Taiga REST API

Table of Contents

1. General notes . . . 1
   1.1. Authentication . . . 1
   1.2. OCC - Optimistic concurrency control . . . 6
   1.3. Pagination . . . 6
   1.4. Internationalization . . . 6
   1.5. Throttling . . . 7
   1.6. Read only fields . . . 7
2. Endpoints Summary . . . 7
   2.1. Auth . . . 7
   2.2. Applications . . . 7
   2.3. Application Tokens . . . 7
   2.4. Resolver . . . 8
   2.5. Searches . . . 8
   2.6. User storage . . . 8
   2.7. Project templates . . . 8
   2.8. Projects . . . 9
   2.9. Memberships/Invitations . . . 11
   2.10. Milestones . . . 11
   2.11. Epics . . . 12
   2.12. Epic status . . . 13
   2.13. Epic custom attribute . . . 13
   2.14. Epic custom attributes values . . . 14
   2.15. User stories . . . 14
   2.16. User story status . . . 16
   2.17. Points . . . 16
   2.18. User story custom attribute . . . 16
   2.19. User story custom attributes values . . . 17
   2.20. Tasks . . . 17
   2.21. Task status . . . 18
   2.22. Task custom attribute . . . 19
   2.23. Task custom attributes values . . . 19
   2.24. Issues . . . 20
   2.25. Issue status . . . 21
   2.26. Issue types . . . 21
   2.27. Priorities . . . 21
   2.28. Severities . . . 22

2.29. Issue custom attribute . . . 22
2.30. Issue custom attributes values . . . 23
2.31. Wiki pages . . . 23
2.32. Wiki links . . . 24
2.33. History . . . 24
2.34. Users . . . 25
2.35. Notify policies . . . 26
2.36. Feedback . . . 26
2.37. Export/Import . . . 27
2.38. Webhooks . . . 27
2.39. Timelines . . . 27
2.40. Locales . . . 27
2.41. Stats . . . 28
3. Auth . . . 28
3.1. Normal login . . . 28
3.2. Github login . . . 28
3.3. Public registry . . . 29
3.4. Private registry . . . 30
4. Applications . . . 30
4.1. Get . . . 30
4.2. Get token . . . 31
5. Application tokens . . . 31
5.1. List . . . 31
5.2. Get . . . 31
5.3. Delete . . . 32
5.4. Authorize . . . 32
5.5. Validate . . . 33
6. Resolver . . . 33
6.1. Projects . . . 33
6.2. User stories . . . 34
6.3. Issues . . . 34
6.4. Tasks . . . 35
6.5. Milestones . . . 35
6.6. Wiki pages . . . 36
6.7. Multiple resolution . . . 36
6.8. By ref value . . . 36
7. Searches . . . 37
7.1. Search . . . 37

8. User storage . . . 37
8.1. List . . . 37
8.2. Create . . . 38
8.3. Get . . . 38
8.4. Edit . . . 38
8.5. Delete . . . 39
9. Project templates . . . 39
9.1. List . . . 39
9.2. Create . . . 39
9.3. Get . . . 52
9.4. Edit . . . 52
9.5. Delete . . . 52
10. Projects . . . 52
10.1. List . . . 53
10.2. Create . . . 54
10.3. Get . . . 55
10.4. Get by slug . . . 55
10.5. Edit . . . 56
10.6. Delete . . . 56
10.7. Bulk update order . . . 56
10.8. Get modules configuration . . . 57
10.9. Edit modules configuration . . . 57
10.10. Stats . . . 58
10.11. Issue stats . . . 58
10.12. Tag colors . . . 71
10.13. Create tag . . . 72
10.14. Edit tag . . . 72
10.15. Delete-tag . . . 72
10.16. Mix tags . . . 73
10.17. Like a project . . . 73
10.18. Unlike a project . . . 74
10.19. List project fans . . . 74
10.20. Watch a project . . . 74
10.21. Stop watching project . . . 74
10.22. List project watchers . . . 75
10.23. Create template . . . 75
10.24. Leave . . . 75
10.25. Change logo . . . 76

10.26. Remove logo . . . 76
10.27. Transfer validate-token . . . 76
10.28. Transfer request . . . 77
10.29. Transfer start . . . 77
10.30. Transfer accept . . . 77
10.31. Transfer reject . . . 78
11. Memberships/Invitations . . . 78
11.1. List . . . 78
11.2. Create . . . 79
11.3. Bulk creation . . . 79
11.4. Get . . . 80
11.5. Edit . . . 80
11.6. Delete . . . 81
11.7. Resend invitation . . . 81
11.8. Get Invitation (by token) . . . 81
12. Milestones . . . 82
12.1. List . . . 82
12.2. Create . . . 82
12.3. Get . . . 83
12.4. Edit . . . 84
12.5. Delete . . . 84
12.6. Stats . . . 84
12.7. Watch a milestone . . . 85
12.8. Stop watching a milestone . . . 85
12.9. List milestone watchers . . . 85
13. Epics . . . 86
13.1. List . . . 86
13.2. Create . . . 86
13.3. Get . . . 87
13.4. Get by ref . . . 88
13.5. Edit . . . 88
13.6. Delete . . . 88
13.7. Bulk creation . . . 89
13.8. Filters data . . . 89
13.9. List related userstories . . . 89
13.10. Create related userstory . . . 90
13.11. Get related userstory . . . 90
13.12. Edit related userstory . . . 91

13.13. Delete related userstory . . . 91
13.14. Bulk related userstories creation . . . 91
13.15. Vote an epic . . . 92
13.16. Remove vote from an epic . . . 92
13.17. Get epic voters list . . . 92
13.18. Watch an epic . . . 93
13.19. Stop watching an epic . . . 93
13.20. List epic watchers . . . 93
13.21. List attachments . . . 94
13.22. Create attachment . . . 94
13.23. Get attachment . . . 94
13.24. Edit attachment . . . 95
13.25. Delete attachment . . . 95
14. Epic status . . . 95
14.1. List . . . 95
14.2. Create . . . 96
14.3. Get . . . 97
14.4. Edit . . . 97
14.5. Delete . . . 97
14.6. Bulk update order . . . 98
15. Epic custom attribute . . . 98
15.1. List . . . 98
15.2. Create . . . 99
15.3. Get . . . 100
15.4. Edit . . . 100
15.5. Delete . . . 101
15.6. Bulk update order . . . 101
16. Epic custom attributes values . . . 101
16.1. Get . . . 102
16.2. Edit . . . 102
17. User stories . . . 102
17.1. List . . . 102
17.2. Create . . . 103
17.3. Get . . . 105
17.4. Get by ref . . . 105
17.5. Edit . . . 105
17.6. Delete . . . 106
17.7. Bulk creation . . . 106

17.8. Bulk update backlog order . . . 107
17.9. Bulk update kanban order . . . 107
17.10. Bulk update sprint order . . . 108
17.11. Bulk update milestone . . . 109
17.12. Filters data . . . 110
17.13. Vote a user story . . . 110
17.14. Remove vote from a user story . . . 111
17.15. Get user story voters list . . . 111
17.16. Watch a user story . . . 111
17.17. Stop watching a user story . . . 111
17.18. List user story watchers . . . 112
17.19. List attachments . . . 112
17.20. Create attachment . . . 112
17.21. Get attachment . . . 113
17.22. Edit attachment . . . 113
17.23. Delete attachment . . . 114
18. User story status . . . 114
18.1. List . . . 114
18.2. Create . . . 114
18.3. Get . . . 115
18.4. Edit . . . 116
18.5. Delete . . . 116
18.6. Bulk update order . . . 116
19. Points . . . 117
19.1. List . . . 117
19.2. Create . . . 118
19.3. Get . . . 118
19.4. Edit . . . 119
19.5. Delete . . . 119
19.6. Bulk update order . . . 119
20. User story custom attribute . . . 120
20.1. List . . . 120
20.2. Create . . . 121
20.3. Get . . . 121
20.4. Edit . . . 122
20.5. Delete . . . 122
20.6. Bulk update order . . . 122
21. User story custom attributes values . . . 123

21.1. Get . . . 123
21.2. Edit . . . 123
22. Tasks . . . 124
22.1. List . . . 124
22.2. Create . . . 125
22.3. Get . . . 126
22.4. Get by ref . . . 127
22.5. Edit . . . 129
22.6. Delete . . . 130
22.7. Bulk creation . . . 130
22.8. Filters data . . . 130
22.9. Vote a task . . . 131
22.10. Remove vote from a task . . . 131
22.11. Get task voters list . . . 131
22.12. Watch a task . . . 132
22.13. Stop watching a task . . . 132
22.14. List task watchers . . . 132
22.15. List attachments . . . 132
22.16. Create attachment . . . 133
22.17. Get attachment . . . 133
22.18. Edit attachment . . . 134
22.19. Delete attachment . . . 134
23. Task status . . . 134
23.1. List . . . 134
23.2. Create . . . 135
23.3. Get . . . 136
23.4. Edit . . . 136
23.5. Delete . . . 136
23.6. Bulk update order . . . 136
24. Task custom attribute . . . 137
24.1. List . . . 137
24.2. Create . . . 138
24.3. Get . . . 138
24.4. Edit . . . 139
24.5. Delete . . . 139
24.6. Bulk update order . . . 139
25. Task custom attributes values . . . 140
25.1. Get . . . 140

25.2. Edit . . . 140
26. Issues . . . 141
26.1. List . . . 141
26.2. Create . . . 142
26.3. Get . . . 144
26.4. Get by ref . . . 144
26.5. Edit . . . 144
26.6. Delete . . . 145
26.7. Filters data . . . 145
26.8. Vote an issue . . . 145
26.9. Remove vote from an issue . . . 145
26.10. Get issue voters list . . . 146
26.11. Watch an issue . . . 146
26.12. Stop watching an issue . . . 146
26.13. List issue watchers . . . 147
26.14. List attachments . . . 147
26.15. Create attachment . . . 147
26.16. Get attachment . . . 148
26.17. Edit attachment . . . 148
26.18. Delete attachment . . . 148
27. Issue status . . . 149
27.1. List . . . 149
27.2. Create . . . 149
27.3. Get . . . 150
27.4. Edit . . . 150
27.5. Delete . . . 151
27.6. Bulk update order . . . 151
28. Issue types . . . 152
28.1. List . . . 152
28.2. Create . . . 153
28.3. Get . . . 153
28.4. Edit . . . 154
28.5. Delete . . . 154
28.6. Bulk update order . . . 154
29. Priorities . . . 155
29.1. List . . . 155
29.2. Create . . . 156
29.3. Get . . . 156

29.4. Edit . . . 157
29.5. Delete . . . 157
29.6. Bulk update order . . . 157
30. Severities . . . 158
30.1. List . . . 158
30.2. Create . . . 159
30.3. Get . . . 159
30.4. Edit . . . 160
30.5. Delete . . . 160
30.6. Bulk update order . . . 160
31. Issue custom attribute . . . 161
31.1. List . . . 161
31.2. Create . . . 162
31.3. Get . . . 162
31.4. Edit . . . 163
31.5. Delete . . . 163
31.6. Bulk update order . . . 163
32. Issue custom attributes values . . . 164
32.1. Get . . . 164
32.2. Edit . . . 164
33. Wiki pages . . . 165
33.1. List . . . 165
33.2. Create . . . 165
33.3. Get . . . 166
33.4. Get by slug . . . 166
33.5. Edit . . . 167
33.6. Delete . . . 167
33.7. Watch a wiki page . . . 167
33.8. Stop watching a wiki page . . . 168
33.9. List wiki page watchers . . . 168
33.10. List attachments . . . 168
33.11. Create attachment . . . 169
33.12. Get attachment . . . 169
33.13. Edit attachment . . . 170
33.14. Delete attachment . . . 170
34. Wiki links . . . 170
34.1. List . . . 170
34.2. Create . . . 171

34.3. Get . . . 172
34.4. Edit . . . 172
34.5. Delete . . . 173
35. History . . . 173
35.1. Get user story, task, issue or wiki page history . . . 173
35.2. Get comment versions . . . 173
35.3. Edit comment . . . 174
35.4. Delete comment . . . 174
35.5. Undelete comment . . . 174
36. Users . . . 175
36.1. List . . . 175
36.2. Get . . . 175
36.3. Me . . . 175
36.4. Get user stats . . . 176
36.5. Get watched content . . . 176
36.6. Get liked content . . . 177
36.7. Get voted content . . . 177
36.8. Edit . . . 178
36.9. Delete . . . 178
36.10. Get contacts . . . 178
36.11. Cancel . . . 178
36.12. Change avatar . . . 179
36.13. Remove avatar . . . 179
36.14. Change email . . . 179
36.15. Change password . . . 180
36.16. Password recovery . . . 180
36.17. Change password from recovery . . . 181
37. Notify policies . . . 181
37.1. List . . . 181
37.2. Get . . . 181
37.3. Edit . . . 182
38. Feedback . . . 182
38.1. Create . . . 182
39. Export/Import . . . 183
39.1. Export . . . 183
39.2. Import . . . 183
40. Webhooks . . . 183
40.1. List . . . 184

40.2. Create . . . 184
40.3. Get . . . 185
40.4. Edit . . . 185
40.5. Delete . . . 185
40.6. Test . . . 186
40.7. Logs list . . . 186
40.8. Log get . . . 186
40.9. Resend request . . . 187
41. Timelines . . . 187
41.1. List user timeline . . . 187
41.2. List profile timeline . . . 187
41.3. List project timeline . . . 188
42. Locales . . . 188
42.1. List . . . 188
43. Stats . . . 188
43.1. Get discover stats . . . 188
43.2. Get system stats . . . 189
44. Objects Summary . . . 189
44.1. Attachment . . . 189
44.2. Application token object . . . 190
44.3. Authorization code object . . . 190
44.4. Cyphered token object . . . 190
44.5. User detail . . . 191
44.6. User contact detail . . . 192
44.7. User authentication-detail . . . 193
44.8. User stats detail . . . 194
44.9. Search results detail . . . 195
44.10. User storage data . . . 209
45. Project templates detail . . . 209
45.1. Project list entry . . . 221
45.2. Project detail . . . 224
45.3. Project modules configuration . . . 250
45.4. Project stats detail . . . 251
45.5. Project issue stats detail . . . 253
45.6. Project tag colors data detail . . . 267
45.7. Project voter detail . . . 270
45.8. Project watcher detail . . . 271
45.9. Membership detail . . . 271

45.10. Milestone detail . . . 271
45.11. Milestone watcher detail . . . 283
45.12. Milestone stats detail . . . 283
45.13. Epic detail . . . 286
45.14. Epic detail (GET) . . . 288
45.15. Epic detail (LIST) . . . 290
45.16. Epic filters data detail . . . 291
45.17. Epic voter detail . . . 312
45.18. Epic watcher detail . . . 312
45.19. Epic status detail . . . 312
45.20. Epic custom attribute detail . . . 313
45.21. Epic custom attributes values detail . . . 313
45.22. Epic related user story detail . . . 313
45.23. User story detail . . . 314
45.24. User story detail (GET) . . . 316
45.25. User story detail (LIST) . . . 318
45.26. Issue filters data detail . . . 320
45.27. User story voter detail . . . 344
45.28. User story watcher detail . . . 344
45.29. User story status detail . . . 344
45.30. Point detail . . . 344
45.31. User story custom attribute detail . . . 344
45.32. User story custom attributes values detail . . . 345
45.33. Task detail . . . 345
45.34. Task detail (GET) . . . 347
45.35. Task detail (LIST) . . . 350
45.36. Task filters data detail . . . 352
45.37. Task voter detail . . . 357
45.38. Task watcher detail . . . 357
45.39. Task status detail . . . 357
45.40. Task custom attribute detail . . . 357
45.41. Task custom attributes values detail . . . 358
45.42. Issue detail . . . 358
45.43. Issue detail (GET) . . . 361
45.44. Issue detail (LIST) . . . 363
45.45. Issue filters data detail . . . 365
45.46. Issue voters detail . . . 389
45.47. Issue watchers detail . . . 389

45.48. Issue status detail . . . 389
45.49. Issue type detail . . . 390
45.50. Priority detail . . . 390
45.51. Severity detail . . . 390
45.52. Issue custom attribute detail . . . 391
45.53. Issue custom attributes values detail . . . 391
45.54. Wiki page . . . 391
45.55. Wiki page watcher detail . . . 394
45.56. Wiki link . . . 394
45.57. History entry comment . . . 394
45.58. History entry . . . 396
45.59. Notify policy . . . 396
45.60. Feedback . . . 396
45.61. Export detail for synch mode . . . 397
45.62. Export accepted response . . . 397
45.63. Import accepted response . . . 397
45.64. Webhook . . . 397
45.65. Webhook log . . . 398
45.66. Timeline entry detail . . . 398
45.67. Locale . . . 400
45.68. Watched . . . 401
45.69. Liked . . . 403
45.70. Voted . . . 405
45.71. Discover stats . . . 406
45.72. System stats . . . 406
46. Contrib plugins . . . 407

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:

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

1.1.2. Application token authentication

This kind of tokes 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_QPCO5mWhT0EI9RRV
45xA7vbWoxeBIjiCL3qwAmlzg.bBWVKwGTkta5y99c.ArycfFtrlmWgyZ4lwXw_JiSVmkn9YF6Xwlh8nVDku0BLW8
kvaxNy3XRbbb17MtZ7mg.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

URL	Method	Functionality
/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
/api/v1/project-templates	POST	Create project template

URL	Method	Functionality
/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

URL	Method	Functionality
/api/v1/projects/{projectId}/edit_tag	POST	Edit project tag
/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

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. 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
/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

URL	Method	Functionality
/api/v1/milestones/{milestoneId}/watchers	GET	Get milestone watchers

2.11. 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
/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

URL	Method	Functionality
/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

2.12. 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

2.13. Epic custom attribute

URL	Method	Functionality
/api/v1/epic-custom-attributes	GET	List epic custom attributes

URL	Method	Functionality
/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

2.14. 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

2.15. 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
/api/v1/userstories/{userStoryId}	PUT	Modify user story

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

2.16. 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

2.17. 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

2.18. 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

2.19. 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

2.20. 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

URL	Method	Functionality
/api/v1/tasks/{taskId}	DELETE	Delete a task
/api/v1/tasks/bulk_create	POST	Create tasks on bulk mode
/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

2.21. 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

URL	Method	Functionality
/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

2.22. 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

2.23. 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

2.24. 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
/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

2.25. 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
/api/v1/issue-statuses/bulk_update_order	POST	Update issue statuses order in bulk mode

2.26. 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

2.27. Priorities

URL	Method	Functionality
/api/v1/priorities	GET	List priorities
/api/v1/priorities	POST	Create priority
/api/v1/priorities/{priorityId}	GET	Get priority

URL	Method	Functionality
/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

2.28. Severities

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

2.29. 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

URL	Method	Functionality
/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

2.30. 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

2.31. 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

URL	Method	Functionality
/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

2.32. 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

2.33. 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

URL	Method	Functionality
/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
/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

2.34. Users

URL	Method	Functionality
/api/v1/users	GET	List users
/api/v1/users/{userId}	GET	Get user

URL	Method	Functionality
/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

2.35. Notify policies

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

2.36. Feedback

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

2.37. Export/Import

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

2.38. 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

2.39. 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

2.40. Locales

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

2.41. Stats

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

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" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-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)

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "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" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "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:

```
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" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-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=119
```

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

```
{
    "issue": 23,
    "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-2016-8-20
```

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**: (talky|appear-in)
- **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**: appear-in or talky, 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": "appear-in",
        "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


```
{
    "closed_issues": 16,
    "issues_per_assigned_to": {
        "0": {
            "color": "black",
            "count": 8,
            "id": 0,
            "name": "No asignado",
            "username": "No asignado"
        },
        "10": {
            "color": "#67CF00",
            "count": 1,
            "id": 10,
            "name": "Marta Carmona",
            "username": "user4"
        },
        "11": {
            "color": "#FFFF00",
            "count": 1,
            "id": 11,
            "name": "German Benitez",
            "username": "user5"
        },
        "12": {
            "color": "#71A6D2",
            "count": 1,
            "id": 12,
            "name": "Pilar Herrera",
            "username": "user6"
```

```
        },
        "13": {
            "color": "#002e33",
            "count": 3,
            "id": 13,
            "name": "Alvaro Molina",
            "username": "user7"
        },
        "14": {
            "color": "#FFCC00",
            "count": 2,
            "id": 14,
            "name": "Andrea Fernandez",
            "username": "user8"
        },
        "15": {
            "color": "#C0FF33",
            "count": 3,
            "id": 15,
            "name": "Catalina Roman",
            "username": "user9"
        },
        "5": {
            "color": "",
            "count": 1,
            "id": 5,
            "name": "Administrator",
            "username": "admin"
        },
        "6": {
            "color": "#4B0082",
            "count": 2,
            "id": 6,
            "name": "Silvia Soto",
            "username": "user6532909695705815086"
        },
        "7": {
            "color": "#B6DA55",
            "count": 2,
            "id": 7,
            "name": "Marcos Ortiz",
            "username": "user1"
        },
        "8": {
            "color": "#D70A53",
            "count": 3,
            "id": 8,
            "name": "Alba Leon",
```

```
            "username": "user2"
        },
        "9": {
            "color": "#FFF8E7",
            "count": 1,
            "id": 9,
            "name": "Esther Ferrer",
            "username": "user3"
        }
    },
    "issues_per_owner": {
        "10": {
            "color": "#67CF00",
            "count": 1,
            "id": 10,
            "name": "Marta Carmona",
            "username": "user4"
        },
        "11": {
            "color": "#FFFF00",
            "count": 1,
            "id": 11,
            "name": "German Benitez",
            "username": "user5"
        },
        "12": {
            "color": "#71A6D2",
            "count": 2,
            "id": 12,
            "name": "Pilar Herrera",
            "username": "user6"
        },
        "13": {
            "color": "#002e33",
            "count": 3,
            "id": 13,
            "name": "Alvaro Molina",
            "username": "user7"
        },
        "14": {
            "color": "#FFCC00",
            "count": 1,
            "id": 14,
            "name": "Andrea Fernandez",
            "username": "user8"
        },
        "15": {
            "color": "#C0FF33",
```

```
            "count": 2,
            "id": 15,
            "name": "Catalina Roman",
            "username": "user9"
        },
        "5": {
            "color": "",
            "count": 3,
            "id": 5,
            "name": "Administrator",
            "username": "admin"
        },
        "6": {
            "color": "#4B0082",
            "count": 9,
            "id": 6,
            "name": "Silvia Soto",
            "username": "user6532909695705815086"
        },
        "7": {
            "color": "#B6DA55",
            "count": 1,
            "id": 7,
            "name": "Marcos Ortiz",
            "username": "user1"
        },
        "8": {
            "color": "#D70A53",
            "count": 3,
            "id": 8,
            "name": "Alba Leon",
            "username": "user2"
        },
        "9": {
            "color": "#FFF8E7",
            "count": 2,
            "id": 9,
            "name": "Esther Ferrer",
            "username": "user3"
        }
    },
    "issues_per_priority": {
        "1": {
            "color": "#666666",
            "count": 13,
            "id": 1,
            "name": "Patch name"
        },
```

```
        "2": {
            "color": "#669933",
            "count": 10,
            "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": 7,
            "id": 2,
            "name": "Minor"
        },
        "3": {
            "color": "#0000FF",
            "count": 7,
            "id": 3,
            "name": "Normal"
        },
        "4": {
            "color": "#FFA500",
            "count": 2,
            "id": 4,
            "name": "Important"
        },
        "5": {
            "color": "#CC0000",
            "count": 7,
            "id": 5,
            "name": "Critical"
        }
    },
    "issues_per_status": {
        "1": {
            "color": "#8C2318",
```

```
            "count": 6,
            "id": 1,
            "name": "Patch status name"
        },
        "2": {
            "color": "#5E8C6A",
            "count": 2,
            "id": 2,
            "name": "In progress"
        },
        "3": {
            "color": "#88A65E",
            "count": 2,
            "id": 3,
            "name": "Ready for test"
        },
        "4": {
            "color": "#BFB35A",
            "count": 6,
            "id": 4,
            "name": "Closed"
        },
        "5": {
            "color": "#89BAB4",
            "count": 2,
            "id": 5,
            "name": "Needs Info"
        },
        "6": {
            "color": "#CC0000",
            "count": 8,
            "id": 6,
            "name": "Rejected"
        },
        "7": {
            "color": "#666666",
            "count": 2,
            "id": 7,
            "name": "Postponed"
        }
    },
    "issues_per_type": {
        "1": {
            "color": "#89BAB4",
            "count": 13,
            "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,
                0,
                0,
                0,
                16
            ],
            "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,
                28
            ]
        },
        "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,
                13
            ],
            "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,
                0,
                10
            ],
```

```
                "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,
                    0,
                    0,
                    5
                ],
                "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,
                0,
                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,
                7
            ],
            "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,
                7
            ],
            "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,
                    0,
                    0,
                    0,
                    2
                ],
                "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,
                    7
                ],
                "id": 5,
                "name": "Critical"
            }
        },
        "by_status": {}
    },
    "opened_issues": 12,
    "total_issues": 28
}
```


