alexpchin / **Setting_upa_new_repo.md**

Last active 2 weeks ago • Report abuse

⭐ Star

`<>` **Code**   ⦿ Revisions 2   ☆ Stars 191   ⑂ Forks 81

Create a new repository on the command line

`<>` **Setting_upa_new_repo.md**

Setting up a new Git Repo

##Create a new repository on the command line

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:alexpchin/<reponame>.git
git push -u origin master
```

##Push an existing repository from the command line

```
git remote add origin git@github.com:alexpchin/<reponame>.git
git push -u origin master
```

Load earlier comments...

Amruth-Smaragdine commented on Jun 21, 2020

@ziizium - I shall wait for your inputs.

---

ziizium commented on Jun 22, 2020 • edited ▾

@Amruth-Smaragdine I tried to create a repo using Microsoft Windows command line I got the same JSON error like you mentioned (see the attached image). I tried with *double quotes* because you said it solved it for you but to no avail. Currently, have not been able to solve it.

```
C:\Users\...\testRepo>curl -u ziiziu ...                    https://api.github.com/user/repos -d '{"name":"testRepo"}'
{
  "message": "Problems parsing JSON",
  "documentation_url": "https://developer.github.com/v3/repos/#create"
}

C:\Users\...\testRepo>
```

How did you solve it?

Pending your reply and solution to the problem, I'll update my initial comment to indicate that the procedure will work with Git Bash on Microsoft Windows.

Amruth-Smaragdine commented on Jun 22, 2020

@ziizium:
Please allow me some time to check and revert ASAP. Maybe I missed a special character.

Amruth-Smaragdine commented on Jun 22, 2020

I am not sure how I missed entering `"\"` in my earlier command.
Here is the correct command: `curl -u Amruth-Smaragdine https://api.github.com/user/repos -d "{\"name\": \"testrepo\"}"`
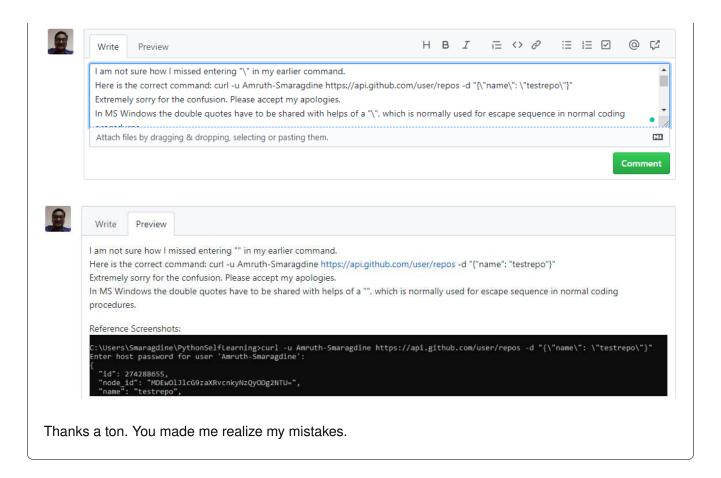Extremely sorry for the confusion. Please accept my apologies.
In MS Windows the double quotes have to be shared with the help of a `"\"` , which is normally used for escape sequence in normal coding procedures.

Reference Screenshots:

```
C:\Users\Smaragdine\PythonSelfLearning>curl -u Amruth-Smaragdine https://api.github.com/user/repos -d "{\"name\": \"testrepo\"}"
Enter host password for user 'Amruth-Smaragdine':
{
  "id": 274288655,
  "node_id": "MDEwOlJlcG9zaXRvcnkyNzQyODg2NTU=",
  "name": "testrepo",
```

Overview   **Repositories 2**   Projects 0   Packages 0   Stars 0   Followers 0   Following 0

Find a repository...   Type: All ▾   Language: All ▾   🖵 New

testrepo   ☆ Star
Updated 11 minutes ago

Note: I have identified something that I ignored earlier. In my earlier comment, I have not included the text with <> insert code option, hence the `"\"` was not getting previewed.

Write | Preview

H B I ≣ <> 🔗 ≔ ≔ ☑ @ ⌇

I am not sure how I missed entering "\" in my earlier command.
Here is the correct command: curl -u Amruth-Smaragdine https://api.github.com/user/repos -d "{\"name\": \"testrepo\"}"
Extremely sorry for the confusion. Please accept my apologies.
In MS Windows the double quotes have to be shared with helps of a "\", which is normally used for escape sequence in normal coding

Attach files by dragging & dropping, selecting or pasting them.

Comment

Write | Preview

I am not sure how I missed entering "" in my earlier command.
Here is the correct command: curl -u Amruth-Smaragdine https://api.github.com/user/repos -d "{"name": "testrepo"}"
Extremely sorry for the confusion. Please accept my apologies.
In MS Windows the double quotes have to be shared with helps of a "", which is normally used for escape sequence in normal coding procedures.

Reference Screenshots:

```
C:\Users\Smaragdine\PythonSelfLearning>curl -u Amruth-Smaragdine https://api.github.com/user/repos -d "{\"name\": \"testrepo\"}"
Enter host password for user 'Amruth-Smaragdine':
{
  "id": 274288655,
  "node_id": "MDEwOlJlcG9zaXRvcnkyNzQyODg2NTU=",
  "name": "testrepo",
```

Thanks a ton. You made me realize my mistakes.

---

ziizium commented on Jun 27, 2020

@Amruth-Smaragdine Success! (see the attached image).

