium ea ex? Laudantium illum possimus veniam impedit qui amet aliquam, quos vero est eligendi asperiores quasi reprehenderit tempora doloremque? Amet laudantium iusto reprehenderit in eveniet voluptatem expedita cupiditate odio explicabo quasi, eveniet quis et est cumque eum nobis nihil aut quasi, soluta consequuntur minima quam quod exercitationem iste illum culpa nemo, aliquid magni nulla accusamus esse libero at.</p>",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": "2020-07-02T11:56:33.756Z",
    "generated_user_stories": null,
    "id": 3,
    "is_blocked": false,
    "is_closed": true,
    "is_voter": true,
    "is_watcher": false,
    "milestone": null,
    "modified_date": "2020-07-03T08:40:51.052Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 43,
            "subject": "Added file copying and processing of images (resizing)"
        },
        "previous": {
            "id": 4,
            "ref": 45,
            "subject": "Add setting to allow regular users to create folders at the root level."
        }
    },
    "owner": 7,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Bego\u00f1a Flores",
        "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
        "id": 7,
        "is_active": true,
        "photo": null,
        "username": "user1"
    },
    "priority": 2,
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,

```
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 44,
    "severity": 1,
    "status": 4,
    "status_extra_info": {
        "color": "#BFB35A",
        "is_closed": true,
        "name": "Closed"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "dicta",
            "#939b44"
        ]
    ],
    "total_voters": 4,
    "total_watchers": 0,
    "type": 2,
    "version": 2,
    "watchers": []
}
```


48.44. Issue detail (GET)


```
{
    "assigned_to": null,
    "assigned_to_extra_info": null,
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2020-07-02T11:56:33.680Z",
    "description": "Non laborum officia quae veritatis vel ab voluptas esse beatae
doloribus ipsa, recusandae possimus debitis tempore iusto aspernatur ad quo,
repudiandae error praesentium, perspiciatis modi aliquam quisquam quos debitis
reiciendis excepturi? Ex aliquam laboriosam sequi minus dolore nisi dolorem quidem
aliquid, inventore doloribus non illum nulla minus recusandae tempore error aut
praesentium, perferendis recusandae possimus accusamus vitae illo, neque quod cumque
temporibus modi rerum eum praesentium ea ex? Laudantium illum possimus veniam impedit
qui amet aliquam, quos vero est eligendi asperiores quasi reprehenderit tempora
doloremque? Amet laudantium iusto reprehenderit in eveniet voluptatem expedita
cupiditate odio explicabo quasi, eveniet quis et est cumque eum nobis nihil aut quasi,
soluta consequuntur minima quam quod exercitationem iste illum culpa nemo, aliquid
magni nulla accusamus esse libero at.",
    "description_html": "<p>Non laborum officia quae veritatis vel ab voluptas esse
beatae doloribus ipsa, recusandae possimus debitis tempore iusto aspernatur ad quo,
```

```
repudiandae error praesentium, perspiciatis modi aliquam quisquam quos debitis
reiciendis excepturi? Ex aliquam laboriosam sequi minus dolore nisi dolorem quidem
aliquid, inventore doloribus non illum nulla minus recusandae tempore error aut
praesentium, perferendis recusandae possimus accusamus vitae illo, neque quod cumque
temporibus modi rerum eum praesentium ea ex? Laudantium illum possimus veniam impedit
qui amet aliquam, quos vero est eligendi asperiores quasi reprehenderit tempora
doloremque? Amet laudantium iusto reprehenderit in eveniet voluptatem expedita
cupiditate odio explicabo quasi, eveniet quis et est cumque eum nobis nihil aut quasi,
soluta consequuntur minima quam quod exercitationem iste illum culpa nemo, aliquid
magni nulla accusamus esse libero at.</p>",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": "2020-07-02T11:56:33.756Z",
    "generated_user_stories": null,
    "id": 3,
    "is_blocked": false,
    "is_closed": true,
    "is_voter": true,
    "is_watcher": false,
    "milestone": null,
    "modified_date": "2020-07-03T08:40:51.052Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 43,
            "subject": "Added file copying and processing of images (resizing)"
        },
        "previous": {
            "id": 4,
            "ref": 45,
            "subject": "Add setting to allow regular users to create folders at the
root level."
        }
    },
    "owner": 7,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Bego\u00f1a Flores",
        "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
        "id": 7,
        "is_active": true,
        "photo": null,
        "username": "user1"
    },
    "priority": 2,
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
```

```
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 44,
    "severity": 1,
    "status": 4,
    "status_extra_info": {
        "color": "#BFB35A",
        "is_closed": true,
        "name": "Closed"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "dicta",
            "#939b44"
        ]
    ],
    "total_voters": 4,
    "total_watchers": 0,
    "type": 2,
    "version": 2,
    "watchers": []
}
```


48.45. Issue detail (LIST)


```
{
    "assigned_to": null,
    "assigned_to_extra_info": null,
    "attachments": [],
    "blocked_note": "",
    "created_date": "2016-08-04T11:08:19Z",
    "due_date": null,
    "due_date_reason": "",
    "due_date_status": "not_set",
    "external_reference": null,
    "finished_date": null,
    "id": 114,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "milestone": null,
    "modified_date": "2016-08-04T11:08:19Z",
    "owner": 13,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Mohamed Ortega",
```

```
        "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
        "id": 13,
        "is_active": true,
        "photo": null,
        "username": "user7"
    },
    "priority": 23,
    "project": 8,
    "project_extra_info": {
        "id": 8,
        "logo_small_url": null,
        "name": "Project Example",
        "slug": "project-example"
    },
    "ref": 59,
    "severity": 38,
    "status": 58,
    "status_extra_info": {
        "color": "#666666",
        "is_closed": false,
        "name": "Postponed"
    },
    "subject": "Exception is thrown if trying to add a folder with existing name",
    "tags": [
        [
            "excepturi",
            null
        ],
        [
            "quidem",
            "#ae6519"
        ],
        [
            "adipisci",
            null
        ],
        [
            "nisi",
            null
        ],
        [
            "voluptatum",
            "#02d22f"
        ],
        [
            "alias",
            "#cdb6fd"
        ],
        [
            "vero",
            "#74e191"
```

```
        ],
        [
            "asperiores",
            null
        ],
        [
            "quibusdam",
            null
        ],
        [
            "omnis",
            "#fc9548"
        ]
    ],
    "total_voters": 0,
    "total_watchers": 1,
    "type": 23,
    "version": 1,
    "watchers": [
        13
    ]
}
```