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": [
            "modi",
            "eligendi",
            "perferendis",
            "voluptates"
        ],
        "to_tag": "modi"
    }' \
-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/3/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": "6:1buyed:Z64dQb1M495E1UczL0atESFKc8E"
    }' \
-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": "6:1buyee:MYrUHif5smCEniHOJ731cId9q2w"
    }' \
-s http://localhost:8000/api/v1/projects/2/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": "6:1buyed:Z64dQb1M495E1UczL0atESFKc8E"
    }' \
-s http://localhost:8000/api/v1/projects/1/transfer_reject
```

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

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
- **email** (required): user email

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "email": "test-user@test.com",
        "project": 1,
        "role": 3
    }' \
-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**
  - **email**
- **invitation_extra_text**: string

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-d '{
        "bulk_memberships": [
            {
                "email": "test@test.com",
                "role_id": 3
            },
            {
                "email": "john@doe.com",
                "role_id": 4
            }
        ],
        "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. Milestones

12.1. 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.

12.2. 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

12.3. 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

12.4. 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

12.5. 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

12.6. 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

12.7. 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

12.8. 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

12.9. 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

13. Epics

13.1. 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
```

13.2. 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

13.3. 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

13.4. 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=106\&project=2
```

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

13.5. 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/10
```

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

13.6. 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/10
```

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

13.7. 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

13.8. 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

13.9. 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/10/related_userstories
```

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

13.10. 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": 10,
        "user_story": 1
    }' \
-s http://localhost:8000/api/v1/epics/10/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

13.11. 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/10/related_userstories/37
```

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

13.12. 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/10/related_userstories/37
```

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

13.13. 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/10/related_userstories/37
```

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

13.14. 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": 2
    }' \
-s http://localhost:8000/api/v1/epics/10/related_userstories/bulk_create
```

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

13.15. 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

13.16. 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

13.17. 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

13.18. 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/10/watch
```

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

13.19. 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/10/unwatch
```

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

13.20. 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/10/watchers
```

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

13.21. 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=642\&project=1
```

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

13.22. 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 object_id=10 \
-F project=2 \
-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

13.23. 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/642
```

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

13.24. 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/642
```

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

13.25. 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/642
```

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

14. Epic status

14.1. 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
```

14.2. 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

14.3. 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

14.4. 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

14.5. 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

14.6. 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

15. Epic custom attribute

15.1. 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
```

15.2. 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

15.3. 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/8
```

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

15.4. 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/8
```

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

15.5. 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/8
```

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 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": [
            [
                8,
                10
            ],
            [
                6,
                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

16. Epic custom attributes values

16.1. 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/10
```

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

16.2. 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": {
            "8": "240 min"
        },
        "version": 1
    }' \
-s http://localhost:8000/api/v1/epics/custom-attributes-values/10
```

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

17. User stories

17.1. 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)
- **watchers**: watching user id
- **assigned_to**: assigned to user id
- **status__is_closed**: (true|false)

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

17.2. 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

17.3. 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

17.4. 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

17.5. 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

17.6. 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

17.7. 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

17.8. 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

17.9. 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

17.10. 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

17.11. 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

17.12. 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

17.13. 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

17.14. 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

17.15. 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

17.16. 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

17.17. 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

17.18. 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": "GitLab",
    "id": 3,
    "username": "gitlab-13aa142f88564bd080fd984fd8fc9a01"
}
```

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

17.19. 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

17.20. 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 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

17.21. 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

17.22. 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

17.23. 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/339
```

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

18. User story status

18.1. 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
```

18.2. 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

18.3. 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

18.4. 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

18.5. 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

18.6. 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

19. Points

19.1. 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
```

19.2. 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

19.3. 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

19.4. 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

19.5. 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

19.6. 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

20. User story custom attribute

20.1. 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
```

20.2. 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

20.3. 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

20.4. 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

20.5. 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

20.6. 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

21. User story custom attributes values

21.1. 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

21.2. 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": {
            "5": "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

22. Tasks

22.1. 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
- **user_story**: user story id
- **milestone**: milestone id
- **watchers**: watching user id
- **assigned_to**: assigned to user id
- **status__is_closed**: (true|false)

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

22.2. 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

22.3. 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

22.4. 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": "Catalina Roman",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2016-10-14T09:03:29.043Z",
    "description": "Perspiciatis ipsum repellat quia quidem officia, totam accusamus
laboriosam, quas expedita quos dolore adipisci animi harum hic?",
    "description_html": "<p>Perspiciatis ipsum repellat quia quidem officia, totam
accusamus laboriosam, quas expedita quos dolore adipisci animi harum hic?</p>",
    "external_reference": null,
    "finished_date": "2016-08-29T23:17:07.508Z",
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": false,
    "is_watcher": true,
    "milestone": 1,
    "milestone_slug": "sprint-2016-8-20",
    "modified_date": "2016-10-14T09:20:35.510Z",
```

```
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 3,
            "subject": "Add tests for bulk operations"
        },
        "previous": null
    },
    "owner": 12,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Pilar Herrera",
        "gravatar_id": "74cb769a5e64d445b8550789e1553502",
        "id": 12,
        "is_active": true,
        "photo": null,
        "username": "user6"
    },
    "project": 1,
    "ref": 2,
    "status": 4,
    "status_extra_info": {
        "color": "#669900",
        "is_closed": true,
        "name": "Cerrada"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "iste",
            null
        ],
        [
            "obcaecati",
            null
        ],
        [
            "vel",
            null
        ],
        [
            "totam",
            null
        ],
        [
            "modi",
            null
        ],
```

```
        [
            "eos",
            null
        ],
        [
            "deleniti",
            null
        ]
    ],
    "taskboard_order": 1476435809043,
    "total_voters": 6,
    "total_watchers": 1,
    "us_order": 1476435809043,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": null,
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": [
        6
    ]
}
```


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

22.5. 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

22.6. 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

22.7. 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

22.8. 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

22.9. 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

22.10. 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

22.11. 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

22.12. 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

22.13. 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

22.14. 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

22.15. 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

22.16. 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 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

22.17. 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/343
```

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

22.18. 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/343
```

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

22.19. 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/343
```

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

23. Task status

23.1. 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
```

23.2. 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

23.3. 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

23.4. 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

23.5. 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

23.6. 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

24. Task custom attribute

24.1. 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
```

24.2. 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

24.3. 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/5
```

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

24.4. 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/5
```

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

24.5. 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/5
```

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 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": [
            [
                5,
                10
            ],
            [
                1,
                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

25. Task custom attributes values

25.1. 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/5
```

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

25.2. 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/5
```

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

26. Issues

26.1. 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
- **watchers**: watching user id
- **status__is_closed**: (true|false)

```
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
```

26.2. 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

26.3. 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

26.4. 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=119\&project=1
```

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

26.5. 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

26.6. 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/23
```

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

26.7. 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

26.8. 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

26.9. 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

26.10. 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

26.11. 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

26.12. 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

26.13. 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

26.14. 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=620\&project=1
```

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

26.15. 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 object_id=23 \
-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

26.16. 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/620
```

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

26.17. 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/620
```

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

26.18. 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/620
```

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

27. Issue status

27.1. 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
```

27.2. 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

27.3. 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

27.4. 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

27.5. 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

27.6. 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

28. Issue types

28.1. 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
```

28.2. 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

28.3. 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

28.4. 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

28.5. 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

28.6. 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

29. Priorities

29.1. 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
```

29.2. 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

29.3. 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

29.4. 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

29.5. 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

29.6. 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

30. Severities

30.1. 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
```

30.2. 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

30.3. 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

30.4. 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

30.5. 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

30.6. 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

31. Issue custom attribute

31.1. 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
```

31.2. 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

31.3. 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

31.4. 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

31.5. 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

31.6. 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
            ],
            [
                4,
                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

32. Issue custom attributes values

32.1. 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/23
```

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

32.2. 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/23
```

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

33. Wiki pages

33.1. 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
```

33.2. 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

33.3. 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/6
```

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

33.4. 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

33.5. 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/6
```

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

33.6. 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/6
```

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

33.7. 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/6/watch
```

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

33.8. 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/6/unwatch
```

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

33.9. 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/6/watchers
```

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

33.10. 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=6\&project=1
```

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

33.11. 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/6/4/8/8/de60dfc7165c22f52d8c0fa427027ad0c6e07d64e9739ea5ff87d1aec1b5/sample_
attachment_1.txt",
    "created_date": "2016-10-14T09:05:18.959Z",
    "description": "Updated description",
    "id": 623,
    "is_deprecated": true,
    "modified_date": "2016-10-14T09:20:31.871Z",
    "name": "sample_attachment_1.txt",
    "object_id": 7,
    "order": 1,
    "owner": 14,
    "project": 2,
    "sha1": "e9913a16efd01bf9b44d712cf601802fdc8869ea",
    "size": 289,
    "thumbnail_card_url": null,
    "url":
"http://localhost:8000/media/attachments/6/4/8/8/de60dfc7165c22f52d8c0fa427027ad0c6e07d64
e9739ea5ff87d1aec1b5/sample_attachment_1.txt"
}
```


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

33.12. 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/623
```

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

33.13. 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/623
```

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

33.14. 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/623
```

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

34. Wiki links

34.1. 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
```

34.2. 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

34.3. 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

34.4. 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

34.5. 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

35. History

35.1. 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

35.2. 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

35.3. 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

35.4. 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

35.5. 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

36. Users

36.1. 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
```

36.2. 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

36.3. 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

36.4. 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

36.5. 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
```

36.6. 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
```

36.7. 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
```

36.8. 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

36.9. 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

36.10. 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 HTTP response is a 200 OK and the response body is a list of JSON contact detail object

36.11. 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:1buyee:wxlHwVGDiHy6K2RlpVb0P3S_b6c"
    }' \
-s http://localhost:8000/api/v1/users/cancel
```

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

36.12. 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

36.13. 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

36.14. 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

36.15. 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

36.16. 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" \
-H "Authorization: Bearer ${AUTH_TOKEN}" \
-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": "\u00a1Correo enviado con \u00e9xito!"
}
```

36.17. 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

37. Notify policies

37.1. 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

37.2. 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/4
```

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

37.3. 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/4
```

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

38. Feedback

38.1. 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

39. Export/Import

39.1. 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.

39.2. 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

40. Webhooks

40.1. 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
```

40.2. 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

40.3. 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

40.4. 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

40.5. 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

40.6. 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.

40.7. 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
```

40.8. 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

40.9. 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.

41. Timelines

41.1. 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

41.2. 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

41.3. 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

42. Locales

42.1. 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

43. Stats

43.1. 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

43.2. 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

44. Objects Summary

44.1. Attachment


```
{
    "attached_file":
"attachments/5/d/5/1/0726cadbfc918e9f372ba83cdd817120b929b0d95b07df0b4ad9c1551e86/sample_
attachment_3.txt",
    "created_date": "2016-10-14T09:04:42.760Z",
    "description": "nam delectus adipisci ratione animi numquam",
    "id": 415,
    "is_deprecated": false,
    "modified_date": "2016-10-14T09:04:42.760Z",
    "name": "sample_attachment_3.txt",
    "object_id": 81,
    "order": 1,
    "owner": 6,
    "project": 2,
    "sha1": "da2631d805f12a1b533738a0912e9b9c2261dbef",
    "size": 1178,
    "thumbnail_card_url": null,
    "url":
"http://localhost:8000/media/attachments/5/d/5/1/0726cadbfc918e9f372ba83cdd817120b929b0d9
5b07df0b4ad9c1551e86/sample_attachment_3.txt"
}
```

```
{
    "description": "description paragraph",
    "icon_url": null,
    "id": "00000000-0000-0000-0000-000000000000",
    "name": "example application",
    "web": "http://example.com"
}
```

44.2. 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": "77b2c8d0-91ef-11e6-9c17-68f72800aadd",
    "id": 1,
    "next_url": "http://example.com?auth_code=77b2c8d0-91ef-11e6-9c17-68f72800aadd",
    "user": 6
}
```

44.3. Authorization code object

```
{
    "auth_code": "77b2c8d0-91ef-11e6-9c17-68f72800aadd",
    "next_url": "http://example.com?auth_code=77b2c8d0-91ef-11e6-9c17-68f72800aadd",
    "state": "random-state"
}
```

44.4. Cyphered token object

```
{
    "token": "00000000-0000-0000-0000-000000000001"
}
```

44.5. User detail


```
{
    "big_photo": null,
    "bio": "",
    "color": "#4B0082",
    "email": "user6532909695705815086@taigaio.demo",
    "full_name": "Silvia Soto",
    "full_name_display": "Silvia Soto",
    "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
    "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,
    "projects_with_me": [
        {
            "id": 9,
            "name": "Beta project",
            "slug": "user6532909695705815086-beta-project"
        },
        {
            "id": 10,
            "name": "Beta project",
            "slug": "user6532909695705815086-beta-project-1"
        },
        {
            "id": 1,
            "name": "Beta project patch",
            "slug": "project-0"
        },
        {
            "id": 8,
            "name": "Project Example",
            "slug": "project-example"
        },
        {
            "id": 2,
            "name": "Project Example 1",
            "slug": "project-1"
        },
        {
            "id": 3,
```

```
            "name": "Project Example 2",
            "slug": "project-2"
        },
        {
            "id": 4,
            "name": "Project Example 3",
            "slug": "project-3"
        },
        {
            "id": 5,
            "name": "Project Example 4",
            "slug": "project-4"
        },
        {
            "id": 6,
            "name": "Project Example 5",
            "slug": "project-5"
        },
        {
            "id": 7,
            "name": "Project Example 6",
            "slug": "project-6"
        }
    ],
    "roles": [
        "Back",
        "Product Owner",
        "Stakeholder",
        "UX"
    ],
    "theme": "",
    "timezone": "",
    "total_private_projects": 2,
    "total_public_projects": 3,
    "username": "patchedusername"
}
```


44.6. User contact detail

```
{
    "big_photo": null,
    "bio": "",
    "color": "#D70A53",
    "full_name": "Alba Leon",
    "full_name_display": "Alba Leon",
    "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
    "id": 8,
    "is_active": true,
    "lang": "",
    "photo": null,
    "projects_with_me": [],
    "roles": [
        "Design",
        "Front",
        "Stakeholder"
    ],
    "theme": "",
    "timezone": "",
    "username": "user2"
}
```


44.7. User authentication-detail

```
{
    "auth_token":
"eyJ1c2VyX2F1dGhlbnRpY2F0aW9uX2lkIjoxNn0:1buyeg:B4yvGHG8N_yIskbIoR8psYIvtMY",
    "big_photo": null,
    "bio": "",
    "color": "#a78b21",
    "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,
    "projects_with_me": [],
    "roles": [
        "Front"
    ],
    "theme": "",
    "timezone": "",
    "total_private_projects": 0,
    "total_public_projects": 0,
    "username": "test-username"
}
```


44.8. User stats detail


```
{
    "roles": [
        "Stakeholder",
        "Back",
        "UX",
        "Product Owner"
    ],
    "total_num_closed_userstories": 0,
    "total_num_contacts": 11,
    "total_num_projects": 10
}
```

44.9. Search results detail


```
{
    "count": 94,
    "epics": [
        {
            "assigned_to": 12,
            "id": 3,
            "ref": 122,
            "status": 2,
            "subject": "Lighttpd x-sendfile support"
        },
        {
            "assigned_to": null,
            "id": 5,
            "ref": 124,
            "status": 2,
            "subject": "Add setting to allow regular users to create folders at the root
level."
        },
        {
            "assigned_to": 11,
            "id": 1,
            "ref": 120,
            "status": 2,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 10,
            "id": 6,
            "ref": 125,
            "status": null,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": null,
            "id": 7,
            "ref": 126,
            "status": 4,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "assigned_to": 9,
            "id": 2,
            "ref": 121,
            "status": 4,
```

```
            "subject": "Implement the form"
        },
        {
            "assigned_to": null,
            "id": 4,
            "ref": 123,
            "status": null,
            "subject": "Feature/improved image admin"
        }
    ],
    "issues": [
        {
            "assigned_to": 5,
            "id": 8,
            "ref": 104,
            "status": 6,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 6,
            "id": 1,
            "ref": 97,
            "status": 6,
            "subject": "Fixing templates for Django 1.6."
        },
        {
            "assigned_to": 9,
            "id": 22,
            "ref": 118,
            "status": 4,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 13,
            "id": 23,
            "ref": 119,
            "status": 5,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "assigned_to": 15,
            "id": 15,
            "ref": 111,
            "status": 4,
            "subject": "Feature/improved image admin"
        },
        {
            "assigned_to": null,
```

```
            "id": 20,
            "ref": 116,
            "status": 6,
            "subject": "Lighttpd x-sendfile support"
        },
        {
            "assigned_to": 7,
            "id": 19,
            "ref": 115,
            "status": 1,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": 8,
            "id": 12,
            "ref": 108,
            "status": 7,
            "subject": "Experimental: modular file types"
        },
        {
            "assigned_to": 8,
            "id": 21,
            "ref": 117,
            "status": 6,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": 7,
            "id": 7,
            "ref": 103,
            "status": 4,
            "subject": "Implement the form"
        },
        {
            "assigned_to": 8,
            "id": 11,
            "ref": 107,
            "status": 4,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 11,
            "id": 2,
            "ref": 98,
            "status": 2,
            "subject": "Implement the form"
        },
        {
```

```
            "assigned_to": null,
            "id": 17,
            "ref": 113,
            "status": 5,
            "subject": "Fixing templates for Django 1.6."
        },
        {
            "assigned_to": 13,
            "id": 3,
            "ref": 99,
            "status": 6,
            "subject": "Patching subject"
        },
        {
            "assigned_to": 13,
            "id": 9,
            "ref": 105,
            "status": 6,
            "subject": "Create the html template"
        },
        {
            "assigned_to": 15,
            "id": 13,
            "ref": 109,
            "status": 3,
            "subject": "Lighttpd x-sendfile support"
        }
    ],
    "tasks": [
        {
            "assigned_to": 7,
            "id": 6,
            "ref": 8,
            "status": 5,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "assigned_to": 7,
            "id": 4,
            "ref": 6,
            "status": 2,
            "subject": "Fixing templates for Django 1.6."
        },
        {
            "assigned_to": 10,
            "id": 54,
            "ref": 71,
            "status": 4,
```

```
            "subject": "Support for bulk actions"
        },
        {
            "assigned_to": 14,
            "id": 46,
            "ref": 61,
            "status": 2,
            "subject": "Experimental: modular file types"
        },
        {
            "assigned_to": 7,
            "id": 52,
            "ref": 69,
            "status": 3,
            "subject": "Lighttpd x-sendfile support"
        },
        {
            "assigned_to": 14,
            "id": 3,
            "ref": 4,
            "status": 2,
            "subject": "Create the html template"
        },
        {
            "assigned_to": 9,
            "id": 21,
            "ref": 27,
            "status": 3,
            "subject": "get_actions() does not check for 'delete_selected' in actions"
        },
        {
            "assigned_to": 14,
            "id": 25,
            "ref": 32,
            "status": 1,
            "subject": "Lighttpd support"
        },
        {
            "assigned_to": 5,
            "id": 40,
            "ref": 53,
            "status": 5,
            "subject": "Exception is thrown if trying to add a folder with existing name"
        },
        {
            "assigned_to": 7,
            "id": 30,
            "ref": 40,
```

```
            "status": 1,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 5,
            "id": 12,
            "ref": 16,
            "status": 5,
            "subject": "Create the user model"
        },
        {
            "assigned_to": 15,
            "id": 33,
            "ref": 44,
            "status": 2,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 15,
            "id": 28,
            "ref": 37,
            "status": 5,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": 15,
            "id": 48,
            "ref": 63,
            "status": 5,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 5,
            "id": 34,
            "ref": 45,
            "status": 5,
            "subject": "Lighttpd x-sendfile support"
        },
        {
            "assigned_to": 15,
            "id": 32,
            "ref": 42,
            "status": 3,
            "subject": "Create the user model"
        },
        {
            "assigned_to": 5,
            "id": 13,
```

```
            "ref": 17,
            "status": 5,
            "subject": "Feature/improved image admin"
        },
        {
            "assigned_to": 12,
            "id": 15,
            "ref": 20,
            "status": 1,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "assigned_to": 9,
            "id": 41,
            "ref": 55,
            "status": 3,
            "subject": "Add setting to allow regular users to create folders at the root
level."
        },
        {
            "assigned_to": 10,
            "id": 44,
            "ref": 58,
            "status": 4,
            "subject": "get_actions() does not check for 'delete_selected' in actions"
        },
        {
            "assigned_to": 9,
            "id": 58,
            "ref": 77,
            "status": 2,
            "subject": "Support for bulk actions"
        },
        {
            "assigned_to": 7,
            "id": 8,
            "ref": 10,
            "status": 5,
            "subject": "Implement the form"
        },
        {
            "assigned_to": 9,
            "id": 7,
            "ref": 9,
            "status": 4,
            "subject": "Fixing templates for Django 1.6."
        },
        {
```

```
            "assigned_to": 8,
            "id": 9,
            "ref": 12,
            "status": 4,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 5,
            "id": 60,
            "ref": 79,
            "status": 3,
            "subject": "Experimental: modular file types"
        },
        {
            "assigned_to": 15,
            "id": 16,
            "ref": 21,
            "status": 5,
            "subject": "Add setting to allow regular users to create folders at the root
level."
        },
        {
            "assigned_to": 15,
            "id": 47,
            "ref": 62,
            "status": 1,
            "subject": "Add setting to allow regular users to create folders at the root
level."
        },
        {
            "assigned_to": 15,
            "id": 29,
            "ref": 38,
            "status": 3,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 11,
            "id": 55,
            "ref": 73,
            "status": 2,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 6,
            "id": 42,
            "ref": 56,
            "status": 4,
```

```
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "assigned_to": 15,
            "id": 59,
            "ref": 78,
            "status": 2,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 5,
            "id": 5,
            "ref": 7,
            "status": 3,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "assigned_to": 7,
            "id": 27,
            "ref": 35,
            "status": 2,
            "subject": "Create the user model"
        },
        {
            "assigned_to": 12,
            "id": 26,
            "ref": 33,
            "status": 4,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 6,
            "id": 18,
            "ref": 24,
            "status": 4,
            "subject": "get_actions() does not check for 'delete_selected' in actions"
        },
        {
            "assigned_to": 12,
            "id": 51,
            "ref": 67,
            "status": 3,
            "subject": "Create the html template"
        },
        {
            "assigned_to": 6,
            "id": 23,
            "ref": 30,
```

```
            "status": 5,
            "subject": "Implement the form"
        },
        {
            "assigned_to": 13,
            "id": 37,
            "ref": 49,
            "status": 5,
            "subject": "Feature/improved image admin"
        },
        {
            "assigned_to": 13,
            "id": 39,
            "ref": 52,
            "status": 1,
            "subject": "Add setting to allow regular users to create folders at the root
level."
        },
        {
            "assigned_to": 7,
            "id": 35,
            "ref": 46,
            "status": 2,
            "subject": "Create testsuite with matrix builds"
        },
        {
            "assigned_to": 15,
            "id": 1,
            "ref": 2,
            "status": 4,
            "subject": "Patching subject"
        },
        {
            "assigned_to": 14,
            "id": 20,
            "ref": 26,
            "status": 5,
            "subject": "Create the html template"
        },
        {
            "assigned_to": 10,
            "id": 45,
            "ref": 59,
            "status": 1,
            "subject": "Add tests for bulk operations"
        },
        {
            "assigned_to": 15,
```

```
            "id": 61,
            "ref": 82,
            "status": 4,
            "subject": "Feature/improved image admin"
        },
        {
            "assigned_to": 7,
            "id": 31,
            "ref": 41,
            "status": 1,
            "subject": "Lighttpd x-sendfile support"
        }
    ],
    "userstories": [
        {
            "id": 2,
            "milestone_name": "Sprint 2016-8-20",
            "milestone_slug": "sprint-2016-8-20",
            "ref": 5,
            "status": 3,
            "subject": "get_actions() does not check for 'delete_selected' in actions",
            "total_points": 44.5
        },
        {
            "id": 16,
            "milestone_name": "Sprint 2016-9-19",
            "milestone_slug": "sprint-2016-9-19",
            "ref": 65,
            "status": 1,
            "subject": "Migrate to Python 3 and milk a beautiful cow",
            "total_points": 48.5
        },
        {
            "id": 5,
            "milestone_name": "Sprint 2016-8-20",
            "milestone_slug": "sprint-2016-8-20",
            "ref": 19,
            "status": 3,
            "subject": "Fixing templates for Django 1.6.",
            "total_points": 21.5
        },
        {
            "id": 28,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 90,
            "status": 2,
            "subject": "Added file copying and processing of images (resizing)",
```

