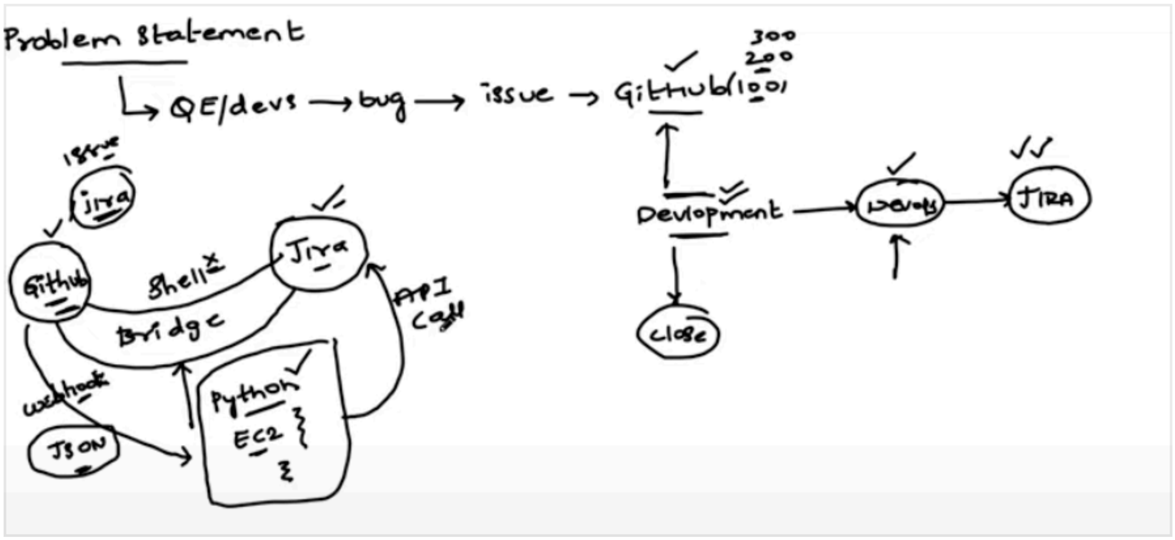
JIRA PROJECT with Python Flask



```
GET

import requests
from requests.auth import HTTPBasicAuth
import json

url = "https://kundan-antyakula.atlassian.net/rest/api/3/
project"


API_TOKEN = "ATATT3xFfGF0pWinVpqi5I7_W6k9G4k9aMRL1IBimLd0oK-
yt1zGeXgWlciI679oNRfnfAZbQvY6mFJfae3fsDyru7pJLnq8z7PpZwPwGBc
2g7kkqMWfbGO1IFfl0F_wCKXsR9IM5ghRXDx0nIHjcw4Wyyk2nxneySs_0kJ
Xk97lYZMu_QjmWrQ=2F3784E9"

auth = HTTPBasicAuth("kundanantyakula@gmail.com", API_TOKEN)

headers = {
  "Accept": "application/json"
}




response = requests.request(
   "GET",
   url,
   headers=headers,
   auth=auth
)
```

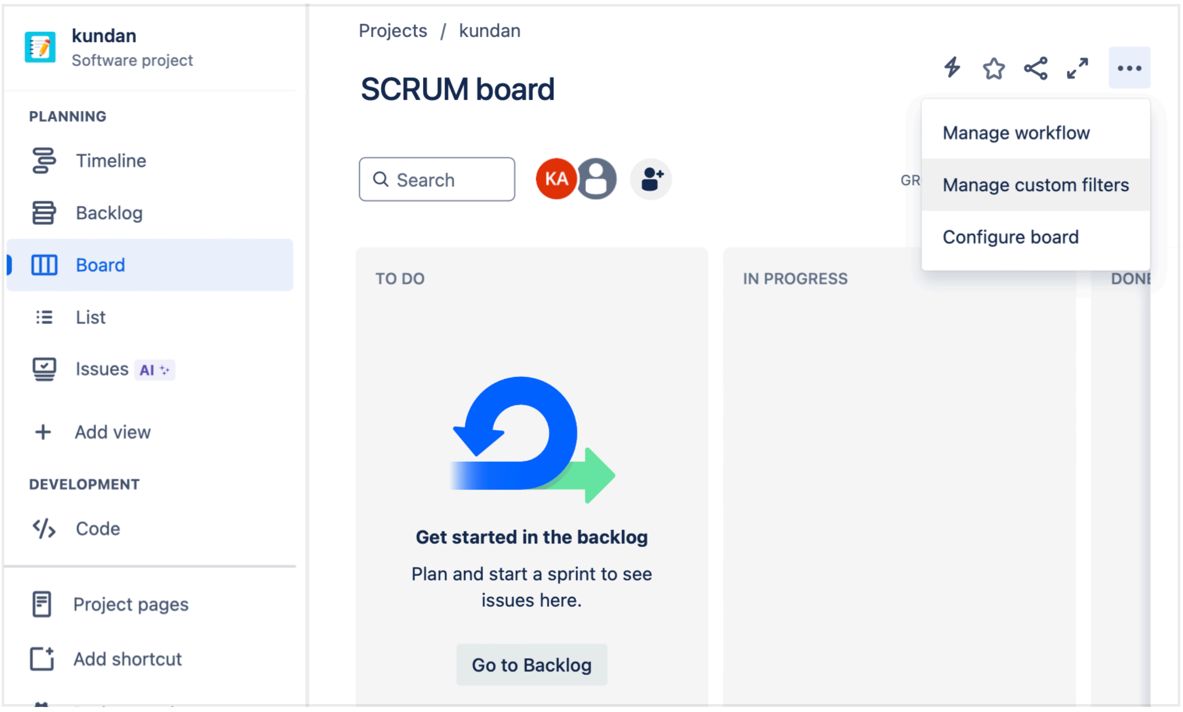
```
print(json.dumps(json.loads(response.text), sort_keys=True,
indent=4, separators=(",", ": ")))

ouput = json.loads(response.text)
name = ouput[0]["name"]
print(name)
for i in ouput:
    print(i['name'])
```