48.46. Issue filters data detail


```
{
    "assigned_to": [
        {
            "count": 8,
            "full_name": "",
            "id": null
        },
        {
            "count": 5,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 1,
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
            "count": 2,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 1,
```

```
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 2,
            "full_name": "Francisco Gil",
            "id": 8
        },
        {
            "count": 1,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 1,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 3,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 3,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 0,
            "full_name": "Virginia Castro",
            "id": 15
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "owners": [
        {
            "count": 2,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 4,
            "full_name": "Bego\u00f1a Flores",
            "id": 7
        },
        {
```

```
            "count": 3,
            "full_name": "Catalina Fernandez",
            "id": 9
        },
        {
            "count": 1,
            "full_name": "Enrique Crespo",
            "id": 10
        },
        {
            "count": 2,
            "full_name": "Miguel Molina",
            "id": 14
        },
        {
            "count": 4,
            "full_name": "Mohamed Ortega",
            "id": 13
        },
        {
            "count": 2,
            "full_name": "Vanesa Garcia",
            "id": 12
        },
        {
            "count": 6,
            "full_name": "Vanesa Torres",
            "id": 6
        },
        {
            "count": 3,
            "full_name": "Virginia Castro",
            "id": 15
        }
    ],
    "priorities": [
        {
            "color": "#666666",
            "count": 9,
            "id": 1,
            "name": "Low",
            "order": 1
        },
        {
            "color": "#669933",
            "count": 13,
            "id": 2,
            "name": "Normal",
            "order": 3
        },
        {
```

```
            "color": "#CC0000",
            "count": 5,
            "id": 3,
            "name": "High",
            "order": 5
        }
    ],
    "roles": [
        {
            "color": null,
            "count": 1,
            "id": 1,
            "name": "Patch name",
            "order": 10
        },
        {
            "color": null,
            "count": 0,
            "id": 43,
            "name": "New role",
            "order": 10
        },
        {
            "color": null,
            "count": 0,
            "id": 44,
            "name": "New role name",
            "order": 10
        },
        {
            "color": null,
            "count": 3,
            "id": 2,
            "name": "Design",
            "order": 20
        },
        {
            "color": null,
            "count": 2,
            "id": 3,
            "name": "Front",
            "order": 30
        },
        {
            "color": null,
            "count": 7,
            "id": 4,
            "name": "Back",
            "order": 40
        },
        {
```

```
            "color": null,
            "count": 3,
            "id": 5,
            "name": "Product Owner",
            "order": 50
        },
        {
            "color": null,
            "count": 3,
            "id": 6,
            "name": "Stakeholder",
            "order": 60
        }
    ],
    "severities": [
        {
            "color": "#0000FF",
            "count": 9,
            "id": 3,
            "name": "Normal",
            "order": 3
        },
        {
            "color": "#FFA500",
            "count": 5,
            "id": 4,
            "name": "Important",
            "order": 4
        },
        {
            "color": "#CC0000",
            "count": 4,
            "id": 5,
            "name": "Critical",
            "order": 5
        },
        {
            "color": "#669933",
            "count": 4,
            "id": 2,
            "name": "Minor",
            "order": 5
        },
        {
            "color": "#AAAAAA",
            "count": 0,
            "id": 41,
            "name": "New severity",
            "order": 8
        },
        {
```

```
            "color": "#999999",
            "count": 0,
            "id": 42,
            "name": "New severity name",
            "order": 10
        },
        {
            "color": "#666666",
            "count": 5,
            "id": 1,
            "name": "Patch name",
            "order": 10
        }
    ],
    "statuses": [
        {
            "color": "#88A65E",
            "count": 4,
            "id": 3,
            "name": "Ready for test",
            "order": 3
        },
        {
            "color": "#BFB35A",
            "count": 3,
            "id": 4,
            "name": "Closed",
            "order": 4
        },
        {
            "color": "#89BAB4",
            "count": 3,
            "id": 5,
            "name": "Needs Info",
            "order": 5
        },
        {
            "color": "#5E8C6A",
            "count": 2,
            "id": 2,
            "name": "In progress",
            "order": 5
        },
        {
            "color": "#CC0000",
            "count": 3,
            "id": 6,
            "name": "Rejected",
            "order": 6
        },
        {
```

```
            "color": "#666666",
            "count": 5,
            "id": 7,
            "name": "Postponed",
            "order": 7
        },
        {
            "color": "#AAAAAA",
            "count": 0,
            "id": 50,
            "name": "New status",
            "order": 8
        },
        {
            "color": "#999999",
            "count": 0,
            "id": 51,
            "name": "New status name",
            "order": 10
        },
        {
            "color": "#8C2318",
            "count": 7,
            "id": 1,
            "name": "Patch status name",
            "order": 10
        }
    ],
    "tags": [
        {
            "color": "#da2361",
            "count": 2,
            "name": "ab"
        },
        {
            "color": "#801cf7",
            "count": 1,
            "name": "accusamus"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusantium"
        },
        {
            "color": null,
            "count": 2,
            "name": "ad"
        },
        {
            "color": "#cdb6fd",
```

```
            "count": 0,
            "name": "alias"
        },
        {
            "color": null,
            "count": 1,
            "name": "aliquam"
        },
        {
            "color": "#f01df5",
            "count": 1,
            "name": "aliquid"
        },
        {
            "color": "#db04fb",
            "count": 1,
            "name": "amet"
        },
        {
            "color": null,
            "count": 0,
            "name": "animi"
        },
        {
            "color": null,
            "count": 0,
            "name": "aperiam"
        },
        {
            "color": null,
            "count": 1,
            "name": "architecto"
        },
        {
            "color": null,
            "count": 3,
            "name": "asperiores"
        },
        {
            "color": null,
            "count": 0,
            "name": "assumenda"
        },
        {
            "color": null,
            "count": 1,
            "name": "at"
        },
        {
            "color": "#713547",
            "count": 2,
```

```
            "name": "atque"
        },
        {
            "color": null,
            "count": 0,
            "name": "autem"
        },
        {
            "color": "#65026b",
            "count": 3,
            "name": "blanditiis"
        },
        {
            "color": "#3b70df",
            "count": 0,
            "name": "commodi"
        },
        {
            "color": "#97176f",
            "count": 0,
            "name": "consectetur"
        },
        {
            "color": "#3ad7db",
            "count": 1,
            "name": "consequatur"
        },
        {
            "color": "#ce24ec",
            "count": 0,
            "name": "consequuntur"
        },
        {
            "color": "#ed9c91",
            "count": 0,
            "name": "corporis"
        },
        {
            "color": "#432493",
            "count": 1,
            "name": "corrupti"
        },
        {
            "color": null,
            "count": 0,
            "name": "culpa"
        },
        {
            "color": null,
            "count": 0,
            "name": "cum"
```

```
        },
        {
            "color": null,
            "count": 1,
            "name": "cumque"
        },
        {
            "color": "#144bba",
            "count": 2,
            "name": "cupiditate"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": null,
            "count": 3,
            "name": "delectus"
        },
        {
            "color": "#6188db",
            "count": 0,
            "name": "deleniti"
        },
        {
            "color": "#e7b695",
            "count": 0,
            "name": "deserunt"
        },
        {
            "color": "#939b44",
            "count": 1,
            "name": "dicta"
        },
        {
            "color": "#79b3c9",
            "count": 0,
            "name": "dignissimos"
        },
        {
            "color": "#1f8960",
            "count": 1,
            "name": "distinctio"
        },
        {
            "color": "#641bd9",
            "count": 1,
            "name": "dolor"
        },
```

```
        {
            "color": null,
            "count": 2,
            "name": "dolore"
        },
        {
            "color": "#604860",
            "count": 1,
            "name": "dolorem"
        },
        {
            "color": null,
            "count": 1,
            "name": "doloremque"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolores"
        },
        {
            "color": null,
            "count": 2,
            "name": "doloribus"
        },
        {
            "color": "#db7ec2",
            "count": 1,
            "name": "dolorum"
        },
        {
            "color": null,
            "count": 1,
            "name": "ea"
        },
        {
            "color": null,
            "count": 1,
            "name": "eaque"
        },
        {
            "color": "#860b86",
            "count": 0,
            "name": "eius"
        },
        {
            "color": "#5d8273",
            "count": 1,
            "name": "eligendi"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "enim"
        },
        {
            "color": null,
            "count": 0,
            "name": "error"
        },
        {
            "color": "#d77661",
            "count": 1,
            "name": "esse"
        },
        {
            "color": null,
            "count": 1,
            "name": "et"
        },
        {
            "color": null,
            "count": 0,
            "name": "eum"
        },
        {
            "color": "#5d26b5",
            "count": 2,
            "name": "eveniet"
        },
        {
            "color": null,
            "count": 0,