```
            "total_points": 53.0
        },
        {
            "id": 29,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 91,
            "status": 2,
            "subject": "Lighttpd support",
            "total_points": 13.5
        },
        {
            "id": 19,
            "milestone_name": "Sprint 2016-10-4",
            "milestone_slug": "sprint-2016-10-4",
            "ref": 76,
            "status": 3,
            "subject": "Implement the form",
            "total_points": 43.0
        },
        {
            "id": 8,
            "milestone_name": "Sprint 2016-9-4",
            "milestone_slug": "sprint-2016-9-4",
            "ref": 34,
            "status": 1,
            "subject": "Support for bulk actions",
            "total_points": 25.5
        },
        {
            "id": 11,
            "milestone_name": "Sprint 2016-9-19",
            "milestone_slug": "sprint-2016-9-19",
            "ref": 43,
            "status": 1,
            "subject": "Lighttpd x-sendfile support",
            "total_points": 34.5
        },
        {
            "id": 15,
            "milestone_name": "Sprint 2016-9-19",
            "milestone_slug": "sprint-2016-9-19",
            "ref": 60,
            "status": 2,
            "subject": "Experimental: modular file types",
            "total_points": 71.0
        },
        {
```

```
            "id": 6,
            "milestone_name": "Sprint 2016-8-20",
            "milestone_slug": "sprint-2016-8-20",
            "ref": 23,
            "status": 3,
            "subject": "Create the user model",
            "total_points": 11.0
        },
        {
            "id": 31,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 93,
            "status": 2,
            "subject": "Add setting to allow regular users to create folders at the root
level.",
            "total_points": 62.0
        },
        {
            "id": 30,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 92,
            "status": 1,
            "subject": "Lighttpd support",
            "total_points": 13.5
        },
        {
            "id": 21,
            "milestone_name": "Sprint 2016-10-4",
            "milestone_slug": "sprint-2016-10-4",
            "ref": 81,
            "status": 4,
            "subject": "Fixing templates for Django 1.6.",
            "total_points": 34.0
        },
        {
            "id": 18,
            "milestone_name": "Sprint 2016-10-4",
            "milestone_slug": "sprint-2016-10-4",
            "ref": 72,
            "status": 3,
            "subject": "Feature/improved image admin",
            "total_points": 9.0
        },
        {
            "id": 12,
            "milestone_name": "Sprint 2016-9-19",
```

```
            "milestone_slug": "sprint-2016-9-19",
            "ref": 48,
            "status": 2,
            "subject": "get_actions() does not check for 'delete_selected' in actions",
            "total_points": 20.0
        },
        {
            "id": 24,
            "milestone_name": "Sprint 2016-10-4",
            "milestone_slug": "sprint-2016-10-4",
            "ref": 85,
            "status": 4,
            "subject": "Add tests for bulk operations",
            "total_points": 63.0
        },
        {
            "id": 13,
            "milestone_name": "Sprint 2016-9-19",
            "milestone_slug": "sprint-2016-9-19",
            "ref": 50,
            "status": 2,
            "subject": "Create the user model",
            "total_points": 28.5
        },
        {
            "id": 7,
            "milestone_name": "Sprint 2016-9-4",
            "milestone_slug": "sprint-2016-9-4",
            "ref": 28,
            "status": 2,
            "subject": "Feature/improved image admin",
            "total_points": 23.0
        },
        {
            "id": 25,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 87,
            "status": 1,
            "subject": "Support for bulk actions",
            "total_points": 86.0
        },
        {
            "id": 26,
            "milestone_name": null,
            "milestone_slug": null,
            "ref": 88,
            "status": 2,
```

```
            "subject": "Create the user model",
            "total_points": 50.0
        }
    ],
    "wikipages": [
        {
            "id": 3,
            "slug": "culpa-quis"
        },
        {
            "id": 2,
            "slug": "labore"
        },
        {
            "id": 5,
            "slug": "ex-quo-illum"
        },
        {
            "id": 1,
            "slug": "home"
        },
        {
            "id": 6,
            "slug": "corporis"
        },
        {
            "id": 4,
            "slug": "pariatur-perspiciatis-sit"
        }
    ]
}
```


44.10. User storage data


```
{
    "created_date": "2016-10-14T09:20:24.772Z",
    "key": "favorite-forest",
    "modified_date": "2016-10-14T09:20:24.820Z",
    "value": "Russian Taiga"
}
```


45. 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_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": "2016-10-14T09:20:37.322Z",
    "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": ""
}
```


45.1. Project list entry


```
{
    "anon_permissions": [],
    "blocked_code": null,
    "created_date": "2016-10-14T09:20:26.747Z",
    "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_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": "2016-10-14T09:20:27.436Z",
    "my_permissions": [
        "comment_us",
        "delete_issue",
        "delete_task",
        "delete_epic",
        "delete_project",
        "modify_us",
        "delete_wiki_link",
        "view_project",
        "modify_issue",
        "view_wiki_links",
        "add_wiki_page",
        "view_tasks",
        "modify_project",
        "comment_issue",
        "remove_member",
        "add_issue",
        "view_wiki_pages",
        "delete_milestone",
```

```
        "add_wiki_link",
        "modify_task",
        "comment_epic",
        "modify_milestone",
        "add_task",
        "add_member",
        "delete_us",
        "add_milestone",
        "view_epics",
        "add_epic",
        "comment_task",
        "modify_epic",
        "admin_project_values",
        "view_issues",
        "delete_wiki_page",
        "comment_wiki_page",
        "admin_roles",
        "add_us",
        "view_milestones",
        "view_us",
        "modify_wiki_link",
        "modify_wiki_page"
    ],
    "name": "Beta project",
    "notify_level": 1,
    "owner": {
        "big_photo": null,
        "full_name_display": "Silvia Soto",
        "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user6532909695705815086"
    },
    "public_permissions": [],
    "slug": "user6532909695705815086-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": "2016-10-14T09:20:27.499Z",
    "videoconferences": null,
    "videoconferences_extra_data": null
}
```


45.2. Project detail


```
{
    "anon_permissions": [],
    "blocked_code": null,
    "created_date": "2016-10-14T09:03:27.385Z",
    "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": "2016-10-14T09:03:28.404438+00:00",
            "description": "blanditiis tempore asperiores odit dolor voluptatibus
officiis perferendis sunt labore quisquam",
            "id": 1,
            "modified_date": "2016-10-14T09:03:28.404469+00:00",
            "name": "dolores sed",
            "order": 1,
            "project_id": 1,
            "type": "multiline"
        },
        {
            "created_date": "2016-10-14T09:03:28.40782+00:00",
            "description": "nobis quo repellendus dolores",
            "id": 2,
            "modified_date": "2016-10-14T09:03:28.40785+00:00",
            "name": "numquam iusto",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
```

```
            "created_date": "2016-10-14T09:03:28.410656+00:00",
            "description": "aperiam nesciunt exercitationem nam hic et deserunt",
            "id": 3,
            "modified_date": "2016-10-14T09:03:28.410687+00:00",
            "name": "et delectus",
            "order": 1,
            "project_id": 1,
            "type": "date"
        },
        {
            "created_date": "2016-10-14T09:03:28.41357+00:00",
            "description": "et officia modi laboriosam reiciendis placeat quisquam
nostrum eum",
            "id": 4,
            "modified_date": "2016-10-14T09:03:28.4136+00:00",
            "name": "adipisci ab rem",
            "order": 1,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2016-10-14T09:03:28.418254+00:00",
            "description": "cum iste corrupti repellat eveniet nam voluptates architecto
perspiciatis",
            "id": 5,
            "modified_date": "2016-10-14T09:03:28.418283+00:00",
            "name": "nihil",
            "order": 1,
            "project_id": 1,
            "type": "multiline"
        },
        {
            "created_date": "2016-10-14T09:19:59.35799+00:00",
            "description": "Duration in minutes",
            "id": 26,
            "modified_date": "2016-10-14T09:19:59.367557+00:00",
            "name": "Duration 2",
            "order": 8,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2016-10-14T09:19:59.407458+00:00",
            "description": "",
            "id": 27,
            "modified_date": "2016-10-14T09:19:59.41694+00:00",
            "name": "Duration 3",
            "order": 1476436799399,
```

```
            "project_id": 1,
            "type": "text"
        }
    ],
    "epic_statuses": [
        {
            "color": "#ff9900",
            "id": 3,
            "is_closed": false,
            "name": "En curso",
            "order": 3,
            "project_id": 1,
            "slug": "in-progress"
        },
        {
            "color": "#fcc000",
            "id": 4,
            "is_closed": false,
            "name": "Lista para testear",
            "order": 4,
            "project_id": 1,
            "slug": "ready-for-test"
        },
        {
            "color": "#669900",
            "id": 5,
            "is_closed": true,
            "name": "Hecha",
            "order": 5,
            "project_id": 1,
            "slug": "done"
        },
        {
            "color": "#ff8a84",
            "id": 2,
            "is_closed": false,
            "name": "Preparada",
            "order": 5,
            "project_id": 1,
            "slug": "ready"
        },
        {
            "color": "#AAAAAA",
            "id": 36,
            "is_closed": true,
            "name": "New status",
            "order": 8,
            "project_id": 1,
```

```
            "slug": "new-status"
        },
        {
            "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_epics_activated": false,
    "is_fan": false,
    "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": "2016-10-14T09:03:28.4859+00:00",
            "description": "corrupti id voluptas officiis voluptates iure",
            "id": 1,
            "modified_date": "2016-10-14T09:03:28.485932+00:00",
            "name": "cupiditate dolore",
            "order": 1,
            "project_id": 1,
            "type": "date"
        },
        {
            "created_date": "2016-10-14T09:03:28.489223+00:00",
            "description": "odio neque rerum eum recusandae facilis",
            "id": 2,
            "modified_date": "2016-10-14T09:03:28.489248+00:00",
```

```
            "name": "quo at",
            "order": 1,
            "project_id": 1,
            "type": "multiline"
        },
        {
            "created_date": "2016-10-14T09:03:28.498281+00:00",
            "description": "nisi cumque magni sint repellat quo sequi distinctio
architecto quis laborum suscipit",
            "id": 3,
            "modified_date": "2016-10-14T09:03:28.498308+00:00",
            "name": "libero aut",
            "order": 1,
            "project_id": 1,
            "type": "multiline"
        },
        {
            "created_date": "2016-10-14T09:03:28.502544+00:00",
            "description": "omnis maiores earum",
            "id": 4,
            "modified_date": "2016-10-14T09:03:28.502588+00:00",
            "name": "ipsa animi",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.505841+00:00",
            "description": "ad temporibus maiores",
            "id": 5,
            "modified_date": "2016-10-14T09:03:28.505878+00:00",
            "name": "accusamus quasi",
            "order": 1,
            "project_id": 1,
            "type": "url"
        }
    ],
    "issue_statuses": [
        {
            "color": "#88A65E",
            "id": 3,
            "is_closed": true,
            "name": "Lista para testear",
            "order": 3,
            "project_id": 1,
            "slug": "ready-for-test"
        },
        {
```

```
            "color": "#BFB35A",
            "id": 4,
            "is_closed": true,
            "name": "Cerrada",
            "order": 4,
            "project_id": 1,
            "slug": "closed"
        },
        {
            "color": "#5E8C6A",
            "id": 2,
            "is_closed": false,
            "name": "En curso",
            "order": 5,
            "project_id": 1,
            "slug": "in-progress"
        },
        {
            "color": "#89BAB4",
            "id": 5,
            "is_closed": false,
            "name": "Necesita informaci\u00f3n",
            "order": 5,
            "project_id": 1,
            "slug": "needs-info"
        },
        {
            "color": "#CC0000",
            "id": 6,
            "is_closed": true,
            "name": "Rechazada",
            "order": 6,
            "project_id": 1,
            "slug": "rejected"
        },
        {
            "color": "#666666",
            "id": 7,
            "is_closed": false,
            "name": "Pospuesta",
            "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": "Pregunta",
            "order": 2,
            "project_id": 1
        },
        {
            "color": "#89a8ba",
            "id": 3,
            "name": "Mejora",
            "order": 3,
            "project_id": 1
        }
    ],
    "issues_csv_uuid": null,
```

```
    "logo_big_url":
"http://localhost:8000/media/project/4/f/7/b/91f142afda39e7bfd3c9ac3dccb0bf62b03b0192e447
6ff19b262ffcf7cc/test.png.300x300_q85_crop.png",
    "logo_small_url":
"http://localhost:8000/media/project/4/f/7/b/91f142afda39e7bfd3c9ac3dccb0bf62b03b0192e447
6ff19b262ffcf7cc/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": 2,
            "role_name": "Design",
            "username": "admin"
        },
        {
            "color": "#D70A53",
            "full_name": "Alba Leon",
            "full_name_display": "Alba Leon",
            "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
            "id": 8,
            "is_active": true,
            "photo": null,
            "role": 6,
            "role_name": "Stakeholder",
            "username": "user2"
        },
        {
            "color": "#002e33",
            "full_name": "Alvaro Molina",
            "full_name_display": "Alvaro Molina",
            "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
            "id": 13,
            "is_active": true,
            "photo": null,
            "role": 3,
            "role_name": "Front",
            "username": "user7"
        },
        {
            "color": "#FFCC00",
            "full_name": "Andrea Fernandez",
```

```
            "full_name_display": "Andrea Fernandez",
            "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
            "id": 14,
            "is_active": true,
            "photo": null,
            "role": 5,
            "role_name": "Product Owner",
            "username": "user8"
        },
        {
            "color": "#C0FF33",
            "full_name": "Catalina Roman",
            "full_name_display": "Catalina Roman",
            "gravatar_id": "69b60d39a450e863609ae3546b12b360",
            "id": 15,
            "is_active": true,
            "photo": null,
            "role": 5,
            "role_name": "Product Owner",
            "username": "user9"
        },
        {
            "color": "#FFF8E7",
            "full_name": "Esther Ferrer",
            "full_name_display": "Esther Ferrer",
            "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
            "id": 9,
            "is_active": true,
            "photo": null,
            "role": 4,
            "role_name": "Back",
            "username": "user3"
        },
        {
            "color": "#FFFF00",
            "full_name": "German Benitez",
            "full_name_display": "German Benitez",
            "gravatar_id": "c9ba9d485f9a9153ebf53758feb0980c",
            "id": 11,
            "is_active": true,
            "photo": null,
            "role": 4,
            "role_name": "Back",
            "username": "user5"
        },
        {
            "color": "#B6DA55",
            "full_name": "Marcos Ortiz",
```

```
            "full_name_display": "Marcos Ortiz",
            "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
            "id": 7,
            "is_active": true,
            "photo": null,
            "role": 3,
            "role_name": "Front",
            "username": "user1"
        },
        {
            "color": "#67CF00",
            "full_name": "Marta Carmona",
            "full_name_display": "Marta Carmona",
            "gravatar_id": "f31e0063c7cd6da19b6467bc48d2b14b",
            "id": 10,
            "is_active": true,
            "photo": null,
            "role": 6,
            "role_name": "Stakeholder",
            "username": "user4"
        },
        {
            "color": "#71A6D2",
            "full_name": "Pilar Herrera",
            "full_name_display": "Pilar Herrera",
            "gravatar_id": "74cb769a5e64d445b8550789e1553502",
            "id": 12,
            "is_active": true,
            "photo": null,
            "role": 1,
            "role_name": "UX",
            "username": "user6"
        },
        {
            "color": "#4B0082",
            "full_name": "Silvia Soto",
            "full_name_display": "Silvia Soto",
            "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
            "id": 6,
            "is_active": true,
            "photo": null,
            "role": 6,
            "role_name": "Stakeholder",
            "username": "user6532909695705815086"
        },
        {
            "color": "#a78b21",
            "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"
        }
    ],
    "milestones": [
        {
            "closed": false,
            "id": 1,
            "name": "Sprint 2016-8-20",
            "slug": "sprint-2016-8-20"
        },
        {
            "closed": false,
            "id": 2,
            "name": "Sprint 2016-9-4",
            "slug": "sprint-2016-9-4"
        },
        {
            "closed": false,
            "id": 3,
            "name": "Sprint 2016-9-19",
            "slug": "sprint-2016-9-19"
        },
        {
            "closed": false,
            "id": 4,
            "name": "Sprint 2016-10-4",
            "slug": "sprint-2016-10-4"
        }
    ],
    "modified_date": "2016-10-14T09:20:26.647Z",
    "my_permissions": [
        "comment_us",
        "delete_issue",
        "delete_task",
        "delete_epic",
        "delete_project",
        "modify_us",
        "delete_wiki_link",
        "view_project",
        "modify_issue",
        "view_wiki_links",
```

```
        "add_wiki_page",
        "view_tasks",
        "modify_project",
        "comment_issue",
        "remove_member",
        "add_issue",
        "view_wiki_pages",
        "delete_milestone",
        "add_wiki_link",
        "modify_task",
        "comment_epic",
        "modify_milestone",
        "add_task",
        "add_member",
        "delete_us",
        "add_milestone",
        "view_epics",
        "add_epic",
        "comment_task",
        "modify_epic",
        "admin_project_values",
        "view_issues",
        "delete_wiki_page",
        "comment_wiki_page",
        "admin_roles",
        "add_us",
        "view_milestones",
        "view_us",
        "modify_wiki_link",
        "modify_wiki_page"
    ],
    "name": "Beta project patch",
    "notify_level": 3,
    "owner": {
        "big_photo": null,
        "full_name_display": "Silvia Soto",
        "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user6532909695705815086"
    },
    "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": "Alta",
            "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": "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"
            ],
            "project_id": 1,
            "slug": "ux"
        },
        {
            "computable": true,
            "id": 2,
            "name": "Dise\u00f1ador",
            "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": "Importante",
            "order": 4,
            "project_id": 1
        },
        {
            "color": "#669933",
            "id": 2,
            "name": "Menor",
            "order": 5,
```

```
            "project_id": 1
        },
        {
            "color": "#CC0000",
            "id": 5,
            "name": "Cr\u00edtica",
            "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": "2016-10-14T09:03:28.451663+00:00",
            "description": "recusandae ipsa nisi non",
            "id": 1,
            "modified_date": "2016-10-14T09:03:28.451696+00:00",
            "name": "est",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.455104+00:00",
            "description": "nesciunt non adipisci debitis laudantium impedit similique
```

```
neque iste cumque hic minus",
            "id": 2,
            "modified_date": "2016-10-14T09:03:28.455131+00:00",
            "name": "odio officiis",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.458051+00:00",
            "description": "rerum ipsum at fugiat laborum neque sit vitae",
            "id": 3,
            "modified_date": "2016-10-14T09:03:28.458086+00:00",
            "name": "consequuntur deserunt velit",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.467002+00:00",
            "description": "deleniti sunt consectetur odit voluptate",
            "id": 4,
            "modified_date": "2016-10-14T09:03:28.467045+00:00",
            "name": "velit excepturi",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.476208+00:00",
            "description": "nulla laborum autem impedit deserunt delectus",
            "id": 5,
            "modified_date": "2016-10-14T09:03:28.476237+00:00",
            "name": "atque quam",
            "order": 1,
            "project_id": 1,
            "type": "url"
        }
    ],
    "task_statuses": [
        {
            "color": "#999999",
            "id": 1,
            "is_closed": false,
            "name": "Patch status name",
            "order": 10,
            "project_id": 1,
            "slug": "patch-status-name"
```

```
        },
        {
            "color": "#ff9900",
            "id": 2,
            "is_closed": false,
            "name": "En curso",
            "order": 5,
            "project_id": 1,
            "slug": "in-progress"
        },
        {
            "color": "#ffcc00",
            "id": 3,
            "is_closed": true,
            "name": "Lista para testear",
            "order": 3,
            "project_id": 1,
            "slug": "ready-for-test"
        },
        {
            "color": "#669900",
            "id": 4,
            "is_closed": true,
            "name": "Cerrada",
            "order": 4,
            "project_id": 1,
            "slug": "closed"
        },
        {
            "color": "#999999",
            "id": 5,
            "is_closed": false,
            "name": "Necesita informaci\u00f3n",
            "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": 42,
            "is_closed": false,
            "name": "New status name",
            "order": 10,
            "project_id": 1,
            "slug": "new-status-name"
        }
    ],
    "tasks_csv_uuid": null,
    "total_activity": 306,
    "total_activity_last_month": 306,
    "total_activity_last_week": 306,
    "total_activity_last_year": 306,
    "total_closed_milestones": 0,
    "total_fans": 6,
    "total_fans_last_month": 6,
    "total_fans_last_week": 6,
    "total_fans_last_year": 6,
    "total_memberships": 17,
    "total_milestones": 8,
    "total_story_points": 1098.0,
    "total_watchers": 15,
    "totals_updated_datetime": "2016-10-14T09:20:26.666Z",
    "transfer_token": "6:1buyed:Z64dQb1M495E1UczL0atESFKc8E",
    "us_statuses": [
        {
            "color": "#999999",
            "id": 1,
            "is_archived": false,
            "is_closed": false,
            "name": "Nueva",
            "order": 1,
            "project_id": 1,
            "slug": "new",
            "wip_limit": null
        },
        {
            "color": "#ff8a84",
            "id": 2,
            "is_archived": false,
            "is_closed": false,
            "name": "Preparada",
            "order": 2,
            "project_id": 1,
            "slug": "ready",
            "wip_limit": null
        },
        {
```

```
            "color": "#ff9900",
            "id": 3,
            "is_archived": false,
            "is_closed": false,
            "name": "En curso",
            "order": 3,
            "project_id": 1,
            "slug": "in-progress",
            "wip_limit": null
        },
        {
            "color": "#fcc000",
            "id": 4,
            "is_archived": false,
            "is_closed": false,
            "name": "Lista para testear",
            "order": 4,
            "project_id": 1,
            "slug": "ready-for-test",
            "wip_limit": null
        },
        {
            "color": "#669900",
            "id": 5,
            "is_archived": false,
            "is_closed": true,
            "name": "Hecha",
            "order": 5,
            "project_id": 1,
            "slug": "done",
            "wip_limit": null
        },
        {
            "color": "#5c3566",
            "id": 6,
            "is_archived": true,
            "is_closed": true,
            "name": "Archivada",
            "order": 6,
            "project_id": 1,
            "slug": "archived",
            "wip_limit": null
        }
    ],
    "userstories_csv_uuid": null,
    "userstory_custom_attributes": [
        {
            "created_date": "2016-10-14T09:03:28.430451+00:00",
```

```
            "description": "optio corrupti nostrum esse at accusamus porro nesciunt vero
sit maxime veritatis",
            "id": 4,
            "modified_date": "2016-10-14T09:03:28.430481+00:00",
            "name": "expedita illum reiciendis",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.441018+00:00",
            "description": "voluptates qui dicta deleniti reiciendis quo",
            "id": 5,
            "modified_date": "2016-10-14T09:03:28.441076+00:00",
            "name": "at officiis",
            "order": 1,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:03:28.427226+00:00",
            "description": "ullam ad in temporibus alias ducimus rerum odio mollitia",
            "id": 3,
            "modified_date": "2016-10-14T09:03:28.42725+00:00",
            "name": "ex",
            "order": 1,
            "project_id": 1,
            "type": "text"
        },
        {
            "created_date": "2016-10-14T09:03:28.424575+00:00",
            "description": "reiciendis minima quaerat veniam nihil illo expedita",
            "id": 2,
            "modified_date": "2016-10-14T09:03:28.4246+00:00",
            "name": "neque voluptatibus et",
            "order": 5,
            "project_id": 1,
            "type": "url"
        },
        {
            "created_date": "2016-10-14T09:20:05.105632+00:00",
            "description": "Duration in minutes",
            "id": 26,
            "modified_date": "2016-10-14T09:20:05.11497+00:00",
            "name": "Duration 2",
            "order": 8,
            "project_id": 1,
            "type": "text"
```

```
        },
        {
            "created_date": "2016-10-14T09:03:28.421275+00:00",
            "description": "accusamus sit corporis ipsum",
            "id": 1,
            "modified_date": "2016-10-14T09:20:05.054876+00:00",
            "name": "Duration 1",
            "order": 10,
            "project_id": 1,
            "type": "date"
        },
        {
            "created_date": "2016-10-14T09:20:05.151976+00:00",
            "description": "",
            "id": 27,
            "modified_date": "2016-10-14T09:20:05.159864+00:00",
            "name": "Duration 3",
            "order": 1476436805145,
            "project_id": 1,
            "type": "text"
        }
    ],
    "videoconferences": null,
    "videoconferences_extra_data": null
}
```