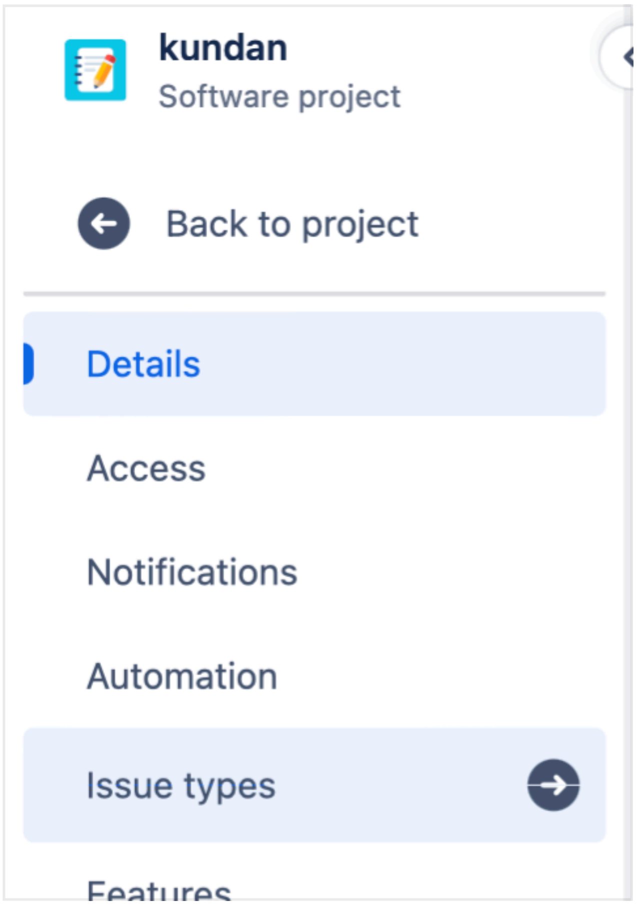
o/p -

```
python3 list_project.py
[
  {
    "avatarUrls": {
      "16x16": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10401?size=xsmall",
      "24x24": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10401?size=small",
      "32x32": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10401?size=medium",
      "48x48": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10401"
    },
    "entityId": "6c11e543-8d19-405d-8092-9573ebe71c4b",
    "expand":
"description,lead,issueTypes,url,projectKeys,permissions,insight",
    "id": "10001",
    "isPrivate": false,
    "key": "KUN",
    "name": "kundan",
    "projectTypeKey": "software",
    "properties": {},
    "self": "https://kundan-antyakula.atlassian.net/rest/api/3/project/10001",
    "simplified": true,
    "style": "next-gen",
    "uuid": "6c11e543-8d19-405d-8092-9573ebe71c4b"
  },
  {
    "avatarUrls": {
      "16x16": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10405?size=xsmall",
      "24x24": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10405?size=small",
      "32x32": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10405?size=medium",
          "48x48": "https://kundan-antyakula.atlassian.net/rest/api/3/
universal_avatar/view/type/project/avatar/10405"
       },
       "expand":
"description,lead,issueTypes,url,projectKeys,permissions,insight",
       "id": "10000",
       "isPrivate": false,
       "key": "SUP",
       "name": "Support",
       "projectTypeKey": "service_desk",
       "properties": {},
       "self": "https://kundan-antyakula.atlassian.net/rest/api/3/project/10000",
       "simplified": false,
       "style": "classic"
    }
]
```

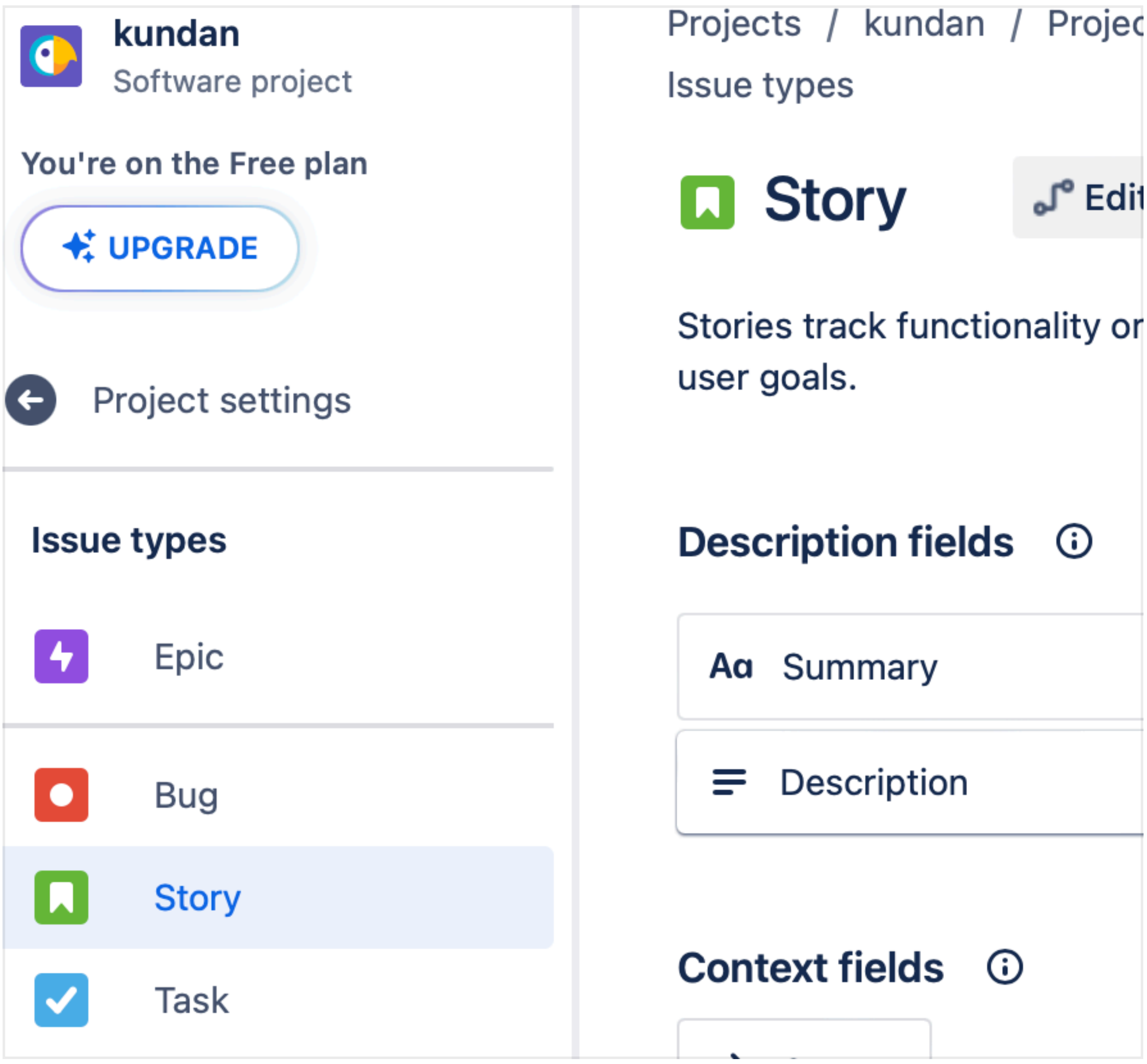
kundan
kundan
Support

Create Automated issues





select story then you will be getting project id in the web link on top of the web page