            "name": "ex"
        },
        {
            "color": null,
            "count": 0,
            "name": "excepturi"
        },
        {
            "color": null,
            "count": 0,
            "name": "exercitationem"
        },
        {
            "color": "#740c41",
            "count": 1,
            "name": "expedita"
        },
        {
            "color": null,
```

```
            "count": 1,
            "name": "explicabo"
        },
        {
            "color": "#113f4a",
            "count": 1,
            "name": "facere"
        },
        {
            "color": "#0f6b6b",
            "count": 0,
            "name": "facilis"
        },
        {
            "color": null,
            "count": 0,
            "name": "fuga"
        },
        {
            "color": "#1c563a",
            "count": 1,
            "name": "fugiat"
        },
        {
            "color": "#9345df",
            "count": 1,
            "name": "fugit"
        },
        {
            "color": null,
            "count": 0,
            "name": "harum"
        },
        {
            "color": "#f75f0b",
            "count": 1,
            "name": "hic"
        },
        {
            "color": null,
            "count": 0,
            "name": "id"
        },
        {
            "color": null,
            "count": 0,
            "name": "illo"
        },
        {
            "color": null,
            "count": 1,
```

```
            "name": "illum"
        },
        {
            "color": null,
            "count": 2,
            "name": "impedit"
        },
        {
            "color": "#af10ef",
            "count": 1,
            "name": "in"
        },
        {
            "color": "#3099ec",
            "count": 1,
            "name": "incidunt"
        },
        {
            "color": null,
            "count": 0,
            "name": "inventore"
        },
        {
            "color": null,
            "count": 0,
            "name": "ipsam"
        },
        {
            "color": "#da3ba4",
            "count": 1,
            "name": "ipsum"
        },
        {
            "color": "#491b3a",
            "count": 0,
            "name": "iste"
        },
        {
            "color": null,
            "count": 1,
            "name": "itaque"
        },
        {
            "color": null,
            "count": 1,
            "name": "iure"
        },
        {
            "color": "#3a10e8",
            "count": 0,
            "name": "iusto"
```

```
        },
        {
            "color": "#67eac4",
            "count": 0,
            "name": "laborum"
        },
        {
            "color": "#9e3f1f",
            "count": 0,
            "name": "laudantium"
        },
        {
            "color": null,
            "count": 1,
            "name": "libero"
        },
        {
            "color": null,
            "count": 2,
            "name": "magni"
        },
        {
            "color": null,
            "count": 2,
            "name": "maiores"
        },
        {
            "color": "#1acc29",
            "count": 0,
            "name": "maxime"
        },
        {
            "color": "#f0048e",
            "count": 1,
            "name": "minima"
        },
        {
            "color": "#59b653",
            "count": 1,
            "name": "minus"
        },
        {
            "color": "#494e30",
            "count": 0,
            "name": "modi"
        },
        {
            "color": "#002e7f",
            "count": 0,
            "name": "mollitia"
        },
```

```
        {
            "color": "#ce4004",
            "count": 1,
            "name": "nam"
        },
        {
            "color": null,
            "count": 1,
            "name": "natus"
        },
        {
            "color": "#84e3b6",
            "count": 2,
            "name": "necessitatibus"
        },
        {
            "color": "#e81498",
            "count": 1,
            "name": "nemo"
        },
        {
            "color": "#150607",
            "count": 0,
            "name": "neque"
        },
        {
            "color": "#4c8404",
            "count": 2,
            "name": "nesciunt"
        },
        {
            "color": "#98a352",
            "count": 2,
            "name": "nihil"
        },
        {
            "color": "#ef7fdc",
            "count": 0,
            "name": "nisi"
        },
        {
            "color": "#37031f",
            "count": 0,
            "name": "non"
        },
        {
            "color": null,
            "count": 1,
            "name": "nulla"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "numquam"
        },
        {
            "color": null,
            "count": 1,
            "name": "obcaecati"
        },
        {
            "color": null,
            "count": 1,
            "name": "odio"
        },
        {
            "color": null,
            "count": 1,
            "name": "odit"
        },
        {
            "color": "#c4f027",
            "count": 1,
            "name": "officia"
        },
        {
            "color": null,
            "count": 0,
            "name": "officiis"
        },
        {
            "color": null,
            "count": 1,
            "name": "omnis"
        },
        {
            "color": null,
            "count": 1,
            "name": "optio"
        },
        {
            "color": "#7b0e4e",
            "count": 0,
            "name": "pariatur"
        },
        {
            "color": "#999645",
            "count": 0,
            "name": "perferendis"
        },
        {
            "color": "#afb825",
```

```
            "count": 3,
            "name": "perspiciatis"
        },
        {
            "color": null,
            "count": 3,
            "name": "placeat"
        },
        {
            "color": null,
            "count": 2,
            "name": "porro"
        },
        {
            "color": null,
            "count": 1,
            "name": "possimus"
        },
        {
            "color": "#0cd131",
            "count": 1,
            "name": "praesentium"
        },
        {
            "color": null,
            "count": 0,
            "name": "provident"
        },
        {
            "color": "#d91a8b",
            "count": 0,
            "name": "quae"
        },
        {
            "color": "#0b4425",
            "count": 0,
            "name": "quaerat"
        },
        {
            "color": null,
            "count": 1,
            "name": "quam"
        },
        {
            "color": "#6e3390",
            "count": 1,
            "name": "quas"
        },
        {
            "color": null,
            "count": 1,
```

```
            "name": "quasi"
        },
        {
            "color": "#61f611",
            "count": 0,
            "name": "qui"
        },
        {
            "color": "#f53074",
            "count": 0,
            "name": "quia"
        },
        {
            "color": "#c49ac2",
            "count": 2,
            "name": "quibusdam"
        },
        {
            "color": null,
            "count": 0,
            "name": "quis"
        },
        {
            "color": "#ebca0b",
            "count": 1,
            "name": "quisquam"
        },
        {
            "color": null,
            "count": 2,
            "name": "quo"
        },
        {
            "color": "#0e5b24",
            "count": 1,
            "name": "quod"
        },
        {
            "color": null,
            "count": 1,
            "name": "quos"
        },
        {
            "color": "#570ce3",
            "count": 1,
            "name": "ratione"
        },
        {
            "color": "#560ff6",
            "count": 0,
            "name": "reiciendis"
```

```
        },
        {
            "color": "#688119",
            "count": 1,
            "name": "rem"
        },
        {
            "color": "#807389",
            "count": 0,
            "name": "repellat"
        },
        {
            "color": null,
            "count": 1,
            "name": "repellendus"
        },
        {
            "color": null,
            "count": 1,
            "name": "reprehenderit"
        },
        {
            "color": "#3a2b71",
            "count": 1,
            "name": "repudiandae"
        },
        {
            "color": null,
            "count": 0,
            "name": "rerum"
        },
        {
            "color": null,
            "count": 2,
            "name": "saepe"
        },
        {
            "color": "#c15b7b",
            "count": 1,
            "name": "sed"
        },
        {
            "color": null,
            "count": 2,
            "name": "sequi"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
```

```
        {
            "color": "#3b2404",
            "count": 1,
            "name": "sint"
        },
        {
            "color": "#abdcde",
            "count": 1,
            "name": "sit"
        },
        {
            "color": "#1398ab",
            "count": 1,
            "name": "soluta"
        },
        {
            "color": null,
            "count": 3,
            "name": "sunt"
        },
        {
            "color": "#38abf3",
            "count": 0,
            "name": "suscipit"
        },
        {
            "color": null,
            "count": 0,
            "name": "tempora"
        },
        {
            "color": "#ae2670",
            "count": 0,
            "name": "tempore"
        },
        {
            "color": null,
            "count": 1,
            "name": "temporibus"
        },
        {
            "color": "#560a5d",
            "count": 2,
            "name": "totam"
        },
        {
            "color": "#98ad13",
            "count": 0,
            "name": "ullam"
        },
        {
```

```
            "color": "#da2470",
            "count": 1,
            "name": "unde"
        },
        {
            "color": "#91e065",
            "count": 0,
            "name": "vel"
        },
        {
            "color": null,
            "count": 1,
            "name": "velit"
        },
        {
            "color": null,
            "count": 0,
            "name": "veniam"
        },
        {
            "color": null,
            "count": 1,
            "name": "vero"
        },
        {
            "color": "#d9fe5e",
            "count": 0,
            "name": "vitae"
        },
        {
            "color": null,
            "count": 1,
            "name": "voluptate"
        },
        {
            "color": null,
            "count": 1,
            "name": "voluptates"
        },
        {
            "color": "#02d22f",
            "count": 1,
            "name": "voluptatum"
        }
    ],
    "types": [
        {
            "color": "#89BAB4",
            "count": 12,
            "id": 1,
            "name": "Bug",
```

```
            "order": 1
        },
        {
            "color": "#ba89a8",
            "count": 7,
            "id": 2,
            "name": "Question",
            "order": 2
        },
        {
            "color": "#89a8ba",
            "count": 8,
            "id": 3,
            "name": "Enhancement",
            "order": 3
        }
    ]
}
```