45.3. Project modules configuration

```
{
    "bitbucket": {
        "secret": "b62efda37fc34868ba212ca888bb2b5e",
        "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=b62efda37fc34868ba212ca888bb2b5e"
    },
    "github": {
        "secret": "36e9cafa298b417e8fbb61193285f7fa",
        "webhooks_url": "http://localhost:8000/api/v1/github-hook?project=1"
    },
    "gitlab": {
        "secret": "6cfba7639ff0461e8b09fce566da5384",
        "valid_origin_ips": [],
        "webhooks_url": "http://localhost:8000/api/v1/gitlab-
hook?project=1&key=6cfba7639ff0461e8b09fce566da5384"
    },
    "gogs": {
        "secret": "eef94a654c8e4dc19ffca590c6f7b80b",
        "webhooks_url": "http://localhost:8000/api/v1/gogs-hook?project=1"
    }
}
```


45.4. Project stats detail


```
{
    "assigned_points": 946.0,
    "assigned_points_per_role": {
        "1": 256.5,
        "2": 243.0,
        "3": 255.5,
        "4": 191.0
    },
    "closed_points": 134.0,
    "closed_points_per_role": {
        "1": 24.0,
        "2": 51.0,
        "3": 19.0,
        "4": 40.0
    },
```

```
    "defined_points": 1375.0,
    "defined_points_per_role": {
        "1": 373.0,
        "2": 295.5,
        "3": 342.5,
        "4": 364.0
    },
    "milestones": [
        {
            "client-increment": 0,
            "evolution": 1098.0,
            "name": "Sprint 2016-8-20",
            "optimal": 1098.0,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": 1056.0,
            "name": "Sprint 2016-9-4",
            "optimal": 960.75,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": 1056.0,
            "name": "Sprint 2016-9-19",
            "optimal": 823.5,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": 998.0,
            "name": "Sprint 2016-10-4",
            "optimal": 686.25,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": 964.0,
            "name": "Sprint futuro",
            "optimal": 549.0,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Sprint futuro",
            "optimal": 411.75,
```

```
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Sprint futuro",
            "optimal": 274.5,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Sprint futuro",
            "optimal": 137.25,
            "team-increment": 0
        },
        {
            "client-increment": 0,
            "evolution": null,
            "name": "Final de proyecto",
            "optimal": 0.0,
            "team-increment": 0
        }
    ],
    "name": "Project Example 0",
    "speed": 0,
    "total_milestones": 8,
    "total_points": 1098.0
}
```


45.5. Project issue stats detail


```
{
    "closed_issues": 16,
    "issues_per_assigned_to": {
        "0": {
            "color": "black",
            "count": 8,
            "id": 0,
            "name": "No asignado",
            "username": "No asignado"
        },
        "10": {
            "color": "#67CF00",
            "count": 1,
            "id": 10,
```

```
            "name": "Marta Carmona",
            "username": "user4"
        },
        "11": {
            "color": "#FFFF00",
            "count": 1,
            "id": 11,
            "name": "German Benitez",
            "username": "user5"
        },
        "12": {
            "color": "#71A6D2",
            "count": 1,
            "id": 12,
            "name": "Pilar Herrera",
            "username": "user6"
        },
        "13": {
            "color": "#002e33",
            "count": 3,
            "id": 13,
            "name": "Alvaro Molina",
            "username": "user7"
        },
        "14": {
            "color": "#FFCC00",
            "count": 2,
            "id": 14,
            "name": "Andrea Fernandez",
            "username": "user8"
        },
        "15": {
            "color": "#C0FF33",
            "count": 3,
            "id": 15,
            "name": "Catalina Roman",
            "username": "user9"
        },
        "5": {
            "color": "",
            "count": 1,
            "id": 5,
            "name": "Administrator",
            "username": "admin"
        },
        "6": {
            "color": "#4B0082",
            "count": 2,
```

```
            "id": 6,
            "name": "Silvia Soto",
            "username": "user6532909695705815086"
        },
        "7": {
            "color": "#B6DA55",
            "count": 2,
            "id": 7,
            "name": "Marcos Ortiz",
            "username": "user1"
        },
        "8": {
            "color": "#D70A53",
            "count": 3,
            "id": 8,
            "name": "Alba Leon",
            "username": "user2"
        },
        "9": {
            "color": "#FFF8E7",
            "count": 1,
            "id": 9,
            "name": "Esther Ferrer",
            "username": "user3"
        }
    },
    "issues_per_owner": {
        "10": {
            "color": "#67CF00",
            "count": 1,
            "id": 10,
            "name": "Marta Carmona",
            "username": "user4"
        },
        "11": {
            "color": "#FFFF00",
            "count": 1,
            "id": 11,
            "name": "German Benitez",
            "username": "user5"
        },
        "12": {
            "color": "#71A6D2",
            "count": 2,
            "id": 12,
            "name": "Pilar Herrera",
            "username": "user6"
        },
```

```
        "13": {
            "color": "#002e33",
            "count": 3,
            "id": 13,
            "name": "Alvaro Molina",
            "username": "user7"
        },
        "14": {
            "color": "#FFCC00",
            "count": 1,
            "id": 14,
            "name": "Andrea Fernandez",
            "username": "user8"
        },
        "15": {
            "color": "#C0FF33",
            "count": 2,
            "id": 15,
            "name": "Catalina Roman",
            "username": "user9"
        },
        "5": {
            "color": "",
            "count": 3,
            "id": 5,
            "name": "Administrator",
            "username": "admin"
        },
        "6": {
            "color": "#4B0082",
            "count": 9,
            "id": 6,
            "name": "Silvia Soto",
            "username": "user6532909695705815086"
        },
        "7": {
            "color": "#B6DA55",
            "count": 1,
            "id": 7,
            "name": "Marcos Ortiz",
            "username": "user1"
        },
        "8": {
            "color": "#D70A53",
            "count": 3,
            "id": 8,
            "name": "Alba Leon",
            "username": "user2"
```

```
        },
        "9": {
            "color": "#FFF8E7",
            "count": 2,
            "id": 9,
            "name": "Esther Ferrer",
            "username": "user3"
        }
    },
    "issues_per_priority": {
        "1": {
            "color": "#666666",
            "count": 13,
            "id": 1,
            "name": "Patch name"
        },
        "2": {
            "color": "#669933",
            "count": 10,
            "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": 7,
            "id": 2,
            "name": "Minor"
        },
        "3": {
            "color": "#0000FF",
            "count": 7,
            "id": 3,
            "name": "Normal"
        },
```

```
        "4": {
            "color": "#FFA500",
            "count": 2,
            "id": 4,
            "name": "Important"
        },
        "5": {
            "color": "#CC0000",
            "count": 7,
            "id": 5,
            "name": "Critical"
        }
    },
    "issues_per_status": {
        "1": {
            "color": "#8C2318",
            "count": 6,
            "id": 1,
            "name": "Patch status name"
        },
        "2": {
            "color": "#5E8C6A",
            "count": 2,
            "id": 2,
            "name": "In progress"
        },
        "3": {
            "color": "#88A65E",
            "count": 2,
            "id": 3,
            "name": "Ready for test"
        },
        "4": {
            "color": "#BFB35A",
            "count": 6,
            "id": 4,
            "name": "Closed"
        },
        "5": {
            "color": "#89BAB4",
            "count": 2,
            "id": 5,
            "name": "Needs Info"
        },
        "6": {
            "color": "#CC0000",
            "count": 8,
            "id": 6,
```

```
            "name": "Rejected"
        },
        "7": {
            "color": "#666666",
            "count": 2,
            "id": 7,
            "name": "Postponed"
        }
    },
    "issues_per_type": {
        "1": {
            "color": "#89BAB4",
            "count": 13,
            "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,
                16
            ],
            "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,
                0,
                0,
                0,
                0,
                28
            ]
        },
        "by_priority": {
            "1": {
                "color": "#666666",
                "data": [
```

```
                0,
                0,
                0,
```

```
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                10
            ],
            "id": 2,
            "name": "Normal"
        },
        "3": {
            "color": "#CC0000",
            "data": [
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
```

```
                    0,
                    5
                ],
                "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,
                    0,
                    0,
                    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,
                0,
                0,
                7
            ],
            "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,
                0,
                7
            ],
            "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,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                0,
                2
```

```
                ],
                "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,
                    0,
                    0,
                    7
                ],
                "id": 5,
                "name": "Critical"
            }
        },
        "by_status": {}
    },
    "opened_issues": 12,
    "total_issues": 28
}
```

45.6. Project tag colors data detail


```
{
    "a": null,
    "ab": null,
    "accusamus": null,
    "accusantium": null,
    "ad": null,
    "adipisci": null,
    "alias": null,
    "aliquam": "#631249",
    "amet": null,
    "animi": null,
    "aperiam": null,
    "architecto": null,
    "asperiores": null,
    "aspernatur": "#82854c",
    "assumenda": null,
    "at": "#27e90d",
    "atque": null,
    "aut": "#9ae4e4",
    "autem": null,
    "beatae": null,
    "blanditiis": null,
    "commodi": null,
    "consectetur": null,
    "consequatur": null,
    "consequuntur": null,
    "corporis": null,
    "corrupti": "#432493",
    "culpa": null,
    "cum": null,
    "cumque": "#ad75ec",
    "cupiditate": "#144bba",
    "debitis": "#9631e4",
    "delectus": "#959608",
    "deleniti": "#6188db",
    "deserunt": "#e7b695",
    "dicta": null,
    "dignissimos": null,
    "dolor": "#641bd9",
    "dolore": "#61b076",
    "dolorem": null,
    "doloremque": null,
    "dolores": "#7fea8e",
    "doloribus": "#fb1b00",
```

```
    "dolorum": null,
    "ducimus": "#ea6bb9",
    "ea": "#2c80b2",
    "eaque": null,
    "earum": null,
    "eius": "#860b86",
    "eligendi": "#5d8273",
    "enim": null,
    "eos": "#8a6433",
    "error": null,
    "esse": null,
    "est": "#665de1",
    "et": null,
    "eum": "#ee6c40",
    "eveniet": null,
    "ex": "#e06613",
    "excepturi": null,
    "exercitationem": "#ac7c74",
    "expedita": "#740c41",
    "explicabo": "#2892cb",
    "facere": null,
    "facilis": null,
    "fuga": "#e86797",
    "fugiat": null,
    "fugit": "#9345df",
    "harum": "#b42d3c",
    "hic": null,
    "id": null,
    "illo": "#3531fd",
    "illum": null,
    "impedit": null,
    "in": null,
    "incidunt": "#3099ec",
    "inventore": "#2fbc07",
    "ipsa": "#ffa8ed",
    "ipsam": null,
    "ipsum": "#da3ba4",
    "iste": "#491b3a",
    "itaque": null,
    "iure": "#019320",
    "iusto": "#3a10e8",
    "labore": "#6fdf52",
    "laboriosam": "#b2966d",
    "laborum": null,
    "laudantium": "#9e3f1f",
    "libero": null,
    "magnam": "#d1fac1",
    "magni": "#429e6f",
```

```
    "maiores": null,
    "maxime": "#1acc29",
    "minima": null,
    "minus": "#59b653",
    "modi": "#494e30",
    "molestiae": null,
    "molestias": "#92db0b",
    "mollitia": "#002e7f",
    "nam": null,
    "natus": null,
    "necessitatibus": "#84e3b6",
    "nemo": null,
    "neque": null,
    "nesciunt": null,
    "nihil": null,
    "nisi": null,
    "nobis": null,
    "non": null,
    "nostrum": "#0cf81b",
    "nulla": "#894727",
    "numquam": null,
    "obcaecati": null,
    "odio": null,
    "odit": null,
    "officia": "#c4f027",
    "officiis": "#964862",
    "omnis": null,
    "optio": null,
    "perferendis": "#999645",
    "perspiciatis": null,
    "placeat": "#d97204",
    "porro": null,
    "possimus": "#fccc1b",
    "provident": "#7fdcf2",
    "quae": "#d91a8b",
    "quaerat": "#0b4425",
    "quam": null,
    "quas": "#6e3390",
    "quasi": "#5dae16",
    "qui": null,
    "quia": null,
    "quibusdam": null,
    "quidem": "#ae6519",
    "quis": null,
    "quisquam": null,
    "quo": "#857670",
    "quod": "#0e5b24",
    "quos": null,
```

```
    "ratione": null,
    "reiciendis": "#560ff6",
    "rem": "#688119",
    "repellat": null,
    "repellendus": null,
    "reprehenderit": null,
    "repudiandae": "#3a2b71",
    "rerum": null,
    "sapiente": "#850c56",
    "sed": null,
    "sequi": "#9f6274",
    "similique": null,
    "sint": null,
    "sit": "#abdcde",
    "suscipit": null,
    "tempora": null,
    "tempore": null,
    "temporibus": "#a2c51a",
    "tenetur": null,
    "totam": "#560a5d",
    "ullam": "#98ad13",
    "unde": "#da2470",
    "ut": "#e74669",
    "vel": null,
    "velit": null,
    "veniam": null,
    "veritatis": "#768459",
    "vero": null,
    "vitae": "#d9fe5e",
    "voluptas": null,
    "voluptate": "#b0eff0",
    "voluptatem": "#00d60c",
    "voluptates": null,
    "voluptatibus": "#681ad4",
    "voluptatum": null
}
```


45.7. Project voter detail


```
{
    "full_name": "Administrator",
    "id": 5,
    "username": "admin"
}
```

45.8. Project watcher detail


```
{
    "full_name": "Administrator",
    "id": 5,
    "username": "admin"
}
```


45.9. Membership detail


```
{
    "color": "#FFCC00",
    "created_at": "2016-10-14T09:03:27.673Z",
    "email": "user8@taigaio.demo",
    "full_name": "Andrea Fernandez",
    "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": 1476435807673
}
```


45.10. Milestone detail


```
{
    "closed": false,
    "closed_points": 42.0,
    "created_date": "2016-08-20T09:03:28.508Z",
    "disponibility": 0.0,
    "estimated_finish": "2016-09-04",
    "estimated_start": "2016-08-20",
```

```
    "id": 1,
    "is_watcher": false,
    "modified_date": "2016-10-14T09:20:39.500Z",
    "name": "Sprint 2",
    "order": 10,
    "owner": 10,
    "project": 1,
    "slug": "sprint-2016-8-20",
    "total_points": 184.0,
    "total_watchers": 0,
    "user_stories": [
        {
            "assigned_to": 9,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Esther Ferrer",
                "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
                "id": 9,
                "is_active": true,
                "photo": null,
                "username": "user3"
            },
            "attachments": [],
            "backlog_order": 10,
            "blocked_note": "",
            "client_requirement": false,
            "comment": "",
            "created_date": "2016-10-14T09:03:28.588Z",
            "epic_order": null,
            "epics": null,
            "external_reference": null,
            "finish_date": null,
            "generated_from_issue": null,
            "id": 1,
            "is_blocked": false,
            "is_closed": false,
            "is_voter": false,
            "is_watcher": false,
            "kanban_order": 10,
            "milestone": 1,
            "milestone_name": "Sprint 2",
            "milestone_slug": "sprint-2016-8-20",
            "modified_date": "2016-10-14T09:20:02.755Z",
            "origin_issue": null,
            "owner": 8,
            "owner_extra_info": {
                "big_photo": null,
                "full_name_display": "Alba Leon",
```

```
            "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
            "id": 8,
            "is_active": true,
            "photo": null,
            "username": "user2"
        },
        "points": {
            "1": 8,
            "2": 2,
            "3": 5,
            "4": 6
        },
        "project": 1,
        "project_extra_info": {
            "id": 1,
            "logo_small_url": null,
            "name": "Beta project patch",
            "slug": "project-0"
        },
        "ref": 1,
        "sprint_order": 10,
        "status": 3,
        "status_extra_info": {
            "color": "#ff9900",
            "is_closed": false,
            "name": "En curso"
        },
        "subject": "Patching subject",
        "tags": [
            [
                "modi",
                null
            ]
        ],
        "tasks": [],
        "team_requirement": false,
        "total_points": 13.0,
        "total_voters": 0,
        "total_watchers": 6,
        "tribe_gig": null,
        "version": 2,
        "watchers": [
            3,
            6,
            7,
            10,
            11,
            15
```

```
            ]
        },
        {
            "assigned_to": 7,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Marcos Ortiz",
                "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
                "id": 7,
                "is_active": true,
                "photo": null,
                "username": "user1"
            },
            "attachments": [],
            "backlog_order": 15,
            "blocked_note": "",
            "client_requirement": false,
            "comment": "",
            "created_date": "2016-10-14T09:03:30.053Z",
            "epic_order": null,
            "epics": null,
            "external_reference": null,
            "finish_date": null,
            "generated_from_issue": null,
            "id": 2,
            "is_blocked": false,
            "is_closed": false,
            "is_voter": false,
            "is_watcher": false,
            "kanban_order": 15,
            "milestone": 1,
            "milestone_name": "Sprint 2",
            "milestone_slug": "sprint-2016-8-20",
            "modified_date": "2016-10-14T09:03:30.316Z",
            "origin_issue": null,
            "owner": 9,
            "owner_extra_info": {
                "big_photo": null,
                "full_name_display": "Esther Ferrer",
                "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
                "id": 9,
                "is_active": true,
                "photo": null,
                "username": "user3"
            },
            "points": {
                "1": 5,
                "2": 3,
```

```
                "3": 12,
                "4": 5
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 5,
            "sprint_order": 15,
            "status": 3,
            "status_extra_info": {
                "color": "#ff9900",
                "is_closed": false,
                "name": "En curso"
            },
            "subject": "get_actions() does not check for 'delete_selected' in actions",
            "tags": [
                [
                    "amet",
                    null
                ],
                [
                    "totam",
                    null
                ],
                [
                    "mollitia",
                    null
                ]
            ],
            "tasks": [],
            "team_requirement": false,
            "total_points": 44.5,
            "total_voters": 2,
            "total_watchers": 7,
            "tribe_gig": null,
            "version": 1,
            "watchers": [
                1,
                5,
                6,
                7,
                9,
                10,
                14
```

```
            ]
        },
        {
            "assigned_to": 7,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Marcos Ortiz",
                "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
                "id": 7,
                "is_active": true,
                "photo": null,
                "username": "user1"
            },
            "attachments": [],
            "backlog_order": 1476435812338,
            "blocked_note": "",
            "client_requirement": false,
            "comment": "",
            "created_date": "2016-10-14T09:03:32.338Z",
            "epic_order": null,
            "epics": [
                {
                    "color": "#4e9a06",
                    "id": 4,
                    "project": {
                        "id": 1,
                        "name": "Beta project patch",
                        "slug": "project-0"
                    },
                    "ref": 123,
                    "subject": "Feature/improved image admin"
                }
            ],
            "external_reference": null,
            "finish_date": "2016-10-14T09:03:32.814Z",
            "generated_from_issue": null,
            "id": 3,
            "is_blocked": false,
            "is_closed": true,
            "is_voter": false,
            "is_watcher": false,
            "kanban_order": 1476435812338,
            "milestone": 1,
            "milestone_name": "Sprint 2",
            "milestone_slug": "sprint-2016-8-20",
            "modified_date": "2016-10-14T09:03:32.638Z",
            "origin_issue": null,
            "owner": 13,
```

```
            "owner_extra_info": {
                "big_photo": null,
                "full_name_display": "Alvaro Molina",
                "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
                "id": 13,
                "is_active": true,
                "photo": null,
                "username": "user7"
            },
            "points": {
                "1": 11,
                "2": 4,
                "3": 4,
                "4": 11
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 11,
            "sprint_order": 1476435812339,
            "status": 4,
            "status_extra_info": {
                "color": "#fcc000",
                "is_closed": false,
                "name": "Lista para testear"
            },
            "subject": "Add tests for bulk operations",
            "tags": [
                [
                    "fugiat",
                    null
                ]
            ],
            "tasks": [],
            "team_requirement": false,
            "total_points": 42.0,
            "total_voters": 1,
            "total_watchers": 2,
            "tribe_gig": null,
            "version": 1,
            "watchers": [
                13,
                15
            ]
```

```
        },
        {
            "assigned_to": 6,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Silvia Soto",
                "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
                "id": 6,
                "is_active": true,
                "photo": null,
                "username": "user6532909695705815086"
            },
            "attachments": [],
            "backlog_order": 1476435813224,
            "blocked_note": "",
            "client_requirement": false,
            "comment": "",
            "created_date": "2016-10-14T09:03:33.224Z",
            "epic_order": null,
            "epics": null,
            "external_reference": null,
            "finish_date": null,
            "generated_from_issue": null,
            "id": 4,
            "is_blocked": false,
            "is_closed": false,
            "is_voter": false,
            "is_watcher": false,
            "kanban_order": 1476435813224,
            "milestone": 1,
            "milestone_name": "Sprint 2",
            "milestone_slug": "sprint-2016-8-20",
            "modified_date": "2016-10-14T09:03:33.529Z",
            "origin_issue": null,
            "owner": 14,
            "owner_extra_info": {
                "big_photo": null,
                "full_name_display": "Andrea Fernandez",
                "gravatar_id": "dce0e8ed702cd85d5132e523121e619b",
                "id": 14,
                "is_active": true,
                "photo": null,
                "username": "user8"
            },
            "points": {
                "1": 12,
                "2": 4,
                "3": 4,
```

```
            "4": 9
        },
        "project": 1,
        "project_extra_info": {
            "id": 1,
            "logo_small_url": null,
            "name": "Beta project patch",
            "slug": "project-0"
        },
        "ref": 13,
        "sprint_order": 1476435813225,
        "status": 1,
        "status_extra_info": {
            "color": "#999999",
            "is_closed": false,
            "name": "Patch status name"
        },
        "subject": "Support for bulk actions",
        "tags": [
            [
                "tempore",
                null
            ]
        ],
        "tasks": [],
        "team_requirement": false,
        "total_points": 52.0,
        "total_voters": 6,
        "total_watchers": 3,
        "tribe_gig": null,
        "version": 1,
        "watchers": [
            3,
            9,
            11
        ]
    },
    {
        "assigned_to": 6,
        "assigned_to_extra_info": {
            "big_photo": null,
            "full_name_display": "Silvia Soto",
            "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
            "id": 6,
            "is_active": true,
            "photo": null,
            "username": "user6532909695705815086"
        },
```

```
            "attachments": [],
            "backlog_order": 1476435815902,
            "blocked_note": "",
            "client_requirement": false,
            "comment": "",
            "created_date": "2016-10-14T09:03:35.902Z",
            "epic_order": null,
            "epics": null,
            "external_reference": null,
            "finish_date": null,
            "generated_from_issue": null,
            "id": 5,
            "is_blocked": false,
            "is_closed": false,
            "is_voter": false,
            "is_watcher": false,
            "kanban_order": 1476435815902,
            "milestone": 1,
            "milestone_name": "Sprint 2",
            "milestone_slug": "sprint-2016-8-20",
            "modified_date": "2016-10-14T09:03:36.305Z",
            "origin_issue": null,
            "owner": 7,
            "owner_extra_info": {
                "big_photo": null,
                "full_name_display": "Marcos Ortiz",
                "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
                "id": 7,
                "is_active": true,
                "photo": null,
                "username": "user1"
            },
            "points": {
                "1": 8,
                "2": 7,
                "3": 3,
                "4": 8
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 19,
            "sprint_order": 1476435815903,
            "status": 3,
```

```
            "status_extra_info": {
                "color": "#ff9900",
                "is_closed": false,
                "name": "En curso"
            },
            "subject": "Fixing templates for Django 1.6.",
            "tags": [
                [
                    "impedit",
                    null
                ],
                [
                    "voluptatem",
                    null
                ]
            ],
            "tasks": [],
            "team_requirement": false,
            "total_points": 21.5,
            "total_voters": 0,
            "total_watchers": 6,
            "tribe_gig": null,
            "version": 1,
            "watchers": [
                2,
                9,
                11,
                12,
                13,
                15
            ]
        },
        {
            "assigned_to": 8,
            "assigned_to_extra_info": {
                "big_photo": null,
                "full_name_display": "Alba Leon",
                "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
                "id": 8,
                "is_active": true,
                "photo": null,
                "username": "user2"
            },
            "attachments": [],
            "backlog_order": 1476435817671,
            "blocked_note": "",
            "client_requirement": false,
            "comment": "",
```

```
            "created_date": "2016-10-14T09:03:37.671Z",
            "epic_order": null,
            "epics": null,
            "external_reference": null,
            "finish_date": null,
            "generated_from_issue": null,
            "id": 6,
            "is_blocked": false,
            "is_closed": false,
            "is_voter": false,
            "is_watcher": false,
            "kanban_order": 1476435817671,
            "milestone": 1,
            "milestone_name": "Sprint 2",
            "milestone_slug": "sprint-2016-8-20",
            "modified_date": "2016-10-14T09:03:37.985Z",
            "origin_issue": null,
            "owner": 8,
            "owner_extra_info": {
                "big_photo": null,
                "full_name_display": "Alba Leon",
                "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
                "id": 8,
                "is_active": true,
                "photo": null,
                "username": "user2"
            },
            "points": {
                "1": 9,
                "2": 2,
                "3": 2,
                "4": 4
            },
            "project": 1,
            "project_extra_info": {
                "id": 1,
                "logo_small_url": null,
                "name": "Beta project patch",
                "slug": "project-0"
            },
            "ref": 23,
            "sprint_order": 1476435817672,
            "status": 3,
            "status_extra_info": {
                "color": "#ff9900",
                "is_closed": false,
                "name": "En curso"
            },
```

```
            "subject": "Create the user model",
            "tags": [
                [
                    "reiciendis",
                    null
                ],
                [
                    "corrupti",
                    null
                ]
            ],
            "tasks": [],
            "team_requirement": false,
            "total_points": 11.0,
            "total_voters": 7,
            "total_watchers": 3,
            "tribe_gig": null,
            "version": 1,
            "watchers": [
                3,
                11,
                14
            ]
        }
    ],
    "watchers": []
}
```