kundan

Software project

You're on the Free plan

UPGRADE

Project settings

Issue types

Epic

Bug

Story

Task

Projects / kundan / Projec

Issue types

Story

Edit

Stories track functionality or user goals.

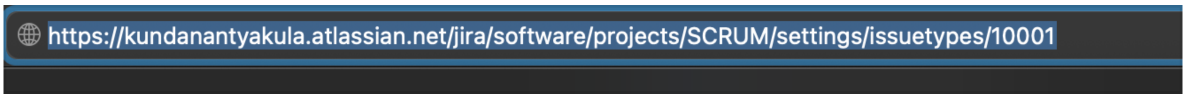
Description fields

Summary

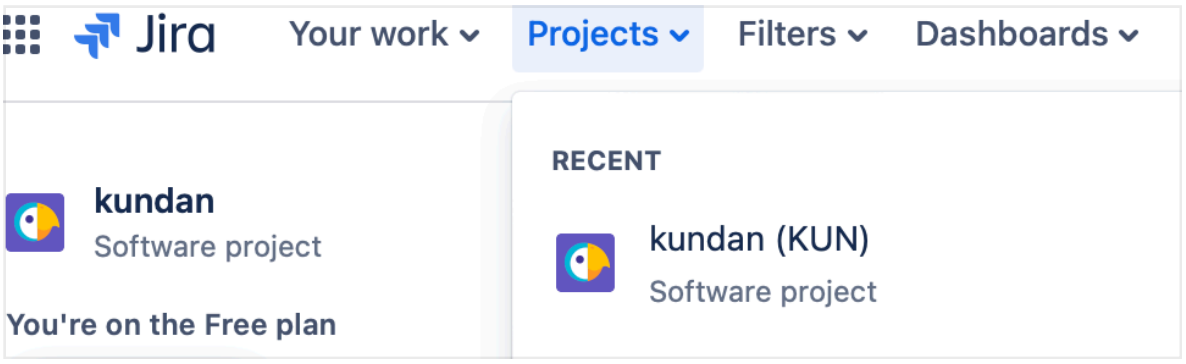
Description

Context fields

similar to this the last one is the issue type id in the code


https://kundanantyakula.atlassian.net/jira/software/projects/SCRUM/settings/issuetypes/10001

Now hover on projects



kundan

Software project

You're on the Free plan

RECENT

kundan (KUN)

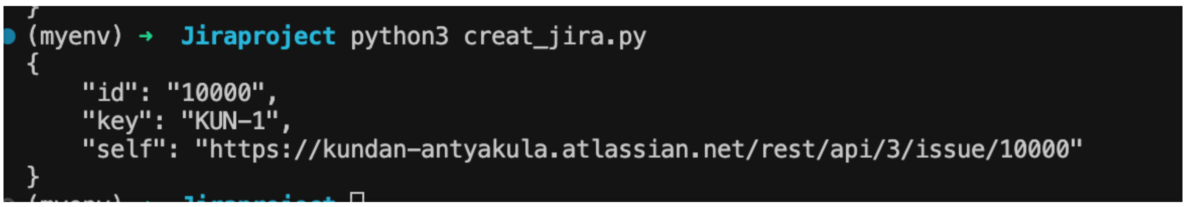
Software project

Project key will be KUN

```
# This code sample uses the 'requests' library:
# http://docs.python-requests.org
import requests
from requests.auth import HTTPBasicAuth
```

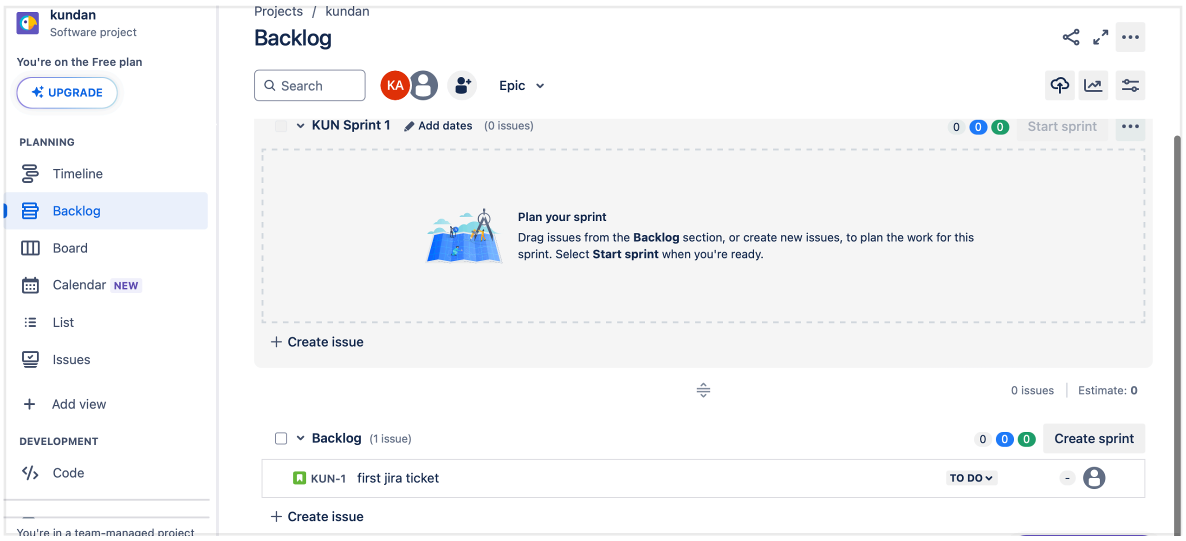
```
import json

url = "https://kundan-antyakula.atlassian.net/rest/api/3/
issue"

APITOKEN = "ATATT3xFfGF0pWinVpqi5I7_W6k9G4k9aMRL1IBimLd0oK-
yt1zGeXgWlciI679oNRfnfAZbQvY6mFJfae3fsDyru7pJLnq8z7PpZwPwGBc
2g7kkqMWfbGO1IFfl0F_wCKXsR9IM5ghRXDx0nIHjcw4Wyyk2nxneySs_0kJ
Xk97lYZMu_QjmWrQ=2F3784E9"
auth = HTTPBasicAuth("kundanantyakula@gmail.com", APITOKEN)

headers = {
  "Accept": "application/json",
  "Content-Type": "application/json"
}

payload = json.dumps( {
  "fields": {

    "description": {
      "content": [
        {
          "content": [
            {
              "text": "Order entry fails when selecting
supplier.",
              "type": "text"
            }
          ],
          "type": "paragraph"
        }
      ],
      "type": "doc",
      "version": 1
    },

    "issuetype": {
      "id": "10006"
    },

    "project": {
      "key": "KUN"
    },

    "summary": "first jira ticket",

  },
    "update": {}
} )

response = requests.request(
    "POST",
    url,
    data=payload,
    headers=headers,
    auth=auth
)

print(json.dumps(json.loads(response.text), sort_keys=True,
indent=4, separators=(",", ": ")))
```