```
C:\Users   \testRepo>curl -u ziizium:                        https://api.github.com/user/repos -d "{\"name\": \"testRepo\"}"
{
  "id": 27!       ,
  "node_id": "MDEw(                       ",
  "name": "testRepo",
  "full_name": "ziizium/testRepo",
  "private": false,
  "owner": {
    "login": "ziizium",
    "id": 4        ,
    "node_id": "MDQ6VXN               ",
    "avatar_url": "https://avatars1.githubusercontent.com/u/          ",
```

I'll update my initial comment giving you the full credit. Thank you very much!.

Amruth-Smaragdine commented on Jun 27, 2020

Thanks a ton, **@ziizium** - I am happy that I could contribute something that helps everyone.

---

slavanorm commented on Jul 4, 2020

> There is this project from github.
> Works like a charm.
> **https://github.com/github/hub**

memeing stackoverflow:
this should be an accepted answer

---

jamiecropley commented on Aug 23, 2020 • edited ▾

> There is this project from github.
> Works like a charm.
> **https://github.com/github/hub**

> memeing stackoverflow:
> this should be an accepted answer

This is actually pretty good. Actually this solution is amazing, cheers for sharing!

HermanNygaard commented on Sep 17, 2020

The offical GitHub CLI is now out of beta: https://github.com/cli/cli
Create a repo with `gh repo create [<name>]`
Docs: https://cli.github.com/manual/gh_repo_create

---

ziizium commented on Sep 27, 2020

@HermanNygaard Thank you for your contribution, I'll definitely have a look.

---

moinabyssinia commented on Oct 4, 2020

> so I thought I'd leave this here to help everyone else who was having similar issues:
>
> ```
> mkdir myDirName     #this is the name of your directory
> cd /myDirName
> git init
> touch readME.md    #this is to create an initial file to push
> git commit -m "enter commit message here"
> git remote add origin git@github.com:YOUR_USERNAME/myDirName.git
>
> curl -u USERNAME:PASSWORD https://api.github.com/user/repos -d
> '{"name":"myDirName"}' #this will create the repo in github.
>
> # if you haven't generated and SSH key for github access then follow these
> steps, otherwise you're good to push your shit to github.
> eval $(ssh-agent -s)
> ssh-keygen -t rsa -b 4096 -C "email@yourdomain.com" #this should be your github
> ```

```
email address
## you'll be prompted to a couple of times. Press enter for the first prompt.
choose a passphrase for the second prompt, or press enter twice for no
passphrase
ssh-add ~/.ssh/id_rsa    #this is your private key
cat ~/.ssh/id_rsa.pub   # copy the output of this command. this is your SSH
public key
curl -u USERNAME:PASSWORD https://api.github.com/user/keys -d
'{"title":"KEY_NAME", "key":"YOUR_RSA_PUBLIC_KEY_HERE"}'   #the value you
copied earlier and your keyname. I recommend using a combination of machine
name and app (My-laptop (Git CLI)
git push -u origin master
```

You can certainly change the order of steps and do the curl commands ssh generation first to create the repo and push the keys, but I used this flow just because.

thanks! it helped.

GoalTV commented on Oct 26, 2020 • edited ▾

Anyone who wants to have their own project.sh simple do varible $name I have modified the code above based on code above.

```
echo Please Provide New Repo Name:
read  name

mkdir $name     #this is the name of your directory
cd /$name
git init
touch $name/readME.md    #this is to create an initial file to push
git commit -m "enter commit message here"
git remote add origin git@github.com:YOUR_USERNAME/$name.git

curl -u USERNAME:PASSWORD https://api.github.com/user/repos -d '{"name":"'$name'"}'
#this will create the repo in github.