45.11. Milestone watcher detail


```
{
    "full_name": "Silvia Soto",
    "id": 6,
    "username": "user6532909695705815086"
}
```


45.12. Milestone stats detail


```
{
    "completed_points": [
        20.0,
        1.0,
        1.0,
        20.0
```

```
    ],
    "completed_tasks": 7,
    "completed_userstories": 1,
    "days": [
        {
            "day": "2016-08-20",
            "name": 20,
            "open_points": 184.0,
            "optimal_points": 184.0
        },
        {
            "day": "2016-08-21",
            "name": 21,
            "open_points": 184.0,
            "optimal_points": 171.73333333333332
        },
        {
            "day": "2016-08-22",
            "name": 22,
            "open_points": 184.0,
            "optimal_points": 159.46666666666664
        },
        {
            "day": "2016-08-23",
            "name": 23,
            "open_points": 184.0,
            "optimal_points": 147.19999999999996
        },
        {
            "day": "2016-08-24",
            "name": 24,
            "open_points": 181.25,
            "optimal_points": 134.93333333333328
        },
        {
            "day": "2016-08-25",
            "name": 25,
            "open_points": 181.25,
            "optimal_points": 122.66666666666661
        },
        {
            "day": "2016-08-26",
            "name": 26,
            "open_points": 181.25,
            "optimal_points": 110.39999999999995
        },
        {
            "day": "2016-08-27",
```

```
            "name": 27,
            "open_points": 181.25,
            "optimal_points": 98.13333333333328
        },
        {
            "day": "2016-08-28",
            "name": 28,
            "open_points": 159.2,
            "optimal_points": 85.86666666666662
        },
        {
            "day": "2016-08-29",
            "name": 29,
            "open_points": 154.86666666666667,
            "optimal_points": 73.59999999999995
        },
        {
            "day": "2016-08-30",
            "name": 30,
            "open_points": 154.86666666666667,
            "optimal_points": 61.333333333333286
        },
        {
            "day": "2016-08-31",
            "name": 31,
            "open_points": 154.86666666666667,
            "optimal_points": 49.06666666666662
        },
        {
            "day": "2016-09-01",
            "name": 1,
            "open_points": 112.86666666666667,
            "optimal_points": 36.799999999999955
        },
        {
            "day": "2016-09-02",
            "name": 2,
            "open_points": 103.96666666666667,
            "optimal_points": 24.53333333333329
        },
        {
            "day": "2016-09-03",
            "name": 3,
            "open_points": 103.96666666666667,
            "optimal_points": 12.266666666666621
        },
        {
            "day": "2016-09-04",
```

```
            "name": 4,
            "open_points": 103.96666666666667,
            "optimal_points": -4.618527782440651e-14
        }
    ],
    "estimated_finish": "2016-09-04",
    "estimated_start": "2016-08-20",
    "iocaine_doses": 0,
    "name": "Sprint 2016-8-20",
    "total_points": {
        "1": 88.0,
        "2": 7.5,
        "3": 44.5,
        "4": 44.0
    },
    "total_tasks": 24,
    "total_userstories": 6
}
```


45.13. Epic detail


```
{
    "assigned_to": 11,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "German Benitez",
        "gravatar_id": "c9ba9d485f9a9153ebf53758feb0980c",
        "id": 11,
        "is_active": true,
        "photo": null,
        "username": "user5"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "color": "#fcaf3e",
    "comment": "",
    "created_date": "2016-10-14T09:04:28.063Z",
    "description": "Labore voluptate omnis voluptatum eius laboriosam quam, sed nostrum
rerum, ab dolorum provident repudiandae ad expedita. Perferendis non qui accusantium
dolor facere voluptatum modi cumque amet esse, ad velit minus consectetur enim fugit
libero officiis iusto eveniet, vel itaque voluptatum eaque? Beatae sunt amet atque
ducimus perferendis inventore accusantium esse voluptatem nulla labore, similique aliquid
amet libero minima.",
    "description_html": "<p>Labore voluptate omnis voluptatum eius laboriosam quam, sed
```

```
nostrum rerum, ab dolorum provident repudiandae ad expedita. Perferendis non qui
accusantium dolor facere voluptatum modi cumque amet esse, ad velit minus consectetur
enim fugit libero officiis iusto eveniet, vel itaque voluptatum eaque? Beatae sunt amet
atque ducimus perferendis inventore accusantium esse voluptatem nulla labore, similique
aliquid amet libero minima.</p>",
    "epics_order": 1476435868063,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "modified_date": "2016-10-14T09:04:28.220Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 121,
            "subject": "Implement the form"
        },
        "previous": {
            "id": 30,
            "ref": 127,
            "subject": "New epic"
        }
    },
    "owner": 7,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Marcos Ortiz",
        "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
        "id": 7,
        "is_active": true,
        "photo": null,
        "username": "user1"
    },
    "project": 1,
    "ref": 120,
    "status": 2,
    "status_extra_info": {
        "color": "#ff8a84",
        "is_closed": false,
        "name": "Preparada"
    },
    "subject": "Add tests for bulk operations",
    "tags": [
        [
            "quaerat",
            "#0b4425"
        ]
```

```
    ],
    "team_requirement": false,
    "total_voters": 3,
    "total_watchers": 0,
    "user_stories_counts": {
        "closed": 0,
        "opened": 0
    },
    "version": 1,
    "watchers": []
}
```


45.14. Epic detail (GET)


```
{
    "assigned_to": 11,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "German Benitez",
        "gravatar_id": "c9ba9d485f9a9153ebf53758feb0980c",
        "id": 11,
        "is_active": true,
        "photo": null,
        "username": "user5"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "color": "#fcaf3e",
    "comment": "",
    "created_date": "2016-10-14T09:04:28.063Z",
    "description": "Labore voluptate omnis voluptatum eius laboriosam quam, sed nostrum
rerum, ab dolorum provident repudiandae ad expedita. Perferendis non qui accusantium
dolor facere voluptatum modi cumque amet esse, ad velit minus consectetur enim fugit
libero officiis iusto eveniet, vel itaque voluptatum eaque? Beatae sunt amet atque
ducimus perferendis inventore accusantium esse voluptatem nulla labore, similique aliquid
amet libero minima.",
    "description_html": "<p>Labore voluptate omnis voluptatum eius laboriosam quam, sed
nostrum rerum, ab dolorum provident repudiandae ad expedita. Perferendis non qui
accusantium dolor facere voluptatum modi cumque amet esse, ad velit minus consectetur
enim fugit libero officiis iusto eveniet, vel itaque voluptatum eaque? Beatae sunt amet
atque ducimus perferendis inventore accusantium esse voluptatem nulla labore, similique
aliquid amet libero minima.</p>",
    "epics_order": 1476435868063,
    "id": 1,
```

```
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "modified_date": "2016-10-14T09:04:28.220Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 121,
            "subject": "Implement the form"
        },
        "previous": {
            "id": 30,
            "ref": 127,
            "subject": "New epic"
        }
    },
    "owner": 7,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Marcos Ortiz",
        "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
        "id": 7,
        "is_active": true,
        "photo": null,
        "username": "user1"
    },
    "project": 1,
    "ref": 120,
    "status": 2,
    "status_extra_info": {
        "color": "#ff8a84",
        "is_closed": false,
        "name": "Preparada"
    },
    "subject": "Add tests for bulk operations",
    "tags": [
        [
            "quaerat",
            "#0b4425"
        ]
    ],
    "team_requirement": false,
    "total_voters": 3,
    "total_watchers": 0,
    "user_stories_counts": {
        "closed": 0,
        "opened": 0
```

```
    },
    "version": 1,
    "watchers": []
}
```


45.15. Epic detail (LIST)


```
{
    "assigned_to": 11,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "German Benitez",
        "gravatar_id": "c9ba9d485f9a9153ebf53758feb0980c",
        "id": 11,
        "is_active": true,
        "photo": null,
        "username": "user5"
    },
    "attachments": [],
    "blocked_note": "",
    "client_requirement": false,
    "color": "#fcaf3e",
    "created_date": "2016-10-14T09:04:28.063Z",
    "epics_order": 1476435868063,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "modified_date": "2016-10-14T09:04:28.220Z",
    "owner": 7,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Marcos Ortiz",
        "gravatar_id": "aed1e43be0f69f07ce6f34a907bc6328",
        "id": 7,
        "is_active": true,
        "photo": null,
        "username": "user1"
    },
    "project": 1,
    "ref": 120,
    "status": 2,
    "status_extra_info": {
        "color": "#ff8a84",
        "is_closed": false,
```

```
        "name": "Preparada"
    },
    "subject": "Add tests for bulk operations",
    "tags": [
        [
            "quaerat",
            "#0b4425"
        ]
    ],
    "team_requirement": false,
    "total_voters": 3,
    "total_watchers": 0,
    "user_stories_counts": {
        "closed": 0,
        "opened": 0
    },
    "version": 1,
    "watchers": []
}
```


45.16. Epic filters data detail


```
{
    "assigned_to": [
        {
            "count": 8,
            "full_name": "",
            "id": null
        },
        {
            "count": 0,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 0,
            "full_name": "Alba Leon",
            "id": 8
        },
        {
            "count": 0,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 0,
```

```
            "full_name": "Andrea Fernandez",
            "id": 14
        },
        {
            "count": 0,
            "full_name": "Catalina Roman",
            "id": 15
        },
        {
            "count": 1,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 1,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 0,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 1,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 0,
            "full_name": "Silvia Soto",
            "id": 6
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "owners": [
        {
            "count": 2,
            "full_name": "Esther Ferrer",
```

```
            "id": 9
        },
        {
            "count": 1,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 2,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 2,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 5,
            "full_name": "Silvia Soto",
            "id": 6
        }
    ],
    "statuses": [
        {
            "color": "#999999",
            "count": 6,
            "id": 1,
            "name": "Nueva",
            "order": 1
        },
        {
            "color": "#ff8a84",
            "count": 4,
            "id": 2,
            "name": "Preparada",
            "order": 2
        },
        {
            "color": "#ff9900",
            "count": 0,
            "id": 3,
            "name": "En curso",
            "order": 3
        },
        {
            "color": "#fcc000",
            "count": 2,
```

```
            "id": 4,
            "name": "Lista para testear",
            "order": 4
        },
        {
            "color": "#669900",
            "count": 0,
            "id": 5,
            "name": "Hecha",
            "order": 5
        }
    ],
    "tags": [
        {
            "color": null,
            "count": 0,
            "name": "a"
        },
        {
            "color": null,
            "count": 0,
            "name": "ab"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusamus"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "ad"
        },
        {
            "color": null,
            "count": 0,
            "name": "adipisci"
        },
        {
            "color": null,
            "count": 0,
            "name": "alias"
        },
```

```
        {
            "color": "#631249",
            "count": 0,
            "name": "aliquam"
        },
        {
            "color": null,
            "count": 1,
            "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": "#82854c",
            "count": 0,
            "name": "aspernatur"
        },
        {
            "color": null,
            "count": 0,
            "name": "assumenda"
        },
        {
            "color": "#27e90d",
            "count": 0,
            "name": "at"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "atque"
        },
        {
            "color": "#9ae4e4",
            "count": 0,
            "name": "aut"
        },
        {
            "color": null,
            "count": 0,
            "name": "autem"
        },
        {
            "color": null,
            "count": 0,
            "name": "beatae"
        },
        {
            "color": null,
            "count": 0,
            "name": "blanditiis"
        },
        {
            "color": null,
            "count": 0,
            "name": "commodi"
        },
        {
            "color": null,
            "count": 0,
            "name": "consectetur"
        },
        {
            "color": null,
            "count": 0,
            "name": "consequatur"
        },
        {
            "color": null,
            "count": 0,
            "name": "consequuntur"
        },
        {
            "color": null,
            "count": 1,
            "name": "corporis"
        },
        {
```

```
            "color": "#432493",
            "count": 0,
            "name": "corrupti"
        },
        {
            "color": null,
            "count": 0,
            "name": "culpa"
        },
        {
            "color": null,
            "count": 0,
            "name": "cum"
        },
        {
            "color": "#ad75ec",
            "count": 0,
            "name": "cumque"
        },
        {
            "color": "#144bba",
            "count": 0,
            "name": "cupiditate"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": "#9631e4",
            "count": 0,
            "name": "debitis"
        },
        {
            "color": "#959608",
            "count": 0,
            "name": "delectus"
        },
        {
            "color": "#6188db",
            "count": 0,
            "name": "deleniti"
        },
        {
            "color": "#e7b695",
            "count": 0,
            "name": "deserunt"
```

```
        },
        {
            "color": null,
            "count": 0,
            "name": "dicta"
        },
        {
            "color": null,
            "count": 1,
            "name": "dignissimos"
        },
        {
            "color": "#641bd9",
            "count": 0,
            "name": "dolor"
        },
        {
            "color": "#61b076",
            "count": 0,
            "name": "dolore"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolorem"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloremque"
        },
        {
            "color": "#7fea8e",
            "count": 0,
            "name": "dolores"
        },
        {
            "color": "#fb1b00",
            "count": 0,
            "name": "doloribus"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolorum"
        },
        {
            "color": "#ea6bb9",
```

```
            "count": 0,
            "name": "ducimus"
        },
        {
            "color": "#2c80b2",
            "count": 0,
            "name": "ea"
        },
        {
            "color": null,
            "count": 0,
            "name": "eaque"
        },
        {
            "color": null,
            "count": 0,
            "name": "earum"
        },
        {
            "color": "#860b86",
            "count": 0,
            "name": "eius"
        },
        {
            "color": null,
            "count": 0,
            "name": "enim"
        },
        {
            "color": "#8a6433",
            "count": 1,
            "name": "eos"
        },
        {
            "color": null,
            "count": 0,
            "name": "error"
        },
        {
            "color": null,
            "count": 0,
            "name": "esse"
        },
        {
            "color": "#665de1",
            "count": 0,
            "name": "est"
        },
```

```
        {
            "color": null,
            "count": 0,
            "name": "et"
        },
        {
            "color": "#ee6c40",
            "count": 0,
            "name": "eum"
        },
        {
            "color": null,
            "count": 0,
            "name": "eveniet"
        },
        {
            "color": "#e06613",
            "count": 0,
            "name": "ex"
        },
        {
            "color": null,
            "count": 0,
            "name": "excepturi"
        },
        {
            "color": "#ac7c74",
            "count": 0,
            "name": "exercitationem"
        },
        {
            "color": "#740c41",
            "count": 0,
            "name": "expedita"
        },
        {
            "color": "#2892cb",
            "count": 0,
            "name": "explicabo"
        },
        {
            "color": null,
            "count": 0,
            "name": "facere"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "facilis"
        },
        {
            "color": "#e86797",
            "count": 0,
            "name": "fuga"
        },
        {
            "color": null,
            "count": 0,
            "name": "fugiat"
        },
        {
            "color": "#9345df",
            "count": 0,
            "name": "fugit"
        },
        {
            "color": "#b42d3c",
            "count": 0,
            "name": "harum"
        },
        {
            "color": null,
            "count": 0,
            "name": "hic"
        },
        {
            "color": null,
            "count": 0,
            "name": "id"
        },
        {
            "color": "#3531fd",
            "count": 0,
            "name": "illo"
        },
        {
            "color": null,
            "count": 0,
            "name": "illum"
        },
        {
            "color": null,
            "count": 0,
            "name": "impedit"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "in"
        },
        {
            "color": "#3099ec",
            "count": 0,
            "name": "incidunt"
        },
        {
            "color": "#2fbc07",
            "count": 0,
            "name": "inventore"
        },
        {
            "color": "#ffa8ed",
            "count": 0,
            "name": "ipsa"
        },
        {
            "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": "#019320",
            "count": 0,
            "name": "iure"
        },
        {
            "color": "#3a10e8",
            "count": 0,
            "name": "iusto"
```

```
        },
        {
            "color": "#6fdf52",
            "count": 0,
            "name": "labore"
        },
        {
            "color": "#b2966d",
            "count": 0,
            "name": "laboriosam"
        },
        {
            "color": null,
            "count": 0,
            "name": "laborum"
        },
        {
            "color": "#9e3f1f",
            "count": 0,
            "name": "laudantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "libero"
        },
        {
            "color": "#d1fac1",
            "count": 0,
            "name": "magnam"
        },
        {
            "color": "#429e6f",
            "count": 0,
            "name": "magni"
        },
        {
            "color": null,
            "count": 1,
            "name": "maiores"
        },
        {
            "color": "#1acc29",
            "count": 0,
            "name": "maxime"
        },
        {
            "color": null,
```

```
            "count": 1,
            "name": "minima"
        },
        {
            "color": "#59b653",
            "count": 0,
            "name": "minus"
        },
        {
            "color": "#494e30",
            "count": 0,
            "name": "modi"
        },
        {
            "color": null,
            "count": 0,
            "name": "molestiae"
        },
        {
            "color": "#92db0b",
            "count": 0,
            "name": "molestias"
        },
        {
            "color": "#002e7f",
            "count": 1,
            "name": "mollitia"
        },
        {
            "color": null,
            "count": 0,
            "name": "nam"
        },
        {
            "color": null,
            "count": 1,
            "name": "natus"
        },
        {
            "color": "#84e3b6",
            "count": 0,
            "name": "necessitatibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "nemo"
        },
```

```
        {
            "color": null,
            "count": 0,
            "name": "neque"
        },
        {
            "color": null,
            "count": 0,
            "name": "nesciunt"
        },
        {
            "color": null,
            "count": 0,
            "name": "nihil"
        },
        {
            "color": null,
            "count": 0,
            "name": "nisi"
        },
        {
            "color": null,
            "count": 0,
            "name": "nobis"
        },
        {
            "color": null,
            "count": 0,
            "name": "non"
        },
        {
            "color": "#0cf81b",
            "count": 0,
            "name": "nostrum"
        },
        {
            "color": "#894727",
            "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": "#964862",
            "count": 0,
            "name": "officiis"
        },
        {
            "color": null,
            "count": 0,
            "name": "omnis"
        },
        {
            "color": null,
            "count": 0,
            "name": "optio"
        },
        {
            "color": null,
            "count": 0,
            "name": "perspiciatis"
        },
        {
            "color": "#d97204",
            "count": 0,
            "name": "placeat"
        },
        {
            "color": null,
            "count": 0,
            "name": "porro"
        },
        {
```

```
            "color": "#fccc1b",
            "count": 0,
            "name": "possimus"
        },
        {
            "color": "#7fdcf2",
            "count": 0,
            "name": "provident"
        },
        {
            "color": "#d91a8b",
            "count": 0,
            "name": "quae"
        },
        {
            "color": "#0b4425",
            "count": 1,
            "name": "quaerat"
        },
        {
            "color": null,
            "count": 0,
            "name": "quam"
        },
        {
            "color": "#6e3390",
            "count": 0,
            "name": "quas"
        },
        {
            "color": "#5dae16",
            "count": 0,
            "name": "quasi"
        },
        {
            "color": null,
            "count": 0,
            "name": "qui"
        },
        {
            "color": null,
            "count": 1,
            "name": "quia"
        },
        {
            "color": null,
            "count": 0,
            "name": "quibusdam"
```

```
        },
        {
            "color": "#ae6519",
            "count": 0,
            "name": "quidem"
        },
        {
            "color": null,
            "count": 0,
            "name": "quis"
        },
        {
            "color": null,
            "count": 0,
            "name": "quisquam"
        },
        {
            "color": "#857670",
            "count": 0,
            "name": "quo"
        },
        {
            "color": "#0e5b24",
            "count": 0,
            "name": "quod"
        },
        {
            "color": null,
            "count": 0,
            "name": "quos"
        },
        {
            "color": null,
            "count": 0,
            "name": "ratione"
        },
        {
            "color": "#560ff6",
            "count": 1,
            "name": "reiciendis"
        },
        {
            "color": "#688119",
            "count": 0,
            "name": "rem"
        },
        {
            "color": null,
```

```
            "count": 0,
            "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": "#850c56",
            "count": 0,
            "name": "sapiente"
        },
        {
            "color": null,
            "count": 0,
            "name": "sed"
        },
        {
            "color": "#9f6274",
            "count": 0,
            "name": "sequi"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
        {
            "color": null,
            "count": 0,
            "name": "similique"
        },
```

```
        {
            "color": null,
            "count": 0,
            "name": "sint"
        },
        {
            "color": "#abdcde",
            "count": 0,
            "name": "sit"
        },
        {
            "color": null,
            "count": 0,
            "name": "suscipit"
        },
        {
            "color": null,
            "count": 1,
            "name": "tempora"
        },
        {
            "color": null,
            "count": 0,
            "name": "tempore"
        },
        {
            "color": "#a2c51a",
            "count": 0,
            "name": "temporibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "tenetur"
        },
        {
            "color": "#560a5d",
            "count": 0,
            "name": "totam"
        },
        {
            "color": "#98ad13",
            "count": 0,
            "name": "ullam"
        },
        {
            "color": "#da2470",
            "count": 0,
```

```
            "name": "unde"
        },
        {
            "color": "#e74669",
            "count": 0,
            "name": "ut"
        },
        {
            "color": null,
            "count": 0,
            "name": "vel"
        },
        {
            "color": null,
            "count": 0,
            "name": "velit"
        },
        {
            "color": null,
            "count": 0,
            "name": "veniam"
        },
        {
            "color": "#768459",
            "count": 0,
            "name": "veritatis"
        },
        {
            "color": null,
            "count": 0,
            "name": "vero"
        },
        {
            "color": "#d9fe5e",
            "count": 0,
            "name": "vitae"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptas"
        },
        {
            "color": "#b0eff0",
            "count": 0,
            "name": "voluptate"
        },
        {
```

```
            "color": "#00d60c",
            "count": 0,
            "name": "voluptatem"
        },
        {
            "color": "#681ad4",
            "count": 0,
            "name": "voluptatibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptatum"
        }
    ]
}
```


45.17. Epic voter detail


```
{
    "full_name": "Gogs",
    "id": 4,
    "username": "gogs-f33ddbd6ecc64da1b18d5b9bd902d2b6"
}
```


45.18. Epic watcher detail


```
{
    "full_name": "Alba Leon",
    "id": 8,
    "username": "user2"
}
```


45.19. Epic status detail

```
{
    "color": "#999999",
    "id": 1,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name"
}
```


45.20. Epic custom attribute detail


```
{
    "created_date": "2016-10-14T09:04:31.724Z",
    "description": "provident maxime facilis nisi voluptatibus repellendus quos porro
sunt minus accusantium similique",
    "id": 8,
    "modified_date": "2016-10-14T09:19:59.317Z",
    "name": "Duration 1",
    "order": 1,
    "project": 2,
    "type": "multiline"
}
```


45.21. Epic custom attributes values detail


```
{
    "attributes_values": {
        "8": "240 min"
    },
    "epic": 10,
    "version": 2
}
```


45.22. Epic related user story detail

```
{
    "epic": 10,
    "order": 100,
    "user_story": 37
}
```


45.23. User story detail


```
{
    "assigned_to": 9,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Esther Ferrer",
        "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
        "id": 9,
        "is_active": true,
        "photo": null,
        "username": "user3"
    },
    "attachments": [],
    "backlog_order": 1476435808588,
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "comment": "",
    "created_date": "2016-10-14T09:03:28.588Z",
    "description": "Aspernatur perspiciatis dolores praesentium quia minima fuga?",
    "description_html": "<p>Aspernatur perspiciatis dolores praesentium quia minima
fuga?</p>",
    "epic_order": null,
    "epics": null,
    "external_reference": null,
    "finish_date": null,
    "generated_from_issue": null,
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "kanban_order": 1476435808588,
    "milestone": 1,
    "milestone_name": "Sprint 2016-8-20",
    "milestone_slug": "sprint-2016-8-20",
    "modified_date": "2016-10-14T09:20:02.755Z",
    "neighbors": {
```

```
        "next": {
            "id": 2,
            "ref": 5,
            "subject": "get_actions() does not check for 'delete_selected' in actions"
        },
        "previous": {
            "id": 150,
            "ref": 135,
            "subject": "Customer personal data"
        }
    },
    "origin_issue": null,
    "owner": 8,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Alba Leon",
        "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
        "id": 8,
        "is_active": true,
        "photo": null,
        "username": "user2"
    },
    "points": {
        "1": 8,
        "2": 2,
        "3": 5,
        "4": 6
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 1,
    "sprint_order": 10,
    "status": 3,
    "status_extra_info": {
        "color": "#ff9900",
        "is_closed": false,
        "name": "En curso"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "modi",
            "#494e30"
```

```
        ]
    ],
    "tasks": [],
    "team_requirement": false,
    "total_points": 13.0,
    "total_voters": 0,
    "total_watchers": 5,
    "tribe_gig": null,
    "version": 2,
    "watchers": [
        3,
        7,
        10,
        11,
        15
    ]
}
```