o/p-



```
(myenv) → Jiraproject python3 creat_jira.py
{
    "id": "10000",
    "key": "KUN-1",
    "self": "https://kundan-antyakula.atlassian.net/rest/api/3/issue/10000"
}
```

finally output indicating that it created a jira ticket

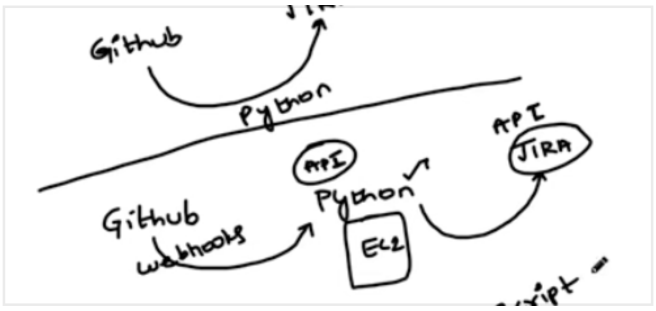


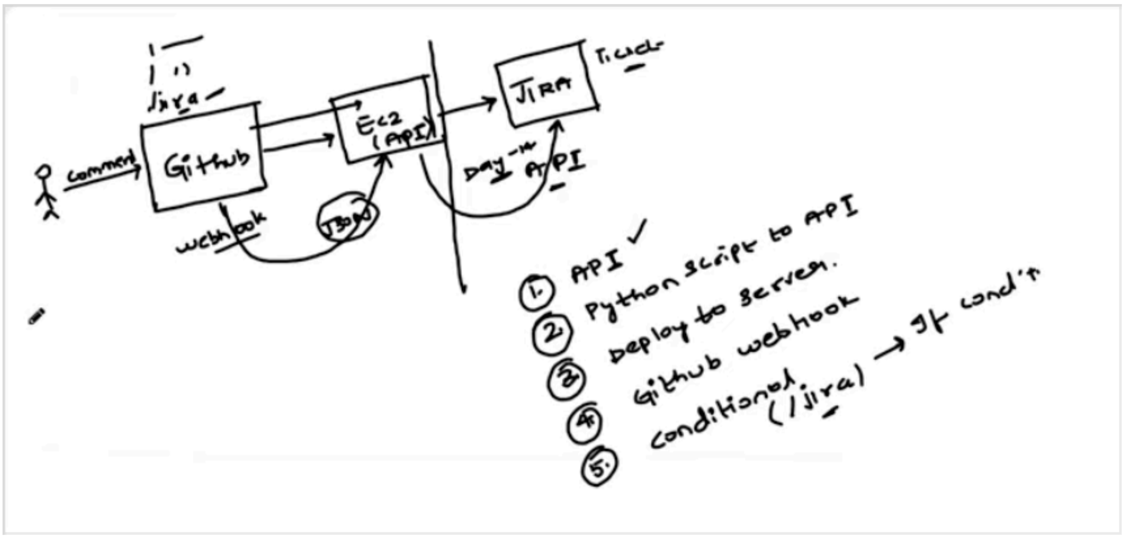
Plan your sprint

Drag issues from the **Backlog** section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