48.47. Issue voters detail


```
{
    "full_name": "GitHub",
    "id": 2,
    "username": "github-ecf32e2347fd4afda1b4914414a965cf"
}
```


48.48. Issue watchers detail


```
{
    "full_name": "Vanesa Torres",
    "id": 6,
    "username": "user2114747470430251528"
}
```


48.49. Issue status detail

```
{
    "color": "#8C2318",
    "id": 1,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name"
}
```

48.50. Issue type detail

```
{
    "color": "#89BAB4",
    "id": 1,
    "name": "Patch type name",
    "order": 1,
    "project": 1
}
```

48.51. Priority detail

```
{
    "color": "#666666",
    "id": 1,
    "name": "Patch name",
    "order": 1,
    "project": 1
}
```

48.52. Severity detail

```
{
    "color": "#666666",
    "id": 1,
    "name": "Patch name",
    "order": 1,
    "project": 1
}
```

48.53. Issue custom attribute detail

```
{
    "created_date": "2020-07-02T11:56:21.171Z",
    "description": "voluptate rem perspiciatis ipsum",
    "extra": null,
    "id": 5,
    "modified_date": "2020-07-03T08:40:59.952Z",
    "name": "Duration 1",
    "order": 1,
    "project": 1,
    "type": "checkbox"
}
```


48.54. Issue custom attributes values detail


```
{
    "attributes_values": {
        "5": "240 min"
    },
    "issue": 22,
    "version": 2
}
```


48.55. Wiki page


```
{
    "content": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.\n\nEsse pariatur commodi similique
tenetur nostrum quae eos sed dolorum natus, incidunt in expedita assumenda nulla
libero, explicabo rem quia possimus repudiandae aut harum consequatur nesciunt
provident? Ea officiis rem laborum eos temporibus veniam, nihil pariatur officiis
voluptate, earum aliquid itaque modi officiis ullam a non nesciunt, explicabo
voluptates reiciendis hic mollitia cupiditate iste beatae earum tenetur, minus
doloribus amet esse?\n\nTempore voluptas consectetur, doloribus at corporis dolorem
excepturi perferendis hic eaque? Sequi quibusdam tempore, iusto tempora corrupti
assumenda est unde fugiat quibusdam autem aliquam neque architecto, ea quam sequi
ratione similique officia veritatis, enim tempore perspiciatis corporis. Quis fugiat
assumenda minima at optio explicabo pariatur numquam dolorum, aspernatur corrupti
rerum.\n\nEveniet id dolor nobis error doloribus atque, doloremque enim impedit atque
aperiam a placeat veniam ipsam debitis in fuga, perspiciatis cumque commodi dolorum
optio nulla a architecto magnam quas. Eos ducimus deserunt beatae inventore sequi
minus est quas temporibus, in facere accusantium nobis ullam dolorum autem harum
doloribus at ad, ea praesentium vitae temporibus esse rem delectus veniam tempora at,
```