45.24. User story detail (GET)


```
{
    "assigned_to": 9,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Esther Ferrer",
        "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
        "id": 9,
        "is_active": true,
        "photo": null,
        "username": "user3"
    },
    "attachments": [],
    "backlog_order": 1476435808588,
    "blocked_note": "",
    "blocked_note_html": "",
    "client_requirement": false,
    "comment": "",
    "created_date": "2016-10-14T09:03:28.588Z",
    "description": "Aspernatur perspiciatis dolores praesentium quia minima fuga?",
    "description_html": "<p>Aspernatur perspiciatis dolores praesentium quia minima
fuga?</p>",
    "epic_order": null,
    "epics": null,
    "external_reference": null,
    "finish_date": null,
    "generated_from_issue": null,
```

```
    "id": 1,
    "is_blocked": false,
    "is_closed": false,
    "is_voter": false,
    "is_watcher": false,
    "kanban_order": 1476435808588,
    "milestone": 1,
    "milestone_name": "Sprint 2016-8-20",
    "milestone_slug": "sprint-2016-8-20",
    "modified_date": "2016-10-14T09:20:02.755Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 5,
            "subject": "get_actions() does not check for 'delete_selected' in actions"
        },
        "previous": {
            "id": 150,
            "ref": 135,
            "subject": "Customer personal data"
        }
    },
    "origin_issue": null,
    "owner": 8,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Alba Leon",
        "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
        "id": 8,
        "is_active": true,
        "photo": null,
        "username": "user2"
    },
    "points": {
        "1": 8,
        "2": 2,
        "3": 5,
        "4": 6
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 1,
    "sprint_order": 10,
```

```
    "status": 3,
    "status_extra_info": {
        "color": "#ff9900",
        "is_closed": false,
        "name": "En curso"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "modi",
            "#494e30"
        ]
    ],
    "tasks": [],
    "team_requirement": false,
    "total_points": 13.0,
    "total_voters": 0,
    "total_watchers": 5,
    "tribe_gig": null,
    "version": 2,
    "watchers": [
        3,
        7,
        10,
        11,
        15
    ]
}
```


NOTE neighbors is a read only field

45.25. User story detail (LIST)


```
{
    "assigned_to": null,
    "assigned_to_extra_info": null,
    "attachments": [],
    "backlog_order": 2,
    "blocked_note": "",
    "client_requirement": false,
    "comment": "",
    "created_date": "2016-10-14T09:20:03.085Z",
    "epic_order": null,
    "epics": null,
    "external_reference": null,
```

```
    "finish_date": null,
    "generated_from_issue": null,
    "id": 150,
    "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": "2016-10-14T09:20:03.105Z",
    "origin_issue": null,
    "owner": 6,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Silvia Soto",
        "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
        "id": 6,
        "is_active": true,
        "photo": null,
        "username": "user6532909695705815086"
    },
    "points": {
        "1": 4,
        "2": 3,
        "3": 2,
        "4": 1
    },
    "project": 1,
    "project_extra_info": {
        "id": 1,
        "logo_small_url": null,
        "name": "Project Example 0",
        "slug": "project-0"
    },
    "ref": 135,
    "sprint_order": 2,
    "status": 2,
    "status_extra_info": {
        "color": "#ff8a84",
        "is_closed": false,
        "name": "Preparada"
    },
    "subject": "Customer personal data",
    "tags": [
        [
            "service catalog",
```

```
            null
        ],
        [
            "customer",
            null
        ]
    ],
    "tasks": [],
    "team_requirement": false,
    "total_points": 1.5,
    "total_voters": 0,
    "total_watchers": 0,
    "tribe_gig": null,
    "version": 1,
    "watchers": []
}
```


45.26. Issue filters data detail


```
{
    "assigned_to": [
        {
            "count": 9,
            "full_name": "",
            "id": null
        },
        {
            "count": 1,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 2,
            "full_name": "Alba Leon",
            "id": 8
        },
        {
            "count": 3,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 1,
            "full_name": "Andrea Fernandez",
            "id": 14
        },
```

```
        {
            "count": 2,
            "full_name": "Catalina Roman",
            "id": 15
        },
        {
            "count": 3,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 2,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 4,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 2,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 10,
            "full_name": "Silvia Soto",
            "id": 6
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "epics": [
        {
            "count": 25,
            "id": null,
            "order": 0,
            "ref": null,
            "subject": null
```

```
        },
        {
            "count": 0,
            "id": 30,
            "order": 2,
            "ref": 127,
            "subject": "New epic"
        },
        {
            "count": 0,
            "id": 1,
            "order": 1476435868063,
            "ref": 120,
            "subject": "Add tests for bulk operations"
        },
        {
            "count": 0,
            "id": 2,
            "order": 1476435868415,
            "ref": 121,
            "subject": "Implement the form"
        },
        {
            "count": 1,
            "id": 3,
            "order": 1476435868817,
            "ref": 122,
            "subject": "Lighttpd x-sendfile support"
        },
        {
            "count": 7,
            "id": 4,
            "order": 1476435869235,
            "ref": 123,
            "subject": "Feature/improved image admin"
        },
        {
            "count": 1,
            "id": 5,
            "order": 1476435869513,
            "ref": 124,
            "subject": "Add setting to allow regular users to create folders at the root
level."
        },
        {
            "count": 1,
            "id": 6,
            "order": 1476435869933,
```

```
            "ref": 125,
            "subject": "Added file copying and processing of images (resizing)"
        },
        {
            "count": 5,
            "id": 7,
            "order": 1476435870299,
            "ref": 126,
            "subject": "Migrate to Python 3 and milk a beautiful cow"
        },
        {
            "count": 0,
            "id": 31,
            "order": 1476436795650,
            "ref": 128,
            "subject": "New epic"
        },
        {
            "count": 0,
            "id": 32,
            "order": 1476436796275,
            "ref": 129,
            "subject": "EPIC 1"
        },
        {
            "count": 0,
            "id": 33,
            "order": 1476436796275,
            "ref": 130,
            "subject": "EPIC 2"
        },
        {
            "count": 0,
            "id": 34,
            "order": 1476436796275,
            "ref": 131,
            "subject": "EPIC 3"
        }
    ],
    "owners": [
        {
            "count": 5,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 5,
            "full_name": "Alba Leon",
```

```
            "id": 8
        },
        {
            "count": 3,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 4,
            "full_name": "Andrea Fernandez",
            "id": 14
        },
        {
            "count": 1,
            "full_name": "Catalina Roman",
            "id": 15
        },
        {
            "count": 5,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 2,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 2,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 2,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 3,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 7,
            "full_name": "Silvia Soto",
            "id": 6
        }
    ],
```

```
    "statuses": [
        {
            "color": "#999999",
            "count": 16,
            "id": 1,
            "name": "Nueva",
            "order": 1
        },
        {
            "color": "#ff8a84",
            "count": 10,
            "id": 2,
            "name": "Preparada",
            "order": 2
        },
        {
            "color": "#ff9900",
            "count": 9,
            "id": 3,
            "name": "En curso",
            "order": 3
        },
        {
            "color": "#fcc000",
            "count": 4,
            "id": 4,
            "name": "Lista para testear",
            "order": 4
        },
        {
            "color": "#669900",
            "count": 0,
            "id": 5,
            "name": "Hecha",
            "order": 5
        },
        {
            "color": "#5c3566",
            "count": 0,
            "id": 6,
            "name": "Archivada",
            "order": 6
        }
    ],
    "tags": [
        {
            "color": null,
            "count": 2,
```

```
            "name": "a"
        },
        {
            "color": null,
            "count": 0,
            "name": "ab"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusamus"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusantium"
        },
        {
            "color": null,
            "count": 1,
            "name": "ad"
        },
        {
            "color": null,
            "count": 1,
            "name": "adipisci"
        },
        {
            "color": null,
            "count": 0,
            "name": "alias"
        },
        {
            "color": "#631249",
            "count": 0,
            "name": "aliquam"
        },
        {
            "color": null,
            "count": 1,
            "name": "amet"
        },
        {
            "color": null,
            "count": 0,
            "name": "animi"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "aperiam"
        },
        {
            "color": null,
            "count": 0,
            "name": "architecto"
        },
        {
            "color": null,
            "count": 2,
            "name": "asperiores"
        },
        {
            "color": "#82854c",
            "count": 0,
            "name": "aspernatur"
        },
        {
            "color": null,
            "count": 0,
            "name": "assumenda"
        },
        {
            "color": "#27e90d",
            "count": 1,
            "name": "at"
        },
        {
            "color": null,
            "count": 0,
            "name": "atque"
        },
        {
            "color": "#9ae4e4",
            "count": 0,
            "name": "aut"
        },
        {
            "color": null,
            "count": 0,
            "name": "autem"
        },
        {
            "color": null,
            "count": 0,
            "name": "beatae"
```

```
        },
        {
            "color": null,
            "count": 0,
            "name": "blanditiis"
        },
        {
            "color": null,
            "count": 1,
            "name": "commodi"
        },
        {
            "color": null,
            "count": 0,
            "name": "consectetur"
        },
        {
            "color": null,
            "count": 0,
            "name": "consequatur"
        },
        {
            "color": null,
            "count": 1,
            "name": "consequuntur"
        },
        {
            "color": null,
            "count": 0,
            "name": "corporis"
        },
        {
            "color": "#432493",
            "count": 1,
            "name": "corrupti"
        },
        {
            "color": null,
            "count": 0,
            "name": "culpa"
        },
        {
            "color": null,
            "count": 0,
            "name": "cum"
        },
        {
            "color": "#ad75ec",
```

```
            "count": 1,
            "name": "cumque"
        },
        {
            "color": "#144bba",
            "count": 0,
            "name": "cupiditate"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": "#9631e4",
            "count": 0,
            "name": "debitis"
        },
        {
            "color": "#959608",
            "count": 0,
            "name": "delectus"
        },
        {
            "color": "#6188db",
            "count": 0,
            "name": "deleniti"
        },
        {
            "color": "#e7b695",
            "count": 1,
            "name": "deserunt"
        },
        {
            "color": null,
            "count": 1,
            "name": "dicta"
        },
        {
            "color": null,
            "count": 0,
            "name": "dignissimos"
        },
        {
            "color": "#641bd9",
            "count": 0,
            "name": "dolor"
        },
```

```
        {
            "color": "#61b076",
            "count": 0,
            "name": "dolore"
        },
        {
            "color": null,
            "count": 1,
            "name": "dolorem"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloremque"
        },
        {
            "color": "#7fea8e",
            "count": 1,
            "name": "dolores"
        },
        {
            "color": "#fb1b00",
            "count": 1,
            "name": "doloribus"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolorum"
        },
        {
            "color": "#ea6bb9",
            "count": 2,
            "name": "ducimus"
        },
        {
            "color": "#2c80b2",
            "count": 1,
            "name": "ea"
        },
        {
            "color": null,
            "count": 1,
            "name": "eaque"
        },
        {
            "color": null,
            "count": 1,
```

```
            "name": "earum"
        },
        {
            "color": "#860b86",
            "count": 1,
            "name": "eius"
        },
        {
            "color": null,
            "count": 1,
            "name": "enim"
        },
        {
            "color": "#8a6433",
            "count": 1,
            "name": "eos"
        },
        {
            "color": null,
            "count": 1,
            "name": "error"
        },
        {
            "color": null,
            "count": 0,
            "name": "esse"
        },
        {
            "color": "#665de1",
            "count": 0,
            "name": "est"
        },
        {
            "color": null,
            "count": 1,
            "name": "et"
        },
        {
            "color": "#ee6c40",
            "count": 1,
            "name": "eum"
        },
        {
            "color": null,
            "count": 1,
            "name": "eveniet"
        },
        {
```

```
            "color": "#e06613",
            "count": 0,
            "name": "ex"
        },
        {
            "color": null,
            "count": 1,
            "name": "excepturi"
        },
        {
            "color": "#ac7c74",
            "count": 0,
            "name": "exercitationem"
        },
        {
            "color": "#740c41",
            "count": 1,
            "name": "expedita"
        },
        {
            "color": "#2892cb",
            "count": 0,
            "name": "explicabo"
        },
        {
            "color": null,
            "count": 0,
            "name": "facere"
        },
        {
            "color": null,
            "count": 1,
            "name": "facilis"
        },
        {
            "color": "#e86797",
            "count": 1,
            "name": "fuga"
        },
        {
            "color": null,
            "count": 2,
            "name": "fugiat"
        },
        {
            "color": "#9345df",
            "count": 0,
            "name": "fugit"
```

```
        },
        {
            "color": "#b42d3c",
            "count": 0,
            "name": "harum"
        },
        {
            "color": null,
            "count": 1,
            "name": "hic"
        },
        {
            "color": null,
            "count": 0,
            "name": "id"
        },
        {
            "color": "#3531fd",
            "count": 1,
            "name": "illo"
        },
        {
            "color": null,
            "count": 1,
            "name": "illum"
        },
        {
            "color": null,
            "count": 1,
            "name": "impedit"
        },
        {
            "color": null,
            "count": 1,
            "name": "in"
        },
        {
            "color": "#3099ec",
            "count": 0,
            "name": "incidunt"
        },
        {
            "color": "#2fbc07",
            "count": 0,
            "name": "inventore"
        },
        {
            "color": "#ffa8ed",
```

```
            "count": 1,
            "name": "ipsa"
        },
        {
            "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": "#019320",
            "count": 0,
            "name": "iure"
        },
        {
            "color": "#3a10e8",
            "count": 0,
            "name": "iusto"
        },
        {
            "color": "#6fdf52",
            "count": 0,
            "name": "labore"
        },
        {
            "color": "#b2966d",
            "count": 0,
            "name": "laboriosam"
        },
        {
            "color": null,
            "count": 0,
            "name": "laborum"
        },
```

```
        {
            "color": "#9e3f1f",
            "count": 1,
            "name": "laudantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "libero"
        },
        {
            "color": "#d1fac1",
            "count": 0,
            "name": "magnam"
        },
        {
            "color": "#429e6f",
            "count": 0,
            "name": "magni"
        },
        {
            "color": null,
            "count": 0,
            "name": "maiores"
        },
        {
            "color": "#1acc29",
            "count": 0,
            "name": "maxime"
        },
        {
            "color": null,
            "count": 2,
            "name": "minima"
        },
        {
            "color": "#59b653",
            "count": 0,
            "name": "minus"
        },
        {
            "color": "#494e30",
            "count": 3,
            "name": "modi"
        },
        {
            "color": null,
            "count": 0,
```

```
            "name": "molestiae"
        },
        {
            "color": "#92db0b",
            "count": 0,
            "name": "molestias"
        },
        {
            "color": "#002e7f",
            "count": 1,
            "name": "mollitia"
        },
        {
            "color": null,
            "count": 0,
            "name": "nam"
        },
        {
            "color": null,
            "count": 0,
            "name": "natus"
        },
        {
            "color": "#84e3b6",
            "count": 1,
            "name": "necessitatibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "nemo"
        },
        {
            "color": null,
            "count": 0,
            "name": "neque"
        },
        {
            "color": null,
            "count": 0,
            "name": "nesciunt"
        },
        {
            "color": null,
            "count": 0,
            "name": "nihil"
        },
        {
```

```
            "color": null,
            "count": 2,
            "name": "nisi"
        },
        {
            "color": null,
            "count": 0,
            "name": "nobis"
        },
        {
            "color": null,
            "count": 0,
            "name": "non"
        },
        {
            "color": "#0cf81b",
            "count": 0,
            "name": "nostrum"
        },
        {
            "color": "#894727",
            "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": 1,
            "name": "officia"
```

```
        },
        {
            "color": "#964862",
            "count": 0,
            "name": "officiis"
        },
        {
            "color": null,
            "count": 0,
            "name": "omnis"
        },
        {
            "color": null,
            "count": 0,
            "name": "optio"
        },
        {
            "color": null,
            "count": 1,
            "name": "perspiciatis"
        },
        {
            "color": "#d97204",
            "count": 0,
            "name": "placeat"
        },
        {
            "color": null,
            "count": 0,
            "name": "porro"
        },
        {
            "color": "#fccc1b",
            "count": 0,
            "name": "possimus"
        },
        {
            "color": "#7fdcf2",
            "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": "#5dae16",
            "count": 0,
            "name": "quasi"
        },
        {
            "color": null,
            "count": 0,
            "name": "qui"
        },
        {
            "color": null,
            "count": 0,
            "name": "quia"
        },
        {
            "color": null,
            "count": 1,
            "name": "quibusdam"
        },
        {
            "color": "#ae6519",
            "count": 0,
            "name": "quidem"
        },
        {
            "color": null,
            "count": 1,
            "name": "quis"
        },
        {
            "color": null,
            "count": 0,
            "name": "quisquam"
        },
```

```
        {
            "color": "#857670",
            "count": 0,
            "name": "quo"
        },
        {
            "color": "#0e5b24",
            "count": 0,
            "name": "quod"
        },
        {
            "color": null,
            "count": 0,
            "name": "quos"
        },
        {
            "color": null,
            "count": 0,
            "name": "ratione"
        },
        {
            "color": "#560ff6",
            "count": 2,
            "name": "reiciendis"
        },
        {
            "color": "#688119",
            "count": 0,
            "name": "rem"
        },
        {
            "color": null,
            "count": 1,
            "name": "repellat"
        },
        {
            "color": null,
            "count": 0,
            "name": "repellendus"
        },
        {
            "color": null,
            "count": 1,
            "name": "reprehenderit"
        },
        {
            "color": "#3a2b71",
            "count": 0,
```

```
            "name": "repudiandae"
        },
        {
            "color": null,
            "count": 1,
            "name": "rerum"
        },
        {
            "color": "#850c56",
            "count": 0,
            "name": "sapiente"
        },
        {
            "color": null,
            "count": 0,
            "name": "sed"
        },
        {
            "color": "#9f6274",
            "count": 0,
            "name": "sequi"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
        {
            "color": null,
            "count": 0,
            "name": "similique"
        },
        {
            "color": null,
            "count": 0,
            "name": "sint"
        },
        {
            "color": "#abdcde",
            "count": 0,
            "name": "sit"
        },
        {
            "color": null,
            "count": 0,
            "name": "suscipit"
        },
        {
```

```
            "color": null,
            "count": 0,
            "name": "tempora"
        },
        {
            "color": null,
            "count": 1,
            "name": "tempore"
        },
        {
            "color": "#a2c51a",
            "count": 1,
            "name": "temporibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "tenetur"
        },
        {
            "color": "#560a5d",
            "count": 1,
            "name": "totam"
        },
        {
            "color": "#98ad13",
            "count": 1,
            "name": "ullam"
        },
        {
            "color": "#da2470",
            "count": 0,
            "name": "unde"
        },
        {
            "color": "#e74669",
            "count": 0,
            "name": "ut"
        },
        {
            "color": null,
            "count": 0,
            "name": "vel"
        },
        {
            "color": null,
            "count": 0,
            "name": "velit"
```

```
        },
        {
            "color": null,
            "count": 0,
            "name": "veniam"
        },
        {
            "color": "#768459",
            "count": 0,
            "name": "veritatis"
        },
        {
            "color": null,
            "count": 0,
            "name": "vero"
        },
        {
            "color": "#d9fe5e",
            "count": 0,
            "name": "vitae"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptas"
        },
        {
            "color": "#b0eff0",
            "count": 0,
            "name": "voluptate"
        },
        {
            "color": "#00d60c",
            "count": 1,
            "name": "voluptatem"
        },
        {
            "color": "#681ad4",
            "count": 0,
            "name": "voluptatibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptatum"
        }
    ]
}
```

45.27. User story voter detail

inlcude::generated/user-stories-get-voters-output.adoc[]

45.28. User story watcher detail

inlcude::generated/user-stories-get-watchers-output.adoc[]

45.29. 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
}
```

45.30. Point detail

```
{
    "id": 1,
    "name": "Patch name",
    "order": 1,
    "project": 1,
    "value": null
}
```

45.31. User story custom attribute detail

```
{
    "created_date": "2016-10-14T09:03:28.421Z",
    "description": "accusamus sit corporis ipsum",
    "id": 1,
    "modified_date": "2016-10-14T09:20:05.054Z",
    "name": "Duration 1",
    "order": 1,
    "project": 1,
    "type": "date"
}
```


45.32. User story custom attributes values detail


```
{
    "attributes_values": {
        "5": "240 min"
    },
    "user_story": 1,
    "version": 2
}
```


45.33. Task detail


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Catalina Roman",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2016-10-14T09:03:29.043Z",
    "description": "Perspiciatis ipsum repellat quia quidem officia, totam accusamus
laboriosam, quas expedita quos dolore adipisci animi harum hic?",
    "description_html": "<p>Perspiciatis ipsum repellat quia quidem officia, totam
accusamus laboriosam, quas expedita quos dolore adipisci animi harum hic?</p>",
```

```
    "external_reference": null,
    "finished_date": "2016-08-29T23:17:07.508Z",
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": true,
    "is_watcher": false,
    "milestone": 1,
    "milestone_slug": "sprint-2016-8-20",
    "modified_date": "2016-10-14T09:20:35.510Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 3,
            "subject": "Add tests for bulk operations"
        },
        "previous": null
    },
    "owner": 12,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Pilar Herrera",
        "gravatar_id": "74cb769a5e64d445b8550789e1553502",
        "id": 12,
        "is_active": true,
        "photo": null,
        "username": "user6"
    },
    "project": 1,
    "ref": 2,
    "status": 4,
    "status_extra_info": {
        "color": "#669900",
        "is_closed": true,
        "name": "Cerrada"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "iste",
            null
        ],
        [
            "obcaecati",
            null
        ],
        [
```

```
            "vel",
            null
        ],
        [
            "totam",
            null
        ],
        [
            "modi",
            null
        ],
        [
            "eos",
            null
        ],
        [
            "deleniti",
            null
        ]
    ],
    "taskboard_order": 1476435809043,
    "total_voters": 7,
    "total_watchers": 0,
    "us_order": 1476435809043,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": null,
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": []
}
```


45.34. Task detail (GET)


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Catalina Roman",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
```

```
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2016-10-14T09:03:29.043Z",
    "description": "Perspiciatis ipsum repellat quia quidem officia, totam accusamus
laboriosam, quas expedita quos dolore adipisci animi harum hic?",
    "description_html": "<p>Perspiciatis ipsum repellat quia quidem officia, totam
accusamus laboriosam, quas expedita quos dolore adipisci animi harum hic?</p>",
    "external_reference": null,
    "finished_date": "2016-08-29T23:17:07.508Z",
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": true,
    "is_watcher": false,
    "milestone": 1,
    "milestone_slug": "sprint-2016-8-20",
    "modified_date": "2016-10-14T09:20:35.510Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 3,
            "subject": "Add tests for bulk operations"
        },
        "previous": null
    },
    "owner": 12,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Pilar Herrera",
        "gravatar_id": "74cb769a5e64d445b8550789e1553502",
        "id": 12,
        "is_active": true,
        "photo": null,
        "username": "user6"
    },
    "project": 1,
    "ref": 2,
    "status": 4,
    "status_extra_info": {
        "color": "#669900",
        "is_closed": true,
        "name": "Cerrada"
    },
```

```
    "subject": "Patching subject",
    "tags": [
        [
            "iste",
            null
        ],
        [
            "obcaecati",
            null
        ],
        [
            "vel",
            null
        ],
        [
            "totam",
            null
        ],
        [
            "modi",
            null
        ],
        [
            "eos",
            null
        ],
        [
            "deleniti",
            null
        ]
    ],
    "taskboard_order": 1476435809043,
    "total_voters": 7,
    "total_watchers": 0,
    "us_order": 1476435809043,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": null,
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": []
}
```

45.35. Task detail (LIST)