KUN-1 first jira ticket

Creating Flask API integrating with github web hook creates issue on /jira

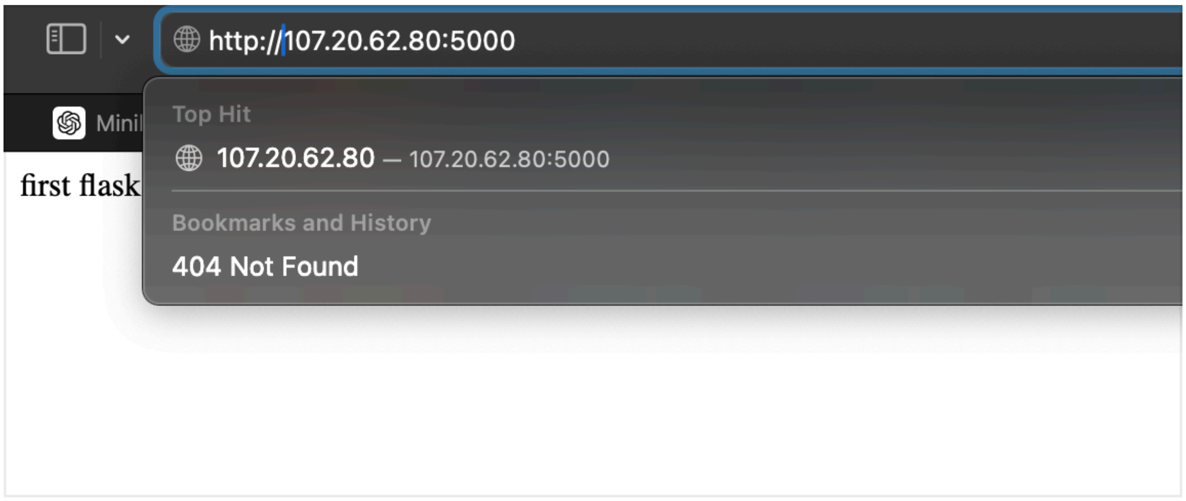




Create Flask app returning hello world

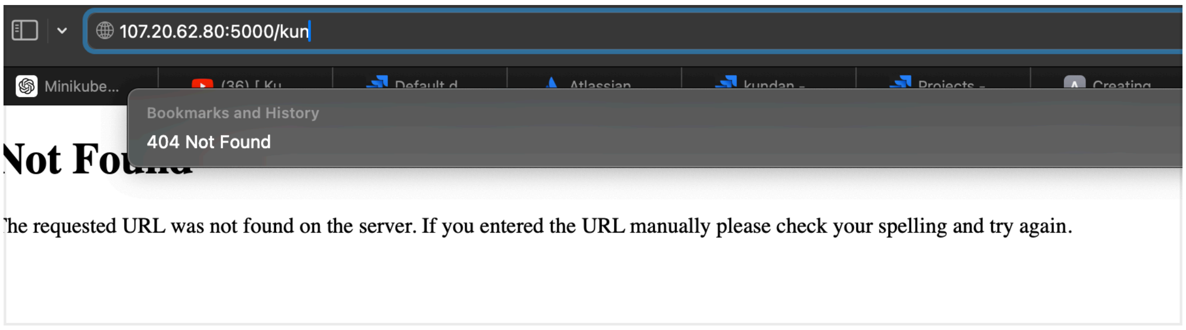
Ec2 ubuntu instance creation

```
from flask import Flask
app = Flask(__name__) #creating flask app instance

@app.route("/") # decorator purpose is before invoking
function it perform action
#if someone is working on API but before he needs to be
authenticated so decorator can be used in such circumstances
#similarly here if someone wants to access this hello API
are they trying to access on particular path or not
def fl():
    return "first flask"

app.run('0.0.0.0')
#to build we need server but flask has inbuilt development
server without deploying on tomcat or bla bla
```



first flask

```
@app.route("/")
this decorator is not letting us enter anything after / can
only accesed before /
```



Not Found

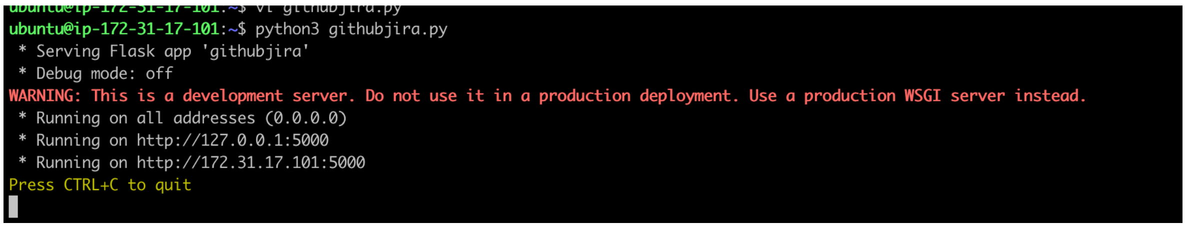
The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

what is we keep someother after / ?

Intergate above falsk code with GET code from jira first program & convert into POST method as we need post operation performed to push issues through /jira in github issues & automatically generate issue in jira .
Run this code on the ec2 instance

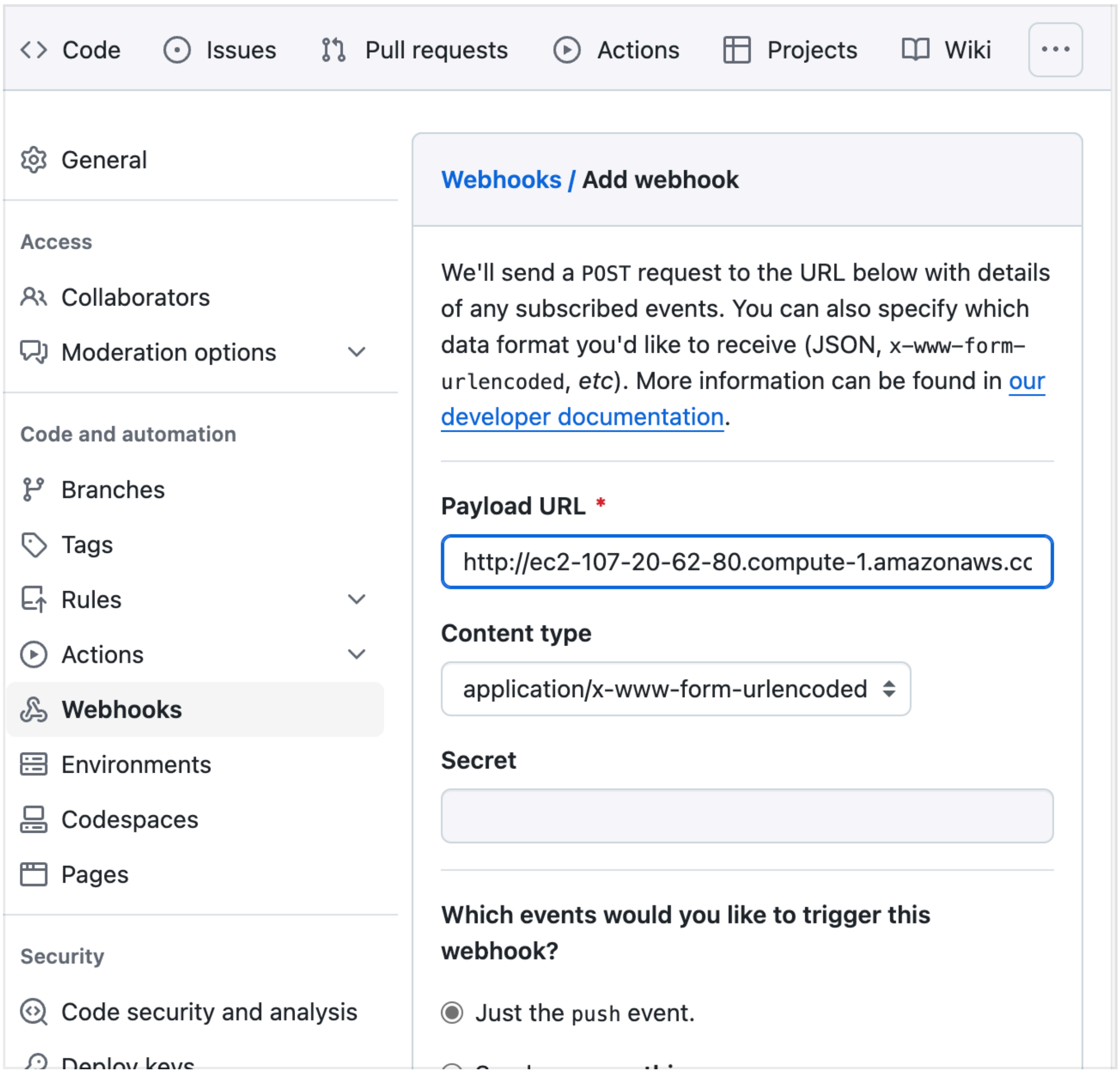
```
# This code sample uses the 'requests' library:
# http://docs.python-requests.org
import requests
from requests.auth import HTTPBasicAuth
import json
from flask import Flask

app = Flask(__name__)

# Define a route that handles GET requests
@app.route('/createJira', methods=['POST'])
def createJira():

    url = "https://kundan-antyakula.atlassian.net/rest/api/
3/issue"
```

```
    auth = HTTPBasicAuth("kundanantyakula@gmail.com",
API_TOKEN)

    headers = {
        "Accept": "application/json",
        "Content-Type": "application/json"
    }

    payload = json.dumps( {
        "fields": {
        "description": {
            "content": [
                {
                    "content": [
                        {
                            "text": "Order entry fails when
selecting supplier.",
                            "type": "text"
                        }
                    ],
                    "type": "paragraph"
                    }
                ],
            "type": "doc",
             "version": 1
        },
        "project": {
           "key": "KUN"
        },
        "issuetype": {
            "id": "10006"
        },
        "summary": "Main order flow broken",
    },
    "update": {}
    } )


    response = requests.request(
        "POST",
        url,
        data=payload,
        headers=headers,
        auth=auth
    )
    return json.dumps(json.loads(response.text),
sort_keys=True, indent=4, separators=(",", ": "))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```