asperiores ullam voluptatibus sunt saepe aut deleniti repudiandae rem animi dolorem unde? Eos ipsum quae in possimus impedit quaerat illo sequi eum, optio cumque sunt in iusto ex quas, consectetur quibusdam laborum molestiae perferendis animi placeat, nemo libero rem ratione sint blanditiis commodi aliquid, minus excepturi itaque cupiditate quisquam?\n\nMagnam nobis eaque odio repellat, praesentium modi eligendi, officiis nisi neque porro vitae optio numquam exercitationem delectus eos, minima quasi magni molestias quae vel officia necessitatibus, quas voluptate minima. Illum cum impedit vitae consequatur dicta, vel perspiciatis officiis quo, quisquam rerum voluptatum dolore eos consequatur ipsam nemo, quos omnis et veniam at, alias dolor necessitatibus quibusdam ex modi laboriosam. Sit laborum ex, earum facere nam fuga, in vel recusandae explicabo dicta sunt commodi animi quod sed, quam quaerat ipsa animi rerum quia facere cumque, non vero odit quas officia suscipit in culpa facere veniam aliquam? Iste numquam cupiditate deserunt consequatur, odit possimus cumque incidunt delectus, possimus sunt a labore quia rerum quo eaque repudiandae laborum, autem fuga ipsam nam neque, distinctio quis facilis ratione.\n\nConsectetur enim et eligendi quod illo itaque sit repudiandae veniam rerum vitae, rem repudiandae atque aut nobis officia officiis quod totam voluptatem in laboriosam, atque laudantium veniam magnam perferendis tenetur, aliquam ad reprehenderit rem inventore vero labore assumenda, ab nihil consectetur accusamus a. Animi odit quia quod aliquid quam magni facilis, esse laboriosam animi rem aspernatur quo molestias vitae.\n\nEum nesciunt harum corporis porro voluptates architecto error voluptate, velit laudantium repellat consequatur, vel tempore eos officiis id dolores earum ex at suscipit, eius tempore nobis, consectetur sequi incidunt nemo natus sed sapiente. Itaque tempora reprehenderit, non architecto libero distinctio qui sed voluptatem quasi iusto sunt corporis optio, officia ex doloremque in ut blanditiis magni possimus illum, amet fuga quasi nostrum, ut eos ex est eum architecto optio itaque alias. Pariatur quod quos laudantium impedit, officiis iusto quod corrupti vero dolorem.\n\nPerferendis dicta exercitationem amet ullam eaque in possimus eligendi provident ratione officia, earum velit ad deserunt eveniet, tempora id ut odio necessitatibus a, dolorum iste labore rerum in dicta cum voluptates, aperiam quos rerum quibusdam ducimus.\n\nQuam quo laudantium tenetur, porro vero distinctio asperiores ab minus sint dolorum consequuntur deserunt iure, corporis cumque perferendis illum velit veritatis odit asperiores natus enim at exercitationem, dolorem cum cupiditate iste, quas explicabo deleniti voluptate neque? Vero explicabo recusandae esse voluptate obcaecati velit fugiat quae cupiditate necessitatibus, pariatur alias ullam, numquam dolores maxime corrupti debitis ea praesentium facere? Ratione earum qui beatae dolor voluptatibus alias dicta repellat vero quis, ipsum ipsam impedit neque possimus quasi, temporibus iure accusantium officia facilis quas necessitatibus ut asperiores, exercitationem mollitia corporis ipsum sint tempore architecto quo ipsam laboriosam dolor officiis, praesentium ullam voluptatum? Repellendus vitae distinctio cumque architecto eius nesciunt voluptate, quisquam commodi eius at dolorum.\n\nPraesentium saepe facilis iste cum, mollitia molestiae cum aperiam fuga vitae sequi repellat nisi maxime perferendis quaerat, facilis corrupti fugit necessitatibus deleniti esse error quaerat temporibus voluptatum magnam recusandae, molestias ducimus temporibus aut culpa illo facere sed hic sequi voluptatem, autem adipisci atque nemo quam sapiente voluptas ipsam molestias possimus in? Beatae eveniet commodi similique incidunt nihil, doloremque necessitatibus ipsum similique laboriosam aliquam delectus reprehenderit earum? Quisquam nostrum consequuntur provident quis et possimus iusto, ad ea nisi eligendi debitis, alias at reiciendis accusantium officia veniam similique, officia odit rerum iure?",
    "created_date": "2020-07-02T11:56:43.109Z",

```
    "editions": 3,
    "html": "<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.</p>\n<p>Esse pariatur commodi similique
tenetur nostrum quae eos sed dolorum natus, incidunt in expedita assumenda nulla
libero, explicabo rem quia possimus repudiandae aut harum consequatur nesciunt
provident? Ea officiis rem laborum eos temporibus veniam, nihil pariatur officiis
voluptate, earum aliquid itaque modi officiis ullam a non nesciunt, explicabo
voluptates reiciendis hic mollitia cupiditate iste beatae earum tenetur, minus
doloribus amet esse?</p>\n<p>Tempore voluptas consectetur, doloribus at corporis
dolorem excepturi perferendis hic eaque? Sequi quibusdam tempore, iusto tempora
corrupti assumenda est unde fugiat quibusdam autem aliquam neque architecto, ea quam
sequi ratione similique officia veritatis, enim tempore perspiciatis corporis. Quis
fugiat assumenda minima at optio explicabo pariatur numquam dolorum, aspernatur
corrupti rerum.</p>\n<p>Eveniet id dolor nobis error doloribus atque, doloremque enim
impedit atque aperiam a placeat veniam ipsam debitis in fuga, perspiciatis cumque
commodi dolorum optio nulla a architecto magnam quas. Eos ducimus deserunt beatae
inventore sequi minus est quas temporibus, in facere accusantium nobis ullam dolorum
autem harum doloribus at ad, ea praesentium vitae temporibus esse rem delectus veniam
tempora at, asperiores ullam voluptatibus sunt saepe aut deleniti repudiandae rem
animi dolorem unde? Eos ipsum quae in possimus impedit quaerat illo sequi eum, optio
cumque sunt in iusto ex quas, consectetur quibusdam laborum molestiae perferendis
animi placeat, nemo libero rem ratione sint blanditiis commodi aliquid, minus
excepturi itaque cupiditate quisquam?</p>\n<p>Magnam nobis eaque odio repellat,
praesentium modi eligendi, officiis nisi neque porro vitae optio numquam
exercitationem delectus eos, minima quasi magni molestias quae vel officia
necessitatibus, quas voluptate minima. Illum cum impedit vitae consequatur dicta, vel
perspiciatis officiis quo, quisquam rerum voluptatum dolore eos consequatur ipsam
nemo, quos omnis et veniam at, alias dolor necessitatibus quibusdam ex modi
laboriosam. Sit laborum ex, earum facere nam fuga, in vel recusandae explicabo dicta
sunt commodi animi quod sed, quam quaerat ipsa animi rerum quia facere cumque, non
vero odit quas officia suscipit in culpa facere veniam aliquam? Iste numquam
cupiditate deserunt consequatur, odit possimus cumque incidunt delectus, possimus sunt
a labore quia rerum quo eaque repudiandae laborum, autem fuga ipsam nam neque,
distinctio quis facilis ratione.</p>\n<p>Consectetur enim et eligendi quod illo itaque
sit repudiandae veniam rerum vitae, rem repudiandae atque aut nobis officia officiis
quod totam voluptatem in laboriosam, atque laudantium veniam magnam perferendis
tenetur, aliquam ad reprehenderit rem inventore vero labore assumenda, ab nihil
consectetur accusamus a. Animi odit quia quod aliquid quam magni facilis, esse
laboriosam animi rem aspernatur quo molestias vitae.</p>\n<p>Eum nesciunt harum
corporis porro voluptates architecto error voluptate, velit laudantium repellat
consequatur, vel tempore eos officiis id dolores earum ex at suscipit, eius tempore
nobis, consectetur sequi incidunt nemo natus sed sapiente. Itaque tempora
reprehenderit, non architecto libero distinctio qui sed voluptatem quasi iusto sunt
corporis optio, officia ex doloremque in ut blanditiis magni possimus illum, amet fuga
quasi nostrum, ut eos ex est eum architecto optio itaque alias. Pariatur quod quos
laudantium impedit, officiis iusto quod corrupti vero dolorem.</p>\n<p>Perferendis
dicta exercitationem amet ullam eaque in possimus eligendi provident ratione officia,
```

```
earum velit ad deserunt eveniet, tempora id ut odio necessitatibus a, dolorum iste
labore rerum in dicta cum voluptates, aperiam quos rerum quibusdam
ducimus.</p>\n<p>Quam quo laudantium tenetur, porro vero distinctio asperiores ab
minus sint dolorum consequuntur deserunt iure, corporis cumque perferendis illum velit
veritatis odit asperiores natus enim at exercitationem, dolorem cum cupiditate iste,
quas explicabo deleniti voluptate neque? Vero explicabo recusandae esse voluptate
obcaecati velit fugiat quae cupiditate necessitatibus, pariatur alias ullam, numquam
dolores maxime corrupti debitis ea praesentium facere? Ratione earum qui beatae dolor
voluptatibus alias dicta repellat vero quis, ipsum ipsam impedit neque possimus quasi,
temporibus iure accusantium officia facilis quas necessitatibus ut asperiores,
exercitationem mollitia corporis ipsum sint tempore architecto quo ipsam laboriosam
dolor officiis, praesentium ullam voluptatum? Repellendus vitae distinctio cumque
architecto eius nesciunt voluptate, quisquam commodi eius at
dolorum.</p>\n<p>Praesentium saepe facilis iste cum, mollitia molestiae cum aperiam
fuga vitae sequi repellat nisi maxime perferendis quaerat, facilis corrupti fugit
necessitatibus deleniti esse error quaerat temporibus voluptatum magnam recusandae,
molestias ducimus temporibus aut culpa illo facere sed hic sequi voluptatem, autem
adipisci atque nemo quam sapiente voluptas ipsam molestias possimus in? Beatae eveniet
commodi similique incidunt nihil, doloremque necessitatibus ipsum similique laboriosam
aliquam delectus reprehenderit earum? Quisquam nostrum consequuntur provident quis et
possimus iusto, ad ea nisi eligendi debitis, alias at reiciendis accusantium officia
veniam similique, officia odit rerum iure?</p>",
    "id": 5,
    "is_watcher": false,
    "last_modifier": 6,
    "modified_date": "2020-07-03T08:40:58.045Z",
    "owner": 13,
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Beta project patch",
        "slug": "project-0"
    },
    "slug": "amet",
    "total_watchers": 0,
    "version": 2
}
```