```
{
    "assigned_to": 15,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Catalina Roman",
        "gravatar_id": "69b60d39a450e863609ae3546b12b360",
        "id": 15,
        "is_active": true,
        "photo": null,
        "username": "user9"
    },
    "attachments": [],
    "blocked_note": "",
    "created_date": "2016-10-14T09:03:29.043Z",
    "external_reference": null,
    "finished_date": "2016-08-29T23:17:07.508Z",
    "id": 1,
    "is_blocked": false,
    "is_closed": true,
    "is_iocaine": false,
    "is_voter": true,
    "is_watcher": true,
    "milestone": 1,
    "milestone_slug": "sprint-2016-8-20",
    "modified_date": "2016-10-14T09:20:35.510Z",
    "owner": 12,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Pilar Herrera",
        "gravatar_id": "74cb769a5e64d445b8550789e1553502",
        "id": 12,
        "is_active": true,
        "photo": null,
        "username": "user6"
    },
    "project": 1,
    "ref": 2,
    "status": 4,
    "status_extra_info": {
        "color": "#669900",
        "is_closed": true,
        "name": "Cerrada"
    },
    "subject": "Patching subject",
```

```
    "tags": [
        [
            "iste",
            null
        ],
        [
            "obcaecati",
            null
        ],
        [
            "vel",
            null
        ],
        [
            "totam",
            null
        ],
        [
            "modi",
            null
        ],
        [
            "eos",
            null
        ],
        [
            "deleniti",
            null
        ]
    ],
    "taskboard_order": 1476435809043,
    "total_voters": 7,
    "total_watchers": 1,
    "us_order": 1476435809043,
    "user_story": 1,
    "user_story_extra_info": {
        "epics": null,
        "id": 1,
        "ref": 1,
        "subject": "Patching subject"
    },
    "version": 2,
    "watchers": [
        6
    ]
}
```

45.36. Task filters data detail


```
{
    "assigned_to": [
        {
            "count": 5,
            "full_name": "",
            "id": null
        },
        {
            "count": 6,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 1,
            "full_name": "Alba Leon",
            "id": 8
        },
        {
            "count": 2,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 7,
            "full_name": "Andrea Fernandez",
            "id": 14
        },
        {
            "count": 11,
            "full_name": "Catalina Roman",
            "id": 15
        },
        {
            "count": 5,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 2,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 12,
```

```
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 5,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 5,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 6,
            "full_name": "Silvia Soto",
            "id": 6
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "owners": [
        {
            "count": 3,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 5,
            "full_name": "Alba Leon",
            "id": 8
        },
        {
            "count": 6,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 4,
            "full_name": "Andrea Fernandez",
            "id": 14
        },
        {
            "count": 6,
            "full_name": "Catalina Roman",
```

```
            "id": 15
        },
        {
            "count": 7,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 10,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 3,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 3,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 11,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 9,
            "full_name": "Silvia Soto",
            "id": 6
        }
    ],
    "statuses": [
        {
            "color": "#ffcc00",
            "count": 9,
            "id": 3,
            "name": "Lista para testear",
            "order": 3
        },
        {
            "color": "#669900",
            "count": 12,
            "id": 4,
            "name": "Cerrada",
            "order": 4
        },
```

```
        {
            "color": "#ff9900",
            "count": 14,
            "id": 2,
            "name": "En curso",
            "order": 5
        },
        {
            "color": "#999999",
            "count": 14,
            "id": 5,
            "name": "Necesita informaci\u00f3n",
            "order": 5
        },
        {
            "color": "#AAAAAA",
            "count": 0,
            "id": 41,
            "name": "New status",
            "order": 8
        },
        {
            "color": "#999999",
            "count": 14,
            "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": "cumque"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
```

```
            "color": null,
            "count": 3,
            "name": "deleniti"
        },
        {
            "color": null,
            "count": 3,
            "name": "eos"
        },
        {
            "color": null,
            "count": 4,
            "name": "expedita"
        },
        {
            "color": null,
            "count": 3,
            "name": "iste"
        },
        {
            "color": null,
            "count": 11,
            "name": "modi"
        },
        {
            "color": null,
            "count": 3,
            "name": "obcaecati"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
        {
            "color": null,
            "count": 1,
            "name": "totam"
        },
        {
            "color": null,
            "count": 3,
            "name": "vel"
        }
    ]
}
```

45.37. Task voter detail

```
{
    "full_name": "Administrator",
    "id": 5,
    "username": "admin"
}
```

45.38. Task watcher detail

```
{
    "full_name": "Silvia Soto",
    "id": 6,
    "username": "user6532909695705815086"
}
```

45.39. Task status detail

```
{
    "color": "#999999",
    "id": 1,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name"
}
```

45.40. Task custom attribute detail

```
{
    "created_date": "2016-10-14T09:03:28.476Z",
    "description": "nulla laborum autem impedit deserunt delectus",
    "id": 5,
    "modified_date": "2016-10-14T09:20:38.934Z",
    "name": "Duration 1",
    "order": 1,
    "project": 1,
    "type": "url"
}
```


45.41. Task custom attributes values detail


```
{
    "attributes_values": {
        "5": "240 min"
    },
    "task": 1,
    "version": 2
}
```


45.42. Issue detail


```
{
    "assigned_to": 13,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Alvaro Molina",
        "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
        "id": 13,
        "is_active": true,
        "photo": null,
        "username": "user7"
    },
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2016-10-14T09:04:14.084Z",
    "description": "Sint temporibus unde tempora exercitationem ullam praesentium,
consequatur veniam voluptas ut? Nihil perspiciatis iusto quas reprehenderit repellat
omnis sequi dolore tempora nisi exercitationem, molestias enim aut recusandae, amet
repellat eum ullam qui nostrum sint deserunt voluptate nulla exercitationem
reprehenderit, mollitia id sed natus?",
```

```
    "description_html": "<p>Sint temporibus unde tempora exercitationem ullam
praesentium, consequatur veniam voluptas ut? Nihil perspiciatis iusto quas reprehenderit
repellat omnis sequi dolore tempora nisi exercitationem, molestias enim aut recusandae,
amet repellat eum ullam qui nostrum sint deserunt voluptate nulla exercitationem
reprehenderit, mollitia id sed natus?</p>",
    "external_reference": null,
    "finished_date": "2016-10-14T09:04:14.085Z",
    "generated_user_stories": null,
    "id": 3,
    "is_blocked": false,
    "is_closed": true,
    "is_voter": false,
    "is_watcher": false,
    "milestone": null,
    "modified_date": "2016-10-14T09:20:21.710Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 98,
            "subject": "Implement the form"
        },
        "previous": {
            "id": 4,
            "ref": 100,
            "subject": "Create testsuite with matrix builds"
        }
    },
    "owner": 5,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Administrator",
        "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
        "id": 5,
        "is_active": true,
        "photo": null,
        "username": "admin"
    },
    "priority": 1,
    "project": 1,
    "ref": 99,
    "severity": 2,
    "status": 6,
    "status_extra_info": {
        "color": "#CC0000",
        "is_closed": true,
        "name": "Rechazada"
    },
    "subject": "Patching subject",
```

```
    "tags": [
        [
            "magni",
            "#429e6f"
        ],
        [
            "minus",
            "#59b653"
        ],
        [
            "provident",
            "#7fdcf2"
        ],
        [
            "quia",
            null
        ],
        [
            "repellat",
            null
        ],
        [
            "quisquam",
            null
        ],
        [
            "temporibus",
            "#a2c51a"
        ],
        [
            "eos",
            "#8a6433"
        ]
    ],
    "total_voters": 3,
    "total_watchers": 3,
    "type": 3,
    "version": 2,
    "watchers": [
        5,
        13,
        15
    ]
}
```

45.43. Issue detail (GET)


```
{
    "assigned_to": 13,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Alvaro Molina",
        "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
        "id": 13,
        "is_active": true,
        "photo": null,
        "username": "user7"
    },
    "blocked_note": "",
    "blocked_note_html": "",
    "comment": "",
    "created_date": "2016-10-14T09:04:14.084Z",
    "description": "Sint temporibus unde tempora exercitationem ullam praesentium,
consequatur veniam voluptas ut? Nihil perspiciatis iusto quas reprehenderit repellat
omnis sequi dolore tempora nisi exercitationem, molestias enim aut recusandae, amet
repellat eum ullam qui nostrum sint deserunt voluptate nulla exercitationem
reprehenderit, mollitia id sed natus?",
    "description_html": "<p>Sint temporibus unde tempora exercitationem ullam
praesentium, consequatur veniam voluptas ut? Nihil perspiciatis iusto quas reprehenderit
repellat omnis sequi dolore tempora nisi exercitationem, molestias enim aut recusandae,
amet repellat eum ullam qui nostrum sint deserunt voluptate nulla exercitationem
reprehenderit, mollitia id sed natus?</p>",
    "external_reference": null,
    "finished_date": "2016-10-14T09:04:14.085Z",
    "generated_user_stories": null,
    "id": 3,
    "is_blocked": false,
    "is_closed": true,
    "is_voter": false,
    "is_watcher": false,
    "milestone": null,
    "modified_date": "2016-10-14T09:20:21.710Z",
    "neighbors": {
        "next": {
            "id": 2,
            "ref": 98,
            "subject": "Implement the form"
        },
        "previous": {
            "id": 4,
            "ref": 100,
```

```
            "subject": "Create testsuite with matrix builds"
        }
    },
    "owner": 5,
    "owner_extra_info": {
        "big_photo": null,
        "full_name_display": "Administrator",
        "gravatar_id": "64e1b8d34f425d19e1ee2ea7236d3028",
        "id": 5,
        "is_active": true,
        "photo": null,
        "username": "admin"
    },
    "priority": 1,
    "project": 1,
    "ref": 99,
    "severity": 2,
    "status": 6,
    "status_extra_info": {
        "color": "#CC0000",
        "is_closed": true,
        "name": "Rechazada"
    },
    "subject": "Patching subject",
    "tags": [
        [
            "magni",
            "#429e6f"
        ],
        [
            "minus",
            "#59b653"
        ],
        [
            "provident",
            "#7fdcf2"
        ],
        [
            "quia",
            null
        ],
        [
            "repellat",
            null
        ],
        [
            "quisquam",
            null
```

```
        ],
        [
            "temporibus",
            "#a2c51a"
        ],
        [
            "eos",
            "#8a6433"
        ]
    ],
    "total_voters": 3,
    "total_watchers": 3,
    "type": 3,
    "version": 2,
    "watchers": [
        5,
        13,
        15
    ]
}
```


45.44. Issue detail (LIST)


```
{
    "assigned_to": null,
    "assigned_to_extra_info": null,
    "blocked_note": "",
    "created_date": "2016-08-04T11:08:19Z",
    "external_reference": null,
    "finished_date": null,
    "id": 138,
    "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": "Alvaro Molina",
        "gravatar_id": "6d7e702bd6c6fc568fca7577f9ca8c55",
        "id": 13,
        "is_active": true,
        "photo": null,
        "username": "user7"
```

```
    },
    "priority": 23,
    "project": 8,
    "ref": 59,
    "severity": 38,
    "status": 58,
    "status_extra_info": {
        "color": "#666666",
        "is_closed": false,
        "name": "Pospuesta"
    },
    "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
    ]
}
```


45.45. Issue filters data detail


```
{
    "assigned_to": [
        {
            "count": 8,
            "full_name": "",
            "id": null
        },
        {
            "count": 1,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 3,
            "full_name": "Alba Leon",
            "id": 8
        },
        {
            "count": 3,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 2,
            "full_name": "Andrea Fernandez",
            "id": 14
        },
        {
            "count": 3,
```

```
            "full_name": "Catalina Roman",
            "id": 15
        },
        {
            "count": 1,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 1,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 2,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 1,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 2,
            "full_name": "Silvia Soto",
            "id": 6
        },
        {
            "count": 0,
            "full_name": "test",
            "id": 16
        }
    ],
    "owners": [
        {
            "count": 3,
            "full_name": "Administrator",
            "id": 5
        },
        {
            "count": 3,
            "full_name": "Alba Leon",
```

```
            "id": 8
        },
        {
            "count": 3,
            "full_name": "Alvaro Molina",
            "id": 13
        },
        {
            "count": 1,
            "full_name": "Andrea Fernandez",
            "id": 14
        },
        {
            "count": 2,
            "full_name": "Catalina Roman",
            "id": 15
        },
        {
            "count": 2,
            "full_name": "Esther Ferrer",
            "id": 9
        },
        {
            "count": 1,
            "full_name": "German Benitez",
            "id": 11
        },
        {
            "count": 1,
            "full_name": "Marcos Ortiz",
            "id": 7
        },
        {
            "count": 1,
            "full_name": "Marta Carmona",
            "id": 10
        },
        {
            "count": 2,
            "full_name": "Pilar Herrera",
            "id": 12
        },
        {
            "count": 9,
            "full_name": "Silvia Soto",
            "id": 6
        }
    ],
```

```
    "priorities": [
        {
            "color": "#666666",
            "count": 13,
            "id": 1,
            "name": "Baja",
            "order": 1
        },
        {
            "color": "#669933",
            "count": 10,
            "id": 2,
            "name": "Normal",
            "order": 3
        },
        {
            "color": "#CC0000",
            "count": 5,
            "id": 3,
            "name": "Alta",
            "order": 5
        }
    ],
    "severities": [
        {
            "color": "#0000FF",
            "count": 7,
            "id": 3,
            "name": "Normal",
            "order": 3
        },
        {
            "color": "#FFA500",
            "count": 2,
            "id": 4,
            "name": "Importante",
            "order": 4
        },
        {
            "color": "#669933",
            "count": 7,
            "id": 2,
            "name": "Menor",
            "order": 5
        },
        {
            "color": "#CC0000",
            "count": 7,
```

```
            "id": 5,
            "name": "Cr\u00edtica",
            "order": 5
        },
        {
            "color": "#AAAAAA",
            "count": 0,
            "id": 41,
            "name": "New severity",
            "order": 8
        },
        {
            "color": "#666666",
            "count": 5,
            "id": 1,
            "name": "Patch name",
            "order": 10
        },
        {
            "color": "#999999",
            "count": 0,
            "id": 42,
            "name": "New severity name",
            "order": 10
        }
    ],
    "statuses": [
        {
            "color": "#88A65E",
            "count": 2,
            "id": 3,
            "name": "Lista para testear",
            "order": 3
        },
        {
            "color": "#BFB35A",
            "count": 6,
            "id": 4,
            "name": "Cerrada",
            "order": 4
        },
        {
            "color": "#5E8C6A",
            "count": 2,
            "id": 2,
            "name": "En curso",
            "order": 5
        },
```

```
        {
            "color": "#89BAB4",
            "count": 2,
            "id": 5,
            "name": "Necesita informaci\u00f3n",
            "order": 5
        },
        {
            "color": "#CC0000",
            "count": 8,
            "id": 6,
            "name": "Rechazada",
            "order": 6
        },
        {
            "color": "#666666",
            "count": 2,
            "id": 7,
            "name": "Pospuesta",
            "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": 6,
            "id": 1,
            "name": "Patch status name",
            "order": 10
        }
    ],
    "tags": [
        {
            "color": null,
            "count": 1,
            "name": "a"
```

```
        },
        {
            "color": null,
            "count": 3,
            "name": "ab"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusamus"
        },
        {
            "color": null,
            "count": 0,
            "name": "accusantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "ad"
        },
        {
            "color": null,
            "count": 1,
            "name": "adipisci"
        },
        {
            "color": null,
            "count": 0,
            "name": "alias"
        },
        {
            "color": "#631249",
            "count": 0,
            "name": "aliquam"
        },
        {
            "color": null,
            "count": 1,
            "name": "amet"
        },
        {
            "color": null,
            "count": 0,
            "name": "animi"
        },
        {
            "color": null,
```

```
            "count": 3,
            "name": "aperiam"
        },
        {
            "color": null,
            "count": 0,
            "name": "architecto"
        },
        {
            "color": null,
            "count": 1,
            "name": "asperiores"
        },
        {
            "color": "#82854c",
            "count": 1,
            "name": "aspernatur"
        },
        {
            "color": null,
            "count": 1,
            "name": "assumenda"
        },
        {
            "color": "#27e90d",
            "count": 1,
            "name": "at"
        },
        {
            "color": null,
            "count": 1,
            "name": "atque"
        },
        {
            "color": "#9ae4e4",
            "count": 0,
            "name": "aut"
        },
        {
            "color": null,
            "count": 0,
            "name": "autem"
        },
        {
            "color": null,
            "count": 2,
            "name": "beatae"
        },
```

```
        {
            "color": null,
            "count": 0,
            "name": "blanditiis"
        },
        {
            "color": null,
            "count": 0,
            "name": "commodi"
        },
        {
            "color": null,
            "count": 1,
            "name": "consectetur"
        },
        {
            "color": null,
            "count": 0,
            "name": "consequatur"
        },
        {
            "color": null,
            "count": 2,
            "name": "consequuntur"
        },
        {
            "color": null,
            "count": 0,
            "name": "corporis"
        },
        {
            "color": "#432493",
            "count": 2,
            "name": "corrupti"
        },
        {
            "color": null,
            "count": 1,
            "name": "culpa"
        },
        {
            "color": null,
            "count": 0,
            "name": "cum"
        },
        {
            "color": "#ad75ec",
            "count": 0,
```

```
            "name": "cumque"
        },
        {
            "color": "#144bba",
            "count": 1,
            "name": "cupiditate"
        },
        {
            "color": null,
            "count": 1,
            "name": "customer"
        },
        {
            "color": "#9631e4",
            "count": 0,
            "name": "debitis"
        },
        {
            "color": "#959608",
            "count": 1,
            "name": "delectus"
        },
        {
            "color": "#6188db",
            "count": 1,
            "name": "deleniti"
        },
        {
            "color": "#e7b695",
            "count": 0,
            "name": "deserunt"
        },
        {
            "color": null,
            "count": 1,
            "name": "dicta"
        },
        {
            "color": null,
            "count": 0,
            "name": "dignissimos"
        },
        {
            "color": "#641bd9",
            "count": 0,
            "name": "dolor"
        },
        {
```

```
            "color": "#61b076",
            "count": 0,
            "name": "dolore"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolorem"
        },
        {
            "color": null,
            "count": 0,
            "name": "doloremque"
        },
        {
            "color": "#7fea8e",
            "count": 0,
            "name": "dolores"
        },
        {
            "color": "#fb1b00",
            "count": 1,
            "name": "doloribus"
        },
        {
            "color": null,
            "count": 0,
            "name": "dolorum"
        },
        {
            "color": "#ea6bb9",
            "count": 1,
            "name": "ducimus"
        },
        {
            "color": "#2c80b2",
            "count": 0,
            "name": "ea"
        },
        {
            "color": null,
            "count": 0,
            "name": "eaque"
        },
        {
            "color": null,
            "count": 5,
            "name": "earum"
```

```
        },
        {
            "color": "#860b86",
            "count": 0,
            "name": "eius"
        },
        {
            "color": null,
            "count": 0,
            "name": "enim"
        },
        {
            "color": "#8a6433",
            "count": 2,
            "name": "eos"
        },
        {
            "color": null,
            "count": 0,
            "name": "error"
        },
        {
            "color": null,
            "count": 1,
            "name": "esse"
        },
        {
            "color": "#665de1",
            "count": 0,
            "name": "est"
        },
        {
            "color": null,
            "count": 1,
            "name": "et"
        },
        {
            "color": "#ee6c40",
            "count": 0,
            "name": "eum"
        },
        {
            "color": null,
            "count": 1,
            "name": "eveniet"
        },
        {
            "color": "#e06613",
```

```
            "count": 1,
            "name": "ex"
        },
        {
            "color": null,
            "count": 0,
            "name": "excepturi"
        },
        {
            "color": "#ac7c74",
            "count": 0,
            "name": "exercitationem"
        },
        {
            "color": "#740c41",
            "count": 0,
            "name": "expedita"
        },
        {
            "color": "#2892cb",
            "count": 0,
            "name": "explicabo"
        },
        {
            "color": null,
            "count": 0,
            "name": "facere"
        },
        {
            "color": null,
            "count": 1,
            "name": "facilis"
        },
        {
            "color": "#e86797",
            "count": 2,
            "name": "fuga"
        },
        {
            "color": null,
            "count": 0,
            "name": "fugiat"
        },
        {
            "color": "#9345df",
            "count": 0,
            "name": "fugit"
        },
```

```
        {
            "color": "#b42d3c",
            "count": 1,
            "name": "harum"
        },
        {
            "color": null,
            "count": 0,
            "name": "hic"
        },
        {
            "color": null,
            "count": 0,
            "name": "id"
        },
        {
            "color": "#3531fd",
            "count": 0,
            "name": "illo"
        },
        {
            "color": null,
            "count": 0,
            "name": "illum"
        },
        {
            "color": null,
            "count": 0,
            "name": "impedit"
        },
        {
            "color": null,
            "count": 0,
            "name": "in"
        },
        {
            "color": "#3099ec",
            "count": 0,
            "name": "incidunt"
        },
        {
            "color": "#2fbc07",
            "count": 2,
            "name": "inventore"
        },
        {
            "color": "#ffa8ed",
            "count": 0,
```

```
            "name": "ipsa"
        },
        {
            "color": null,
            "count": 1,
            "name": "ipsam"
        },
        {
            "color": "#da3ba4",
            "count": 1,
            "name": "ipsum"
        },
        {
            "color": "#491b3a",
            "count": 3,
            "name": "iste"
        },
        {
            "color": null,
            "count": 1,
            "name": "itaque"
        },
        {
            "color": "#019320",
            "count": 2,
            "name": "iure"
        },
        {
            "color": "#3a10e8",
            "count": 0,
            "name": "iusto"
        },
        {
            "color": "#6fdf52",
            "count": 2,
            "name": "labore"
        },
        {
            "color": "#b2966d",
            "count": 1,
            "name": "laboriosam"
        },
        {
            "color": null,
            "count": 1,
            "name": "laborum"
        },
        {
```

```
            "color": "#9e3f1f",
            "count": 2,
            "name": "laudantium"
        },
        {
            "color": null,
            "count": 0,
            "name": "libero"
        },
        {
            "color": "#d1fac1",
            "count": 0,
            "name": "magnam"
        },
        {
            "color": "#429e6f",
            "count": 1,
            "name": "magni"
        },
        {
            "color": null,
            "count": 0,
            "name": "maiores"
        },
        {
            "color": "#1acc29",
            "count": 1,
            "name": "maxime"
        },
        {
            "color": null,
            "count": 0,
            "name": "minima"
        },
        {
            "color": "#59b653",
            "count": 1,
            "name": "minus"
        },
        {
            "color": "#494e30",
            "count": 5,
            "name": "modi"
        },
        {
            "color": null,
            "count": 0,
            "name": "molestiae"
```

```
        },
        {
            "color": "#92db0b",
            "count": 2,
            "name": "molestias"
        },
        {
            "color": "#002e7f",
            "count": 1,
            "name": "mollitia"
        },
        {
            "color": null,
            "count": 2,
            "name": "nam"
        },
        {
            "color": null,
            "count": 1,
            "name": "natus"
        },
        {
            "color": "#84e3b6",
            "count": 3,
            "name": "necessitatibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "nemo"
        },
        {
            "color": null,
            "count": 1,
            "name": "neque"
        },
        {
            "color": null,
            "count": 0,
            "name": "nesciunt"
        },
        {
            "color": null,
            "count": 0,
            "name": "nihil"
        },
        {
            "color": null,
```

```
            "count": 0,
            "name": "nisi"
        },
        {
            "color": null,
            "count": 1,
            "name": "nobis"
        },
        {
            "color": null,
            "count": 0,
            "name": "non"
        },
        {
            "color": "#0cf81b",
            "count": 0,
            "name": "nostrum"
        },
        {
            "color": "#894727",
            "count": 1,
            "name": "nulla"
        },
        {
            "color": null,
            "count": 0,
            "name": "numquam"
        },
        {
            "color": null,
            "count": 0,
            "name": "obcaecati"
        },
        {
            "color": null,
            "count": 1,
            "name": "odio"
        },
        {
            "color": null,
            "count": 0,
            "name": "odit"
        },
        {
            "color": "#c4f027",
            "count": 1,
            "name": "officia"
        },
```

```
        {
            "color": "#964862",
            "count": 0,
            "name": "officiis"
        },
        {
            "color": null,
            "count": 0,
            "name": "omnis"
        },
        {
            "color": null,
            "count": 1,
            "name": "optio"
        },
        {
            "color": null,
            "count": 1,
            "name": "perspiciatis"
        },
        {
            "color": "#d97204",
            "count": 0,
            "name": "placeat"
        },
        {
            "color": null,
            "count": 0,
            "name": "porro"
        },
        {
            "color": "#fccc1b",
            "count": 0,
            "name": "possimus"
        },
        {
            "color": "#7fdcf2",
            "count": 1,
            "name": "provident"
        },
        {
            "color": "#d91a8b",
            "count": 1,
            "name": "quae"
        },
        {
            "color": "#0b4425",
            "count": 0,
```