```
ubuntu@ip-172-31-17-101:~$ vi githubjira.py
ubuntu@ip-172-31-17-101:~$ python3 githubjira.py
 * Serving Flask app 'githubjira'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.17.101:5000
Press CTRL+C to quit
```

Github Repo settings -> Now, Add web hook from public dns of EC2 instance



Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *

http://ec2-107-20-62-80.compute-1.amazonaws.cc

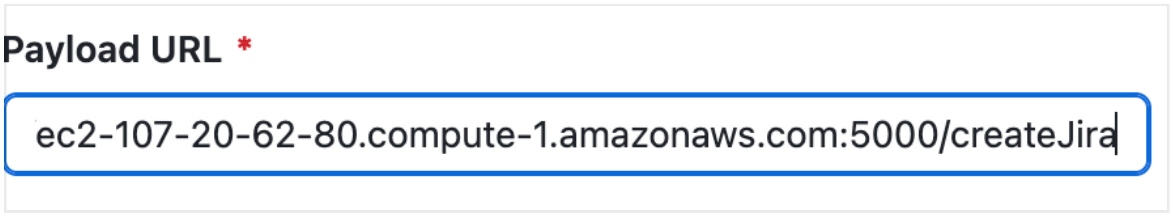
Content type

application/x-www-form-urlencoded

Secret

Which events would you like to trigger this webhook?

● Just the push event.

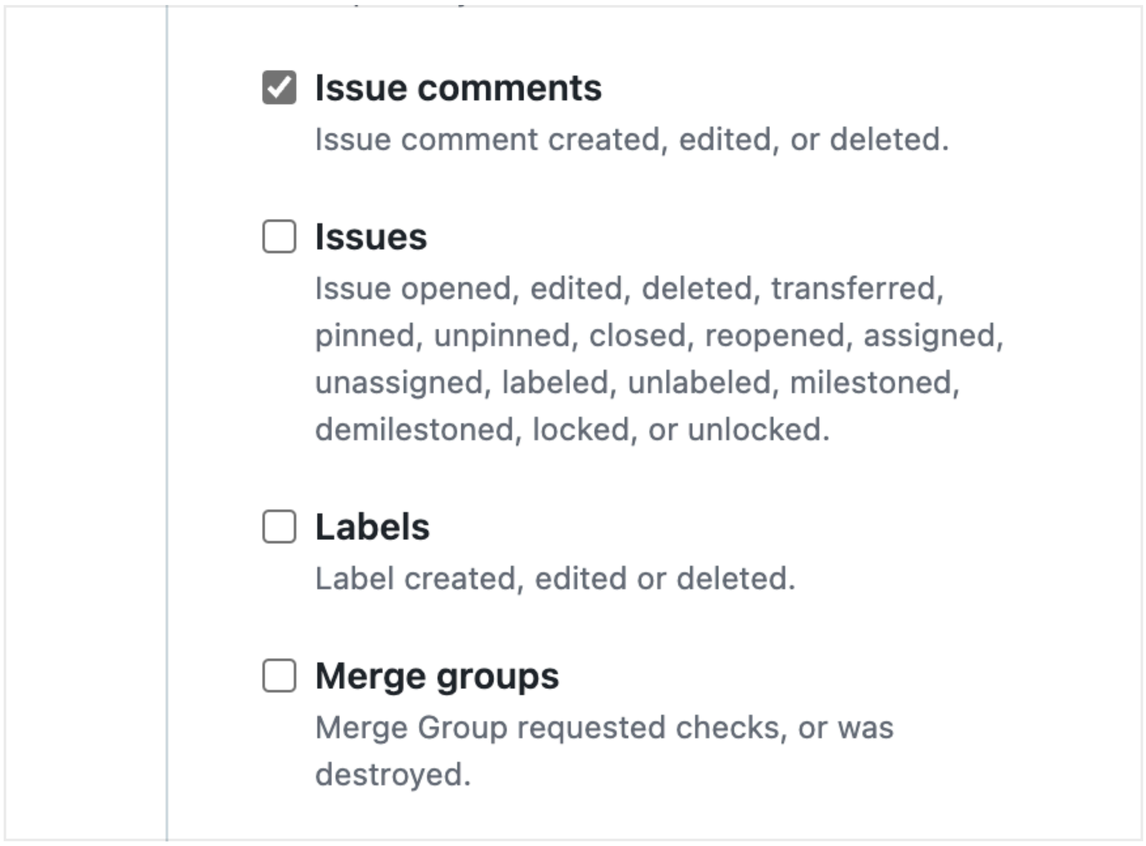


Payload URL *

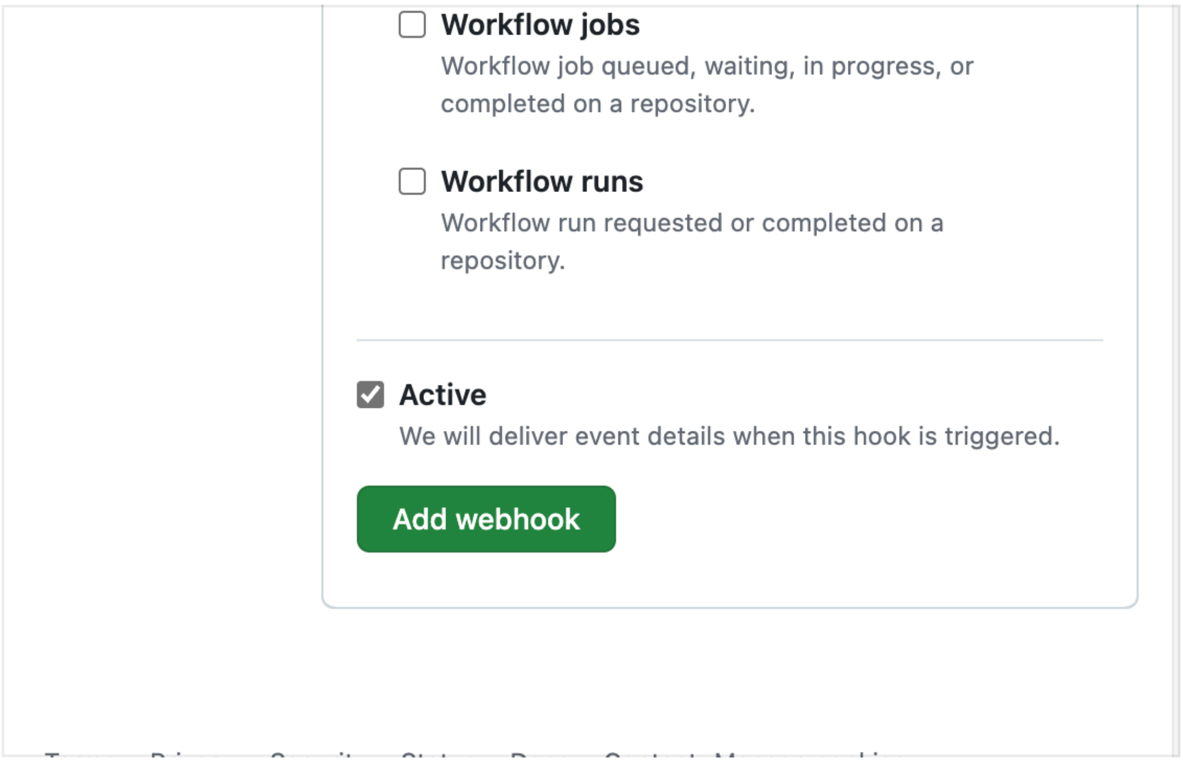
ec2-107-20-62-80.compute-1.amazonaws.com:5000/createJira

Be extra cautious while entering payload URL make sure port no and createJira added in code is similar.

Select radio button -> Let me select individual events -> Issue Comments

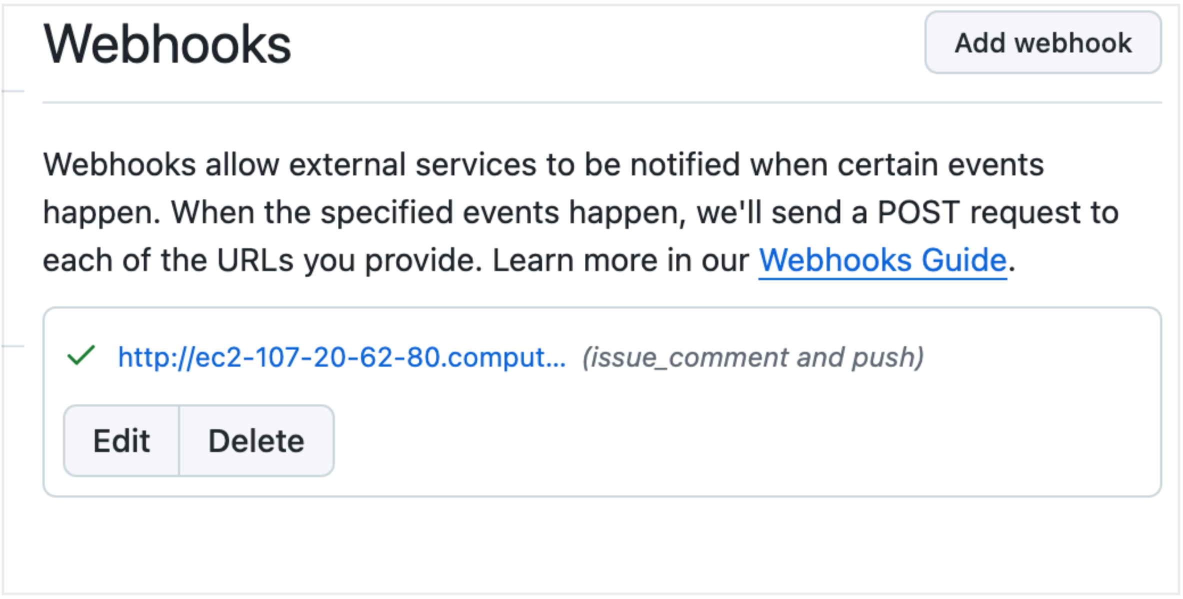


- [✓] **Issue comments**
  Issue comment created, edited, or deleted.
- [ ] **Issues**
  Issue opened, edited, deleted, transferred, pinned, unpinned, closed, reopened, assigned, unassigned, labeled, unlabeled, milestoned, demilestoned, locked, or unlocked.
- [ ] **Labels**
  Label created, edited or deleted.