48.56. Wiki page watcher detail


```
{
    "full_name": "Vanesa Torres",
    "id": 6,
    "username": "user2114747470430251528"
}
```

48.57. Wiki link

```
{
    "href": "numquam",
    "id": 1,
    "order": 1593690999327,
    "project": 1,
    "title": "numquam"
}
```

48.58. History entry comment

```
{
    "comment": "Dolore iste earum ut accusantium magni sit quidem reiciendis cum
quae?",
    "comment_html": "<p>Dolore iste earum ut accusantium magni sit quidem reiciendis
cum quae?</p>",
    "date": "2020-07-02T11:56:22.731Z",
    "user": {
        "big_photo": null,
        "bio": "",
        "color": "#40826D",
        "full_name": "Vanesa Torres",
        "full_name_display": "Vanesa Torres",
        "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
        "id": 6,
        "is_active": true,
        "lang": "",
        "photo": null,
        "roles": [
            "Design",
            "Front",
            "Product Owner",
            "UX"
        ],
        "theme": "",
        "timezone": "",
        "username": "user2114747470430251528"
    }
}
```

48.59. History entry

```
{
    "comment": "Dolore iste earum ut accusantium magni sit quidem reiciendis cum
quae?",
    "comment_html": "<p>Dolore iste earum ut accusantium magni sit quidem reiciendis
cum quae?</p>",
    "created_at": "2020-07-02T11:56:22.731Z",
    "delete_comment_date": null,
    "delete_comment_user": null,
    "diff": {},
    "edit_comment_date": null,
    "id": "0c94506a-bc5b-11ea-816c-e4a7a0b1f521",
    "is_hidden": false,
    "is_snapshot": false,
    "key": "userstories.userstory:2",
    "snapshot": null,
    "type": 1,
    "user": {
        "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
        "is_active": true,
        "name": "Mohamed Ortega",
        "photo": null,
        "pk": 13,
        "username": "user7"
    },
    "values": {
        "users": {}
    },
    "values_diff": {}
}
```


48.60. Notify policy


```
{
    "id": 7,
    "live_notify_level": 1,
    "notify_level": 2,
    "project": 1,
    "project_name": "Project Example 0",
    "web_notify_level": true
}
```


48.61. Feedback

```
{
    "comment": "Testing feedback",
    "created_date": "2020-07-03T08:40:59+0000",
    "email": "user2114747470430251528@taigaio.demo",
    "full_name": "Vanesa Torres",
    "id": 1
}
```

48.62. Export detail for synch mode

```
{
    "url": "http://localhost:8000/media/exports/1/project-0-
0c8920cf44704a9f8d39de7f8c52d321.json"
}
```

48.63. Export accepted response

```
{
    "export_id": "e338555a-3918-4203-8fc6-81b3f7933c79"
}
```

48.64. Import accepted response

```
{
    "import_id": "e338555a-3918-4203-8fc6-81b3f7933c79"
}
```

48.65. Webhook

```
{
    "id": 1,
    "key": "test-key",
    "logs_counter": 1,
    "name": "My service name",
    "project": 1,
    "url": "http://localhost:3000/htbin/test.py"
}
```

48.66. Webhook log

```
{
    "created": "2020-07-03T08:40:52.867Z",
    "duration": 0.0,
    "id": 1,
    "request_data": {
        "action": "test",
        "by": {
            "full_name": "Vanesa Torres",
            "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
            "id": 6,
            "permalink": "http://localhost:9001/profile/user2114747470430251528",
            "photo": null,
            "username": "user2114747470430251528"
        },
        "data": {
            "test": "test"
        },
        "date": "2020-07-03T08:40:52.864Z",
        "type": "test"
    },
    "request_headers": {
        "Content-Length": "318",
        "Content-Type": "application/json",
        "X-Hub-Signature": "sha1=6d5d0cf85bdb12e974dea282592952a3a575801d",
        "X-TAIGA-WEBHOOK-SIGNATURE": "6d5d0cf85bdb12e974dea282592952a3a575801d"
    },
    "response_data": "error-in-request: ('Connection aborted.',
RemoteDisconnected('Remote end closed connection without response'))",
    "response_headers": {},
    "status": 0,
    "url": "http://localhost:3000/htbin/test.py",
    "webhook": 1
}
```