# if you haven't generated and SSH key for github access then follow these steps,
otherwise you're good to push your shit to github.
eval $(ssh-agent -s)
ssh-keygen -t rsa -b 4096 -C "email@yourdomain.com" #this should be your github
email address
## you'll be prompted to a couple of times. Press enter for the first prompt. choose
a passphrase for the second prompt, or press enter twice for no passphrase
ssh-add ~/.ssh/id_rsa   #this is your private key
cat ~/.ssh/id_rsa.pub   # copy the output of this command. this is your SSH public
key
curl -u USERNAME:PASSWORD https://api.github.com/user/keys -d '{"title":"KEY_NAME",
"key":"YOUR_RSA_PUBLIC_KEY_HERE"}'   #the value you copied earlier and your keyname.
I recommend using a combination of machine name and app (My-laptop (Git CLI)
git push -u origin master
```

Magentu commented on Nov 28, 2020

worked

---

cappelletto commented on Dec 7, 2020

> I have created this script. It allows for a repo to be created from terminal without having to create it on github/bitbucket first.
>
> It uses Personal Access Token for authentication on github and regular user:password auth on bitbucket
>
> git-new CLI

Worked like a charm, thank you!

Kumar160190 commented on Mar 24, 2021

how to create new repo in CLI?

---

maxirios commented on Apr 20, 2021

> how to create new repo in CLI?

just write `gh repo create [name] [--flags]` (brackets means optional values)

Ramesh-kumar-S commented on May 22, 2021

I think instead of SSH , This Might also Works Fine!!

curl https://api.github.com/user/repos?access_token=myAccessToken -d '{"name":"REPO ANME"}'

If you Don't have an Access Token You could Just go to Git Hub Settings > Developer Settings > Personal Access Tokens and Get your Token Replace that in the Curl Command.

git remote set-url origin https://github.com/user name/Repo name.git

git remote add origin https://github.com/user name/repo name.git

git push -u origin main

And yeah!! you are all Set!!

Go ahead and Have Fun!!

nettcaster87 commented on Jun 15, 2021

This command worked for me.

curl -u USERNAME:ACCESS_TOKEN https://api.github.com/user/repos -d '{"name":"myDirName"}'
#this will create the repo in github.

I doesn't work if I just enter my username and wait for the password prompt, or enter my password after my username. I'm not sure why, but the access token worked first time, no stress.

Does anyone know why it fails with username & password? just out of interest, I suspect it's a security issue.

ygautomo commented on Jul 3, 2021 • edited ▾

@nettcaster87. you should using PAT (Personal Access Token) for authentication.
$ curl -u username:token https://api.github.com/user

The current methods are depreciated
Here's the reference.
https://docs.github.com/en/rest/overview/other-authentication-methods#via-oauth-and-personal-access-tokens.
https://docs.github.com/en/rest/guides/basics-of-authentication#accepting-user-authorization.

JotGoyal commented on Aug 2, 2021

```
ERROR: Repository not found.
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

SOLUTION --------
use command

1. git remote -v
2. git remote remove origin (remote name)

Hiroki180 commented on Nov 13, 2021

git config
git init
git clone
git add
git commit
git diff
git reset
git status
git rm
git log
git show
git tag
git branch
git checkout
git merge
git remote
git push
git pull
git stash

108806 commented on Nov 13, 2021

Never works for me. I think git was created as a joke.

---

ghost commented on Dec 16, 2021

If you still need to do the whole create repo on github website.
Just create repo, git clone and then push. So commands are just useless.

---

alcorkrishna commented on Apr 29, 2022

The above example talks creating repo at user level, how to create remote repo at organizational level.
Thanks

---

auwsom commented on May 8, 2022 • edited ▾

@Amruth-Smaragdine
" curl -u USERNAME:PERSONAL_ACCESS_TOKEN https://api.github.com/user/repos -d
'{"name":"myDirName"}' #this will create the repo in github.
**This last command will save your Personal Access Token or password in your Bash history. If I
find a way to prevent this I will update this comment**."

The way to keep the PAC key from bash history is to use one space in front of any command.

If you want to delete the last history line manually use:
```
history -d $(($HISTCMD - 2)); history -w;
```
Or find the line number and use:
```
history -d <line number>; history -w;
```

Amondale commented on Aug 1, 2022 • edited ▾

since August 2021, using bare HTTPS with a username/password no longer works to create a repo on git. Instead, use a PAT (Personal Access token) created in the web API.

The samples above return:
```
{ "message": "Requires authentication", "documentation_url": "https://docs.github.com
/rest/reference/repos#create-a-repository-for-the-authenticated-user" }
```

Here's the way that will create a repository using cmdline:
```
curl -H "Authorization: token {your_personal_access_token}" https://api.github.com
/user/repos -d '{"name":"{new_repo_name}"}'
```

---

devxnsh commented on Oct 19, 2022

@Amondale Thanks for the reply, however I am still getting a Bad Credentials error. I also noticed that this command does not issue username...is there a previous command which you used? Thanks.

---

auwsom commented on Oct 19, 2022

I still have some issues when using this command on Linux so I just switched to using the GitHub CLI package (`apt install gh`) and `gh repo create <newrepo> --public` . `gh auth login` has a quick little login utility.

Amondale commented on Oct 19, 2022

hey **@devxnsh** if you set up your PAT in the web browser correctly, it won't be requiring the username/password, see [this article](#) for more info. Now that `gh` is out of beta, it should be the best approach rather than doing `curl` ing, as the Canadians say, eh?/

---

jimmy8585 commented 2 weeks ago

if i create new repository in git i want see automatically this step , how to create new repository in command line