- [ ] **Merge groups**
  Merge Group requested checks, or was destroyed.


☐ **Workflow jobs**
Workflow job queued, waiting, in progress, or completed on a repository.

☐ **Workflow runs**
Workflow run requested or completed on a repository.

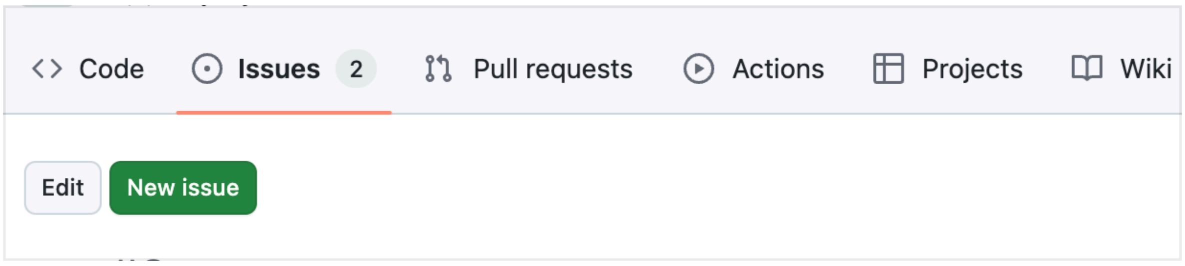
☑ **Active**
We will deliver event details when this hook is triggered.

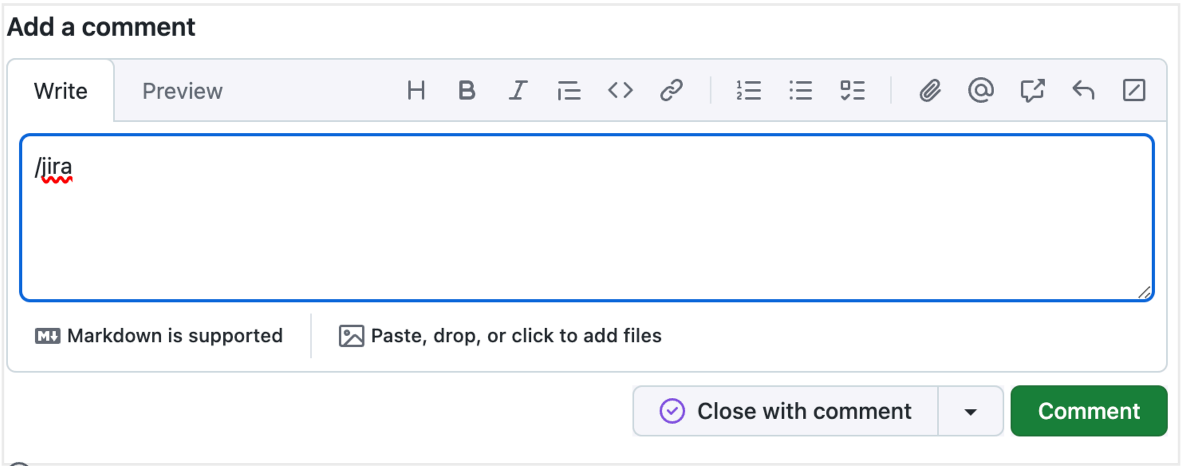

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://ec2-107-20-62-80.comput... *(issue_comment and push)*

Click on New issue -> add comment





Add a comment

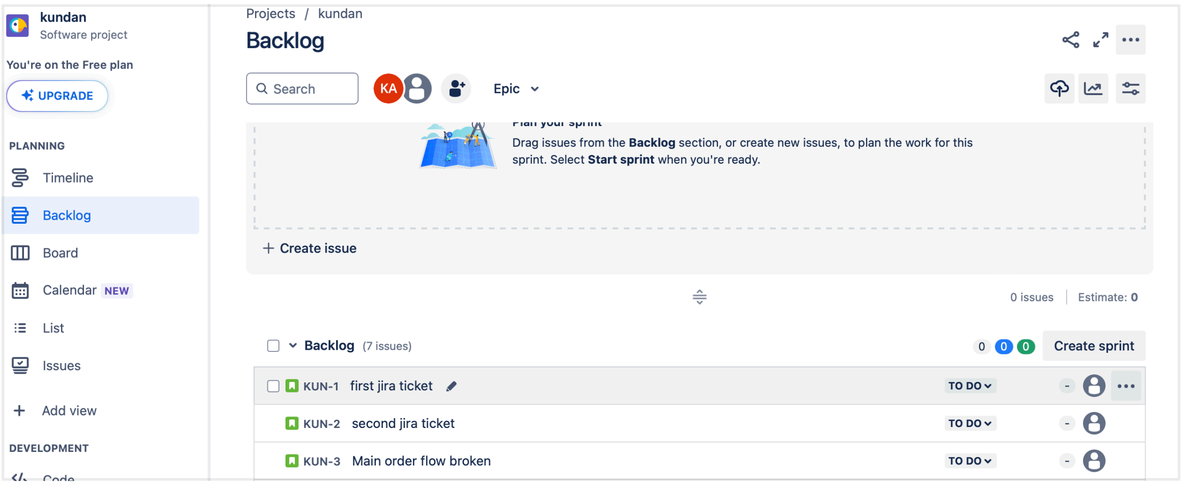
Write Preview

/jira

Markdown is supported   Paste, drop, or click to add files

Close with comment   Comment

"summary": "Main order flow broken" issue is generated in jira automatically



kundan
Software project
You're on the Free plan
UPGRADE

PLANNING
Timeline
Backlog
Board
Calendar NEW
List
Issues
Add view

DEVELOPMENT
Code

Projects / kundan

Backlog

Search   Epic

Plan your sprint
Drag issues from the **Backlog** section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

+ Create issue

0 issues | Estimate: 0

Backlog (7 issues)   0 0 0   Create sprint

KUN-1	first jira ticket	TO DO	-
KUN-2	second jira ticket	TO DO	-
KUN-3	Main order flow broken	TO DO	-