48.67. Timeline entry detail


```
{
    "content_type": 13,
    "created": "2020-07-03T08:40:37.973Z",
    "data": {
        "comment": "",
        "comment_html": "",
        "milestone": {
            "id": 1,
            "name": "Sprint 2020-5-8",
            "slug": "sprint-2020-5-8"
        },
        "project": {
```

```
            "description": "Project example 0 description",
            "id": 1,
            "name": "Project Example 0",
            "slug": "project-0"
        },
        "user": {
            "big_photo": null,
            "date_joined": "2020-07-02T11:56:19.209Z",
            "gravatar_id": "b579f05d7d36f4588b11887093e4ce44",
            "id": 6,
            "is_profile_visible": true,
            "name": "Vanesa Torres",
            "photo": null,
            "username": "user2114747470430251528"
        },
        "userstory": {
            "id": 1,
            "ref": 1,
            "subject": "Patching subject"
        },
        "values_diff": {
            "attachments": {
                "changed": [
                    {
                        "changes": {
                            "description": [
                                "mollitia illo sed expedita voluptates necessitatibus
ratione sapiente laudantium",
                                "patching description"
                            ]
                        },
                        "filename": "sample_attachment_1.txt",
                        "thumb_url": null,
                        "url":
"http://localhost:8000/media/attachments/1/e/d/0/fbfccd7f166b43a4a9fceeb6447cd3b823264
ba72824b09d08047cc71149/sample_attachment_1.txt"
                    }
                ],
                "deleted": [],
                "new": []
            }
        }
    },
    "data_content_type": 49,
    "event_type": "userstories.userstory.change",
    "id": 7893,
    "namespace": "project:1",
    "object_id": 1,
    "project": 1
}
```

48.68. Locale


```
[
    {
        "bidi": false,
        "code": "ca",
        "name": "Catal\u00e0"
    },
    {
        "bidi": false,
        "code": "de",
        "name": "Deutsch"
    },
    {
        "bidi": false,
        "code": "en",
        "name": "English (US)"
    },
    {
        "bidi": false,
        "code": "es",
        "name": "Espa\u00f1ol"
    },
    {
        "bidi": false,
        "code": "eu",
        "name": "Euskara"
    },
    {
        "bidi": true,
        "code": "fa",
        "name": "\u0641\u0627\u0631\u0633\u06cc\u200f"
    },
    {
        "bidi": false,
        "code": "fi",
        "name": "Suomi"
    },
    {
        "bidi": false,
        "code": "fr",
        "name": "Fran\u00e7ais"
    },
    {
        "bidi": true,
        "code": "he",
        "name": "\u05e2\u05d1\u05e8\u05d9\u05ea\u200f"
    },
    {
        "bidi": false,
```

```
        "code": "it",
        "name": "Italiano"
    },
    {
        "bidi": false,
        "code": "ja",
        "name": "\u65e5\u672c\u8a9e"
    },
    {
        "bidi": false,
        "code": "ko",
        "name": "\ud55c\uad6d\uc5b4"
    },
    {
        "bidi": false,
        "code": "lv",
        "name": "Latvie\u0161u"
    },
    {
        "bidi": false,
        "code": "nb",
        "name": "Norsk (bokm\u00e5l)"
    },
    {
        "bidi": false,
        "code": "nl",
        "name": "Nederlands"
    },
    {
        "bidi": false,
        "code": "pl",
        "name": "Polski"
    },
    {
        "bidi": false,
        "code": "pt-br",
        "name": "Portugu\u00eas (Brasil)"
    },
    {
        "bidi": false,
        "code": "ru",
        "name": "\u0420\u0443\u0441\u0441\u043a\u0438\u0439"
    },
    {
        "bidi": false,
        "code": "sv",
        "name": "Svenska"
    },
    {
        "bidi": false,
        "code": "tr",
```

```
        "name": "T\u00fcrk\u00e7e"
    },
    {
        "bidi": false,
        "code": "uk",
        "name": "\u0423\u043a\u0440\u0430\u0457\u043d\u0441\u044c\u043a\u0430"
    },
    {
        "bidi": false,
        "code": "zh-hans",
        "name": "\u4e2d\u6587(\u7b80\u4f53)"
    },
    {
        "bidi": false,
        "code": "zh-hant",
        "name": "\u4e2d\u6587(\u9999\u6e2f)"
    }
]
```


48.69. Watched

```
{
    "assigned_to": 12,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Vanesa Garcia",
        "gravatar_id": "74cb769a5e64d445b8550789e1553502",
        "id": 12,
        "is_active": true,
        "photo": null,
        "username": "user6"
    },
    "created_date": "2020-07-02T11:59:19.011Z",
    "description": null,
    "id": 25,
    "is_private": null,
    "is_voter": false,
    "is_watcher": false,
    "logo_small_url": null,
    "name": null,
    "project": 7,
    "project_blocked_code": "blocked-by-staff",
    "project_is_private": false,
    "project_name": "Project Example 6",
    "project_slug": "project-6",
    "ref": 50,
    "slug": null,
    "status": "Ready for test",
    "status_color": "#fcc000",
    "subject": "Fixing templates for Django 1.6.",
    "tags_colors": [
        {
            "color": "#c4f027",
            "name": "officia"
        },
        {
            "color": null,
            "name": "perferendis"
        },
        {
            "color": null,
            "name": "minima"
        }
    ],
    "total_voters": 2,
    "total_watchers": 1,
    "type": "epic"
}
```

48.70. Liked


```
[
    {
        "assigned_to": null,
        "assigned_to_extra_info": null,
        "created_date": "2020-07-03T08:40:57.282Z",
        "description": "Beta description",
        "id": 1,
        "is_fan": true,
        "is_private": true,
        "is_watcher": false,
        "logo_small_url": null,
        "name": "Beta project patch",
        "project": null,
        "project_blocked_code": null,
        "project_is_private": null,
        "project_name": null,
        "project_slug": null,
        "ref": null,
        "slug": "project-0",
        "status": null,
        "status_color": null,
        "subject": null,
        "tags_colors": [],
        "total_fans": 10,
        "total_watchers": 15,
        "type": "project"
    },
    {
        "assigned_to": null,
        "assigned_to_extra_info": null,
        "created_date": "2020-07-02T11:57:21.421Z",
        "description": "Project example 2 description",
        "id": 3,
        "is_fan": true,
        "is_private": true,
        "is_watcher": true,
        "logo_small_url": null,
        "name": "Project Example 2",
        "project": null,
        "project_blocked_code": null,
        "project_is_private": null,
        "project_name": null,
        "project_slug": null,
        "ref": null,
        "slug": "project-2",
        "status": null,
        "status_color": null,
        "subject": null,
```

```
        "tags_colors": [
            {
                "color": null,
                "name": "consequatur"
            }
        ],
        "total_fans": 11,
        "total_watchers": 15,
        "type": "project"
    }
]
```