```
            "name": "quaerat"
        },
        {
            "color": null,
            "count": 2,
            "name": "quam"
        },
        {
            "color": "#6e3390",
            "count": 0,
            "name": "quas"
        },
        {
            "color": "#5dae16",
            "count": 2,
            "name": "quasi"
        },
        {
            "color": null,
            "count": 0,
            "name": "qui"
        },
        {
            "color": null,
            "count": 1,
            "name": "quia"
        },
        {
            "color": null,
            "count": 2,
            "name": "quibusdam"
        },
        {
            "color": "#ae6519",
            "count": 0,
            "name": "quidem"
        },
        {
            "color": null,
            "count": 0,
            "name": "quis"
        },
        {
            "color": null,
            "count": 2,
            "name": "quisquam"
        },
        {
```

```
            "color": "#857670",
            "count": 1,
            "name": "quo"
        },
        {
            "color": "#0e5b24",
            "count": 1,
            "name": "quod"
        },
        {
            "color": null,
            "count": 0,
            "name": "quos"
        },
        {
            "color": null,
            "count": 1,
            "name": "ratione"
        },
        {
            "color": "#560ff6",
            "count": 0,
            "name": "reiciendis"
        },
        {
            "color": "#688119",
            "count": 1,
            "name": "rem"
        },
        {
            "color": null,
            "count": 2,
            "name": "repellat"
        },
        {
            "color": null,
            "count": 0,
            "name": "repellendus"
        },
        {
            "color": null,
            "count": 1,
            "name": "reprehenderit"
        },
        {
            "color": "#3a2b71",
            "count": 1,
            "name": "repudiandae"
```

```
        },
        {
            "color": null,
            "count": 1,
            "name": "rerum"
        },
        {
            "color": "#850c56",
            "count": 1,
            "name": "sapiente"
        },
        {
            "color": null,
            "count": 1,
            "name": "sed"
        },
        {
            "color": "#9f6274",
            "count": 0,
            "name": "sequi"
        },
        {
            "color": null,
            "count": 1,
            "name": "service catalog"
        },
        {
            "color": null,
            "count": 1,
            "name": "similique"
        },
        {
            "color": null,
            "count": 1,
            "name": "sint"
        },
        {
            "color": "#abdcde",
            "count": 0,
            "name": "sit"
        },
        {
            "color": null,
            "count": 1,
            "name": "suscipit"
        },
        {
            "color": null,
```

```
            "count": 0,
            "name": "tempora"
        },
        {
            "color": null,
            "count": 0,
            "name": "tempore"
        },
        {
            "color": "#a2c51a",
            "count": 1,
            "name": "temporibus"
        },
        {
            "color": null,
            "count": 1,
            "name": "tenetur"
        },
        {
            "color": "#560a5d",
            "count": 1,
            "name": "totam"
        },
        {
            "color": "#98ad13",
            "count": 0,
            "name": "ullam"
        },
        {
            "color": "#da2470",
            "count": 0,
            "name": "unde"
        },
        {
            "color": "#e74669",
            "count": 1,
            "name": "ut"
        },
        {
            "color": null,
            "count": 0,
            "name": "vel"
        },
        {
            "color": null,
            "count": 0,
            "name": "velit"
        },
```

```
        {
            "color": null,
            "count": 2,
            "name": "veniam"
        },
        {
            "color": "#768459",
            "count": 1,
            "name": "veritatis"
        },
        {
            "color": null,
            "count": 3,
            "name": "vero"
        },
        {
            "color": "#d9fe5e",
            "count": 0,
            "name": "vitae"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptas"
        },
        {
            "color": "#b0eff0",
            "count": 0,
            "name": "voluptate"
        },
        {
            "color": "#00d60c",
            "count": 1,
            "name": "voluptatem"
        },
        {
            "color": "#681ad4",
            "count": 1,
            "name": "voluptatibus"
        },
        {
            "color": null,
            "count": 0,
            "name": "voluptatum"
        }
    ],
    "types": [
        {
```

```
            "color": "#89BAB4",
            "count": 13,
            "id": 1,
            "name": "Bug",
            "order": 1
        },
        {
            "color": "#ba89a8",
            "count": 7,
            "id": 2,
            "name": "Pregunta",
            "order": 2
        },
        {
            "color": "#89a8ba",
            "count": 8,
            "id": 3,
            "name": "Mejora",
            "order": 3
        }
    ]
}
```


45.46. Issue voters detail


```
{
    "full_name": "Gogs",
    "id": 4,
    "username": "gogs-f33ddbd6ecc64da1b18d5b9bd902d2b6"
}
```


45.47. Issue watchers detail


```
{
    "full_name": "Administrator",
    "id": 5,
    "username": "admin"
}
```


45.48. Issue status detail

```
{
    "color": "#8C2318",
    "id": 1,
    "is_closed": false,
    "name": "Patch status name",
    "order": 1,
    "project": 1,
    "slug": "patch-status-name"
}
```

45.49. Issue type detail

```
{
    "color": "#89BAB4",
    "id": 1,
    "name": "Patch type name",
    "order": 1,
    "project": 1
}
```

45.50. Priority detail

```
{
    "color": "#666666",
    "id": 1,
    "name": "Patch name",
    "order": 1,
    "project": 1
}
```

45.51. Severity detail

```
{
    "color": "#666666",
    "id": 1,
    "name": "Patch name",
    "order": 1,
    "project": 1
}
```

45.52. Issue custom attribute detail


```
{
    "created_date": "2016-10-14T09:03:28.505Z",
    "description": "ad temporibus maiores",
    "id": 5,
    "modified_date": "2016-10-14T09:20:33.085Z",
    "name": "Duration 1",
    "order": 1,
    "project": 1,
    "type": "url"
}
```


45.53. Issue custom attributes values detail


```
{
    "attributes_values": {
        "5": "240 min"
    },
    "issue": 23,
    "version": 2
}
```


45.54. 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\nHarum distinctio rerum nulla quo nisi, explicabo placeat
doloribus earum asperiores repellat nesciunt porro. Recusandae ducimus provident pariatur
similique porro est sint doloremque asperiores, corporis aspernatur provident alias eos a
doloribus tempora, non ullam omnis hic rem temporibus harum amet voluptate, reiciendis
tempora nostrum asperiores autem consequatur inventore saepe ducimus odio. Beatae error
commodi magni repellendus quod consequatur repudiandae necessitatibus magnam ut, neque
laudantium facilis porro quas, molestiae eveniet explicabo magni iure dolore laudantium
reprehenderit tenetur alias illum?\n\nQuo enim distinctio facere eum atque nulla
excepturi eius pariatur voluptate, quasi sapiente in atque, quibusdam repudiandae non
aperiam, quis similique magni, quisquam omnis inventore nihil illum sint cum animi
nesciunt doloribus nulla officia? Dolor illo fugit dicta sint ipsam animi distinctio
```

asperiores eius sequi itaque, asperiores tempora sapiente error excepturi inventore exercitationem nulla blanditiis id? Soluta fugiat hic doloremque dolores amet quisquam veniam nisi quae sunt mollitia, tempora dolor itaque quia deleniti soluta quas, recusandae ullam quod nobis corporis eos magni porro ipsa.\n\nMolestiae ullam ex eius a perspiciatis reiciendis, cumque enim obcaecati quae magni provident, a sit repellendus eum architecto asperiores, dolorem asperiores recusandae adipisci aspernatur iste aliquam. Error cumque quis, quasi vel molestiae, non inventore eveniet a natus ea. Labore at nostrum expedita omnis atque dolores culpa accusantium nemo. Hic aperiam quaerat.\n\nLibero in et dicta molestiae fugiat ipsam cumque totam illo, ex voluptatum accusantium sequi magni placeat nesciunt, quidem sed quo quisquam quis optio ex esse quod consectetur, delectus incidunt quibusdam, doloribus odio fuga reiciendis? Soluta eligendi eius eaque hic accusamus, sequi veniam amet soluta vel nam porro aspernatur iste dolorum eius?\n\nAperiam repudiandae expedita quos numquam excepturi qui illo pariatur quasi modi molestias, voluptas sit minus aliquam enim temporibus veritatis. Debitis ad sapiente ipsum saepe nesciunt officia minus soluta ut labore, vel possimus facilis dolores neque in quos error iure placeat qui ipsam, totam ex optio ad accusamus doloremque aut reiciendis, officiis itaque libero tenetur aliquam velit pariatur. Magnam vero nisi quidem blanditiis incidunt adipisci impedit quasi?\n\nSint veniam sed pariatur aliquam totam voluptatum mollitia minus? Suscipit inventore consectetur consequatur ipsum, id ea esse maxime repudiandae aut nihil vel similique placeat aliquam, eum molestiae facere libero quasi deleniti ea consequatur saepe pariatur, architecto sapiente dolorem aperiam unde nisi repellat odit reiciendis labore optio.\n\nRepellendus accusamus dolor sint quidem sequi odit repellat rerum ullam aspernatur a, deleniti eius sint maiores impedit, quo et nesciunt esse delectus deserunt repellendus, in expedita error ex voluptate itaque. Ipsam autem iure tenetur mollitia in quaerat, earum enim vitae voluptas ipsam officia. Tempore quidem ut odit ad omnis culpa, saepe praesentium sed amet voluptatibus dolorem minus, error provident libero aliquid labore perferendis, enim molestias nostrum, minima numquam unde doloremque nostrum placeat. Temporibus consequuntur quisquam perferendis harum labore a possimus recusandae, ullam veniam quia voluptate, deserunt corrupti unde amet quidem voluptas harum debitis, neque molestiae earum necessitatibus dignissimos dolorem excepturi ipsa dolores rem quis.",
    "created_date": "2016-10-14T09:04:26.857Z",
    "editions": 3,
    "html": " 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 Harum distinctio rerum nulla quo nisi, explicabo placeat doloribus earum asperiores repellat nesciunt porro. Recusandae ducimus provident pariatur similique porro est sint doloremque asperiores, corporis aspernatur provident alias eos a doloribus tempora, non ullam omnis hic rem temporibus harum amet voluptate, reiciendis tempora nostrum asperiores autem consequatur inventore saepe ducimus odio. Beatae error commodi magni repellendus quod consequatur repudiandae necessitatibus magnam ut, neque laudantium facilis porro quas, molestiae eveniet explicabo magni iure dolore laudantium reprehenderit tenetur alias illum? \n Quo enim distinctio facere eum atque nulla excepturi eius pariatur voluptate, quasi sapiente in atque, quibusdam repudiandae non aperiam, quis similique magni, quisquam omnis inventore nihil illum sint cum animi

```
nesciunt doloribus nulla officia? Dolor illo fugit dicta sint ipsam animi distinctio
asperiores eius sequi itaque, asperiores tempora sapiente error excepturi inventore
exercitationem nulla blanditiis id? Soluta fugiat hic doloremque dolores amet quisquam
veniam nisi quae sunt mollitia, tempora dolor itaque quia deleniti soluta quas,
recusandae ullam quod nobis corporis eos magni porro ipsa.</p>\n<p>Molestiae ullam ex
eius a perspiciatis reiciendis, cumque enim obcaecati quae magni provident, a sit
repellendus eum architecto asperiores, dolorem asperiores recusandae adipisci aspernatur
iste aliquam. Error cumque quis, quasi vel molestiae, non inventore eveniet a natus ea.
Labore at nostrum expedita omnis atque dolores culpa accusantium nemo. Hic aperiam
quaerat.</p>\n<p>Libero in et dicta molestiae fugiat ipsam cumque totam illo, ex
voluptatum accusantium sequi magni placeat nesciunt, quidem sed quo quisquam quis optio
ex esse quod consectetur, delectus incidunt quibusdam, doloribus odio fuga reiciendis?
Soluta eligendi eius eaque hic accusamus, sequi veniam amet soluta vel nam porro
aspernatur iste dolorum eius?</p>\n<p>Aperiam repudiandae expedita quos numquam excepturi
qui illo pariatur quasi modi molestias, voluptas sit minus aliquam enim temporibus
veritatis. Debitis ad sapiente ipsum saepe nesciunt officia minus soluta ut labore, vel
possimus facilis dolores neque in quos error iure placeat qui ipsam, totam ex optio ad
accusamus doloremque aut reiciendis, officiis itaque libero tenetur aliquam velit
pariatur. Magnam vero nisi quidem blanditiis incidunt adipisci impedit
quasi?</p>\n<p>Sint veniam sed pariatur aliquam totam voluptatum mollitia minus? Suscipit
inventore consectetur consequatur ipsum, id ea esse maxime repudiandae aut nihil vel
similique placeat aliquam, eum molestiae facere libero quasi deleniti ea consequatur
saepe pariatur, architecto sapiente dolorem aperiam unde nisi repellat odit reiciendis
labore optio.</p>\n<p>Repellendus accusamus dolor sint quidem sequi odit repellat rerum
ullam aspernatur a, deleniti eius sint maiores impedit, quo et nesciunt esse delectus
deserunt repellendus, in expedita error ex voluptate itaque. Ipsam autem iure tenetur
mollitia in quaerat, earum enim vitae voluptas ipsam officia. Tempore quidem ut odit ad
omnis culpa, saepe praesentium sed amet voluptatibus dolorem minus, error provident
libero aliquid labore perferendis, enim molestias nostrum, minima numquam unde doloremque
nostrum placeat. Temporibus consequuntur quisquam perferendis harum labore a possimus
recusandae, ullam veniam quia voluptate, deserunt corrupti unde amet quidem voluptas
harum debitis, neque molestiae earum necessitatibus dignissimos dolorem excepturi ipsa
dolores rem quis.</p>",
    "id": 6,
    "is_watcher": false,
    "last_modifier": 6,
    "modified_date": "2016-10-14T09:20:31.315Z",
    "owner": 8,
    "project": 1,
    "slug": "corporis",
    "total_watchers": 0,
    "version": 2
}
```

45.55. Wiki page watcher detail

```
{
    "full_name": "Silvia Soto",
    "id": 6,
    "username": "user6532909695705815086"
}
```

45.56. Wiki link

```
{
    "href": "labore",
    "id": 1,
    "order": 1476435861955,
    "project": 1,
    "title": "labore"
}
```

45.57. History entry comment

```
{
    "comment": "Distinctio praesentium quod odit ex ut voluptatum, molestiae quos
assumenda, impedit dolor eligendi, repellat ea ipsam officiis ratione reprehenderit
repudiandae earum deleniti voluptatum deserunt, sequi labore molestiae eligendi tempora
hic nostrum adipisci accusamus veniam? Ullam totam asperiores dolorum maiores est amet
odio a rem, dignissimos doloremque eligendi facilis, aspernatur quidem modi ipsum
voluptatem eveniet autem ullam mollitia eius sint dolor. Iste aliquam quaerat dicta quo
dolores dolore praesentium, eos eius nobis molestias at nostrum omnis, a repudiandae
mollitia quam, ab amet corporis libero at consequuntur? Pariatur explicabo iusto enim
aut, optio similique ipsum ducimus nam unde modi pariatur velit nulla dolore?",
    "comment_html": "<p>Distinctio praesentium quod odit ex ut voluptatum, molestiae quos
assumenda, impedit dolor eligendi, repellat ea ipsam officiis ratione reprehenderit
repudiandae earum deleniti voluptatum deserunt, sequi labore molestiae eligendi tempora
hic nostrum adipisci accusamus veniam? Ullam totam asperiores dolorum maiores est amet
odio a rem, dignissimos doloremque eligendi facilis, aspernatur quidem modi ipsum
voluptatem eveniet autem ullam mollitia eius sint dolor. Iste aliquam quaerat dicta quo
dolores dolore praesentium, eos eius nobis molestias at nostrum omnis, a repudiandae
mollitia quam, ab amet corporis libero at consequuntur? Pariatur explicabo iusto enim
aut, optio similique ipsum ducimus nam unde modi pariatur velit nulla dolore?</p>",
    "date": "2016-10-14T09:03:30.362Z",
    "user": {
        "big_photo": null,
        "bio": "",
        "color": "#4B0082",
        "full_name": "Silvia Soto",
        "full_name_display": "Silvia Soto",
        "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
        "id": 6,
        "is_active": true,
        "lang": "",
        "photo": null,
        "projects_with_me": [],
        "roles": [
            "Back",
            "Product Owner",
            "Stakeholder",
            "UX"
        ],
        "theme": "",
        "timezone": "",
        "username": "user6532909695705815086"
    }
}
```

45.58. History entry


```
{
    "comment": "comment edition",
    "comment_html": "<p>comment edition</p>",
    "created_at": "2016-10-14T09:03:30.362Z",
    "delete_comment_date": "2016-10-14T09:20:19.231Z",
    "delete_comment_user": {
        "name": "Silvia Soto",
        "pk": 6
    },
    "diff": {},
    "edit_comment_date": "2016-10-14T09:20:19.003Z",
    "id": "00000000-0000-0000-0000-000000000000",
    "is_hidden": false,
    "is_snapshot": false,
    "key": "userstories.userstory:2",
    "snapshot": null,
    "type": 1,
    "user": {
        "gravatar_id": "9971a763f5dfc5cbd1ce1d2865b4fcfa",
        "is_active": true,
        "name": "Esther Ferrer",
        "photo": null,
        "pk": 9,
        "username": "user3"
    },
    "values": {},
    "values_diff": {}
}
```


45.59. Notify policy


```
{
    "id": 4,
    "notify_level": 2,
    "project": 1,
    "project_name": "Project Example 0"
}
```


45.60. Feedback

```
{
    "comment": "Testing feedback",
    "created_date": "2016-10-14T09:20:32+0000",
    "email": "user6532909695705815086@taigaio.demo",
    "full_name": "Silvia Soto",
    "id": 1
}
```


45.61. Export detail for synch mode


```
{
    "url": "http://localhost:8000/media/exports/1/project-0-
e3c65f5aafd047a194c49c9cbea63107.json"
}
```


45.62. Export accepted response


```
{
    "export_id": "e338555a-3918-4203-8fc6-81b3f7933c79"
}
```


45.63. Import accepted response


```
{
    "import_id": "e338555a-3918-4203-8fc6-81b3f7933c79"
}
```


45.64. Webhook


```
{
    "id": 1,
    "key": "test-key",
    "logs_counter": 1,
    "name": "My service name",
    "project": 1,
    "url": "http://localhost:3000/htbin/test.py"
}
```

45.65. Webhook log


```
{
    "created": "2016-10-14T09:20:23.744Z",
    "duration": 0.0,
    "id": 1,
    "request_data": {
        "action": "test",
        "by": {
            "full_name": "Silvia Soto",
            "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
            "id": 6,
            "permalink": "http://localhost:9001/profile/user6532909695705815086",
            "photo": null,
            "username": "user6532909695705815086"
        },
        "data": {
            "test": "test"
        },
        "date": "2016-10-14T09:20:23.741Z",
        "type": "test"
    },
    "request_headers": {
        "Content-Length": "316",
        "Content-Type": "application/json",
        "X-Hub-Signature": "sha1=c545a527a816c7c95afab27b3c9e6a39142520e3",
        "X-TAIGA-WEBHOOK-SIGNATURE": "c545a527a816c7c95afab27b3c9e6a39142520e3"
    },
    "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
}
```


45.66. Timeline entry detail


```
{
    "content_type": 15,
    "created": "2016-10-14T09:20:03.917Z",
    "data": {
        "comment": "",
        "comment_html": "",
```

```
        "milestone": {
            "id": 1,
            "name": "Sprint 2016-8-20",
            "slug": "sprint-2016-8-20"
        },
        "project": {
            "description": "Project example 0 description",
            "id": 1,
            "name": "Project Example 0",
            "slug": "project-0"
        },
        "user": {
            "big_photo": null,
            "date_joined": "2016-10-14T09:03:27.019Z",
            "gravatar_id": "ece2f7a2dec5f21b2858fecabdcacacc",
            "id": 6,
            "is_profile_visible": true,
            "name": "Silvia Soto",
            "photo": null,
            "username": "user6532909695705815086"
        },
        "userstory": {
            "id": 1,
            "ref": 1,
            "subject": "Patching subject"
        },
        "values_diff": {
            "attachments": {
                "changed": [
                    {
                        "changes": {
                            "description": [
                                "earum vel odit ipsum dolores eos corrupti totam
voluptatibus quod unde",
                                "patching description"
                            ]
                        },
                        "filename": "sample_attachment_2.txt",
                        "thumb_url": null,
                        "url":
"http://localhost:8000/media/attachments/6/5/8/1/94a2f000d46c71d7931f1d29a02e25a235edd3c3
1f71d2305fd5d09698ea/sample_attachment_2.txt"
                    }
                ],
                "deleted": [],
                "new": []
            }
        }
```

```
    },
    "data_content_type": 45,
    "event_type": "userstories.userstory.change",
    "id": 8779,
    "namespace": "project:1",
    "object_id": 1,
    "project": 1
}
```


45.67. 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": "fi",
        "name": "Suomi"
    },
    {
        "bidi": false,
        "code": "fr",
        "name": "Fran\u00e7ais"
    },
    {
        "bidi": false,
        "code": "it",
        "name": "Italiano"
```

```
    },
    {
        "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": "zh-hant",
        "name": "\u4e2d\u6587(\u9999\u6e2f)"
    }
]
```


45.68. Watched


```
{
```

```
    "assigned_to": 8,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Alba Leon",
        "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
        "id": 8,
        "is_active": true,
        "photo": null,
        "username": "user2"
    },
    "created_date": "2016-10-14T09:07:38.005Z",
    "description": null,
    "id": 113,
    "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": true,
    "project_name": "Project Example 6",
    "project_slug": "project-6",
    "ref": 76,
    "slug": null,
    "status": "New",
    "status_color": "#8C2318",
    "subject": "Create the user model",
    "tags_colors": [
        {
            "color": null,
            "name": "dicta"
        },
        {
            "color": "#50a0d5",
            "name": "quos"
        },
        {
            "color": "#da3ba4",
            "name": "ipsum"
        },
        {
            "color": "#db7ec2",
            "name": "dolorum"
        },
        {
            "color": "#9631e4",
            "name": "debitis"
```

```
        },
        {
            "color": null,
            "name": "praesentium"
        },
        {
            "color": null,
            "name": "laborum"
        }
    ],
    "total_voters": 4,
    "total_watchers": 7,
    "type": "issue"
}
```


45.69. Liked


```
[
    {
        "assigned_to": null,
        "assigned_to_extra_info": null,
        "created_date": "2016-10-14T09:20:29.623Z",
        "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": 7,
        "total_watchers": 15,
        "type": "project"
    },
    {
        "assigned_to": null,
```

```
        "assigned_to_extra_info": null,
        "created_date": "2016-10-14T09:07:04.287Z",
        "description": "Project example 5 description",
        "id": 6,
        "is_fan": true,
        "is_private": true,
        "is_watcher": true,
        "logo_small_url": null,
        "name": "Project Example 5",
        "project": null,
        "project_blocked_code": null,
        "project_is_private": null,
        "project_name": null,
        "project_slug": null,
        "ref": null,
        "slug": "project-5",
        "status": null,
        "status_color": null,
        "subject": null,
        "tags_colors": [],
        "total_fans": 11,
        "total_watchers": 12,
        "type": "project"
    },
    {
        "assigned_to": null,
        "assigned_to_extra_info": null,
        "created_date": "2016-10-14T09:07:03.338Z",
        "description": "Project example 4 description",
        "id": 5,
        "is_fan": true,
        "is_private": true,
        "is_watcher": true,
        "logo_small_url": null,
        "name": "Project Example 4",
        "project": null,
        "project_blocked_code": null,
        "project_is_private": null,
        "project_name": null,
        "project_slug": null,
        "ref": null,
        "slug": "project-4",
        "status": null,
        "status_color": null,
        "subject": null,
        "tags_colors": [],
        "total_fans": 9,
        "total_watchers": 12,
```

```
        "type": "project"
    }
]
```


45.70. Voted


```
{
    "assigned_to": 8,
    "assigned_to_extra_info": {
        "big_photo": null,
        "full_name_display": "Alba Leon",
        "gravatar_id": "5c921c7bd676b7b4992501005d243c42",
        "id": 8,
        "is_active": true,
        "photo": null,
        "username": "user2"
    },
    "created_date": "2016-10-14T09:07:35.474Z",
    "description": null,
    "id": 104,
    "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": true,
    "project_name": "Project Example 6",
    "project_slug": "project-6",
    "ref": 67,
    "slug": null,
    "status": "In progress",
    "status_color": "#5E8C6A",
    "subject": "Create the user model",
    "tags_colors": [
        {
            "color": null,
            "name": "ut"
        },
        {
            "color": "#7fea8e",
            "name": "dolores"
        },
        {
            "color": null,
```

```
            "name": "nesciunt"
        }
    ],
    "total_voters": 5,
    "total_watchers": 1,
    "type": "issue"
}
```


45.71. Discover stats


```
{
    "projects": {
        "total": 2
    }
}
```


45.72. 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": 7,
        "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": {
            "2016-10-10": 13
        },
        "today": 12,
        "total": 13
    },
    "userstories": {
        "average_last_five_working_days": 0.0,
        "average_last_seven_days": 0.0,
        "today": 151,
        "total": 180
    }
}
```


46. 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)