48.71. Voted


```
{
    "assigned_to": 5,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Administrator",
        "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
        "id": 5,
        "is_active": true,
        "photo": null,
        "username": "admin"
    },
    "created_date": "2020-07-02T11:59:14.599Z",
    "description": null,
    "id": 88,
    "is_private": null,
    "is_voter": true,
    "is_watcher": false,
    "logo_small_url": null,
    "name": null,
    "project": 7,
    "project_blocked_code": "blocked-by-staff",
    "project_is_private": false,
    "project_name": "Project Example 6",
    "project_slug": "project-6",
    "ref": 46,
    "slug": null,
    "status": "Needs Info",
    "status_color": "#89BAB4",
    "subject": "Create the user model",
    "tags_colors": [
        {
            "color": "#9f6274",
            "name": "sequi"
        },
        {
```

```
            "color": null,
            "name": "dolorum"
        },
        {
            "color": null,
            "name": "voluptate"
        },
        {
            "color": "#7b0e4e",
            "name": "pariatur"
        },
        {
            "color": "#47e087",
            "name": "recusandae"
        },
        {
            "color": null,
            "name": "at"
        },
        {
            "color": "#86f7e4",
            "name": "a"
        },
        {
            "color": null,
            "name": "velit"
        },
        {
            "color": null,
            "name": "alias"
        },
        {
            "color": null,
            "name": "incidunt"
        }
    ],
    "total_voters": 9,
    "total_watchers": 3,
    "type": "issue"
}
```


48.72. Contact

```
{
    "comment": "Comment to admins",
    "created_date": "2020-07-03T08:40:56+0000",
    "id": 1,
    "project": 3,
    "user": 6
}
```

48.73. Discover stats

```
{
    "projects": {
        "total": 6
    }
}
```

48.74. System stats

```
{
    "projects": {
        "average_last_five_working_days": 0.0,
        "average_last_seven_days": 0.0,
        "percent_with_backlog": 0.0,
        "percent_with_backlog_and_kanban": 100.0,
        "percent_with_kanban": 0.0,
        "today": 0,
        "total": 8,
        "total_with_backlog": 0,
        "total_with_backlog_and_kanban": 8,
        "total_with_kanban": 0
    },
    "users": {
        "average_last_five_working_days": 0.0,
        "average_last_seven_days": 0.0,
        "counts_last_year_per_week": {
            "2020-06-29": 13
        },
        "today": 2,
        "total": 13
    },
    "userstories": {
        "average_last_five_working_days": 0.0,
        "average_last_seven_days": 0.0,
        "today": 8,
        "total": 172
    }
}
```


48.75. Importer Trello Auth Url


```
{
    "url":
"https://trello.com/1/OAuthAuthorizeToken?oauth_token=b00000000000000000a0&scope=read
,write,account&expiration=1day&name=Taiga&return_url=http://localhost:9001/project/new
/import/trello"
}
```


48.76. Importer Trello Auth Token


```
{
    "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

48.77. Importer Trello list users

```
[
    {
        "email": null,
        "full_name": "Trello user",
        "id": "trello-user",
        "user": null
    },
    {
        "email": "other-trello-user@email.com",
        "full_name": "Other Trello user",
        "id": "other-trello-user",
        "user": {
            "full_name": "Taiga user",
            "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
            "id": 12345,
            "photo": "/user-photo-url"
        }
    }
]
```

48.78. Importer Trello list projects

```
[
    {
        "description": "My trello project",
        "id": "123ABC",
        "is_private": false,
        "name": "Trello project"
    },
    {
        "description": "My other trello project",
        "id": "ABC123",
        "is_private": true,
        "name": "Other trello project"
    }
]
```

48.79. Importer Trello import project

```
{
    "is_backlog_activated": false,
    "is_kanban_activated": true,
    "my_permissions": [
        "view_us"
    ],
    "slug": "my-username-new-project-name"
}
```

48.80. Importer Github Auth Url

```
{
    "url":
"https://github.com/login/oauth/authorize?client_id=XXXXXX_get_a_valid_client_id_from_
github_XXXXXX&scope=user,repo"
}
```

48.81. Importer Github Auth Token

```
{
    "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

48.82. Importer Github list users

```
[
    {
        "full_name": "Github user",
        "id": 12345,
        "user": null,
        "username": "github-user"
    },
    {
        "full_name": "Other Github user",
        "id": 12345,
        "user": {
            "full_name": "Taiga user",
            "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
            "id": 54321,
            "photo": "/user-photo-url"
        },
        "username": "other-github-user"
    }
]
```

48.83. Importer Github list projects

```
[
    {
        "description": "My github project",
        "id": "user/project",
        "is_private": false,
        "name": "Github project"
    },
    {
        "description": "My other github project",
        "id": "user/other-project",
        "is_private": true,
        "name": "Other github project"
    }
]
```

48.84. Importer Github import project

```
{
    "is_backlog_activated": false,
    "is_kanban_activated": true,
    "my_permissions": [
        "view_us"
    ],
    "slug": "my-username-new-project-name"
}
```

48.85. Importer Jira Auth Url

```
{
    "url":
"http://your.jira.server/plugins/servlet/oauth/authorize?oauth_token=xxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx"
}
```

48.86. Importer Jira Auth Token

```
{
    "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "url": "http://your.jira.server"
}
```

48.87. Importer Jira list users

```
[
    {
        "email": null,
        "full_name": "Jira user",
        "id": "jira-user",
        "user": null
    },
    {
        "email": "other-jira-user@email.com",
        "full_name": "Other Jira user",
        "id": "other-jira-user",
        "user": {
            "full_name": "Taiga user",
            "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
            "id": 12345,
            "photo": "/user-photo-url"
        }
    }
]
```

48.88. Importer Jira list projects

```
[
    {
        "description": "My jira project",
        "id": "123",
        "is_private": false,
        "name": "Jira project",
        "type": "project"
    },
    {
        "description": "My other jira project",
        "id": "456",
        "is_private": true,
        "name": "Other jira project",
        "type": "board"
    }
]
```

48.89. Importer Jira import project

```
{
    "is_backlog_activated": false,
    "is_kanban_activated": true,
    "my_permissions": [
        "view_us"
    ],
    "slug": "my-username-new-project-name"
}
```

48.90. Importer Import project task accepted

```
{
    "task_id": "00000000-0000-0000-0000-000000000000"
}
```

49. Contrib plugins

Taiga allows adding features through contrib plugins, each plugin can add new API endpoints, and has its own documentation.

Current supported contrib plugins that adding endpoints:

- taiga-contrib-slack: Slack integration (documentation)
- taiga-contrib-hall: Hall.com integration (documentation)