How to deal with "refusing to merge unrelated histories" error?

git pull --allow-unrelated-histories

<https://github.community/t5/How-to-use-Git-and-GitHub/How-to-deal-with-quot-refusing-to-merge-unrelated-histories-quot/td-p/12619>

**error: cannot lock ref 'refs/remotes/origin/Release': is at** ed55abf2446a1a39b38d779b11065bb25f56ee08 but expected 3aece492c08f6c984bb2c5bb8fbf5b2d2b6a7967

From https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/dws-attendance

! 3aece49..4b8e1a7 Release -> origin/Release

(unable to update local ref)

git gc --prune=now //working once

update-ref for branches having issue from master branch.

git update-ref -d refs/remotes/origin/<branch-name> //working

QueryFailedError: tables can have atmost 1600 columns

>Take a backup of the DB.

> Drop the DB.

>Import the backup.

Make a Transaction Table with multiple Relations without joining tables use ManyToOne

**Connecting To Postgres on server machine:**

1)Connect to server machine.

>sudo -i -u postgres

>psql

>Enterpassword:postgres

2) Postgres commands: <http://www.postgresqltutorial.com/psql-commands/>

>\c dbName ---------To connect to db

>\l -------------- List all DBs

>\dt ------------ List all tables in the db

>\q ------------ Exit from postgres cli

>\dn ---------- List all schemas

>\d tableName ------- Describe table

3) >logout ----Exit from postgres

**Entity Updated but not reflecting/ re-deploy server**

>start putty

>git pull

>npm run build

>npm run start

**Validation Failed for number in Path & Query parameter in Fastest Validator**

Give {type:’string’} for all query and path parameters and use parseInt() to retrieve them.

**Cannot read property 'databaseName' of undefined"**

Check for invalid column names inside the query .

**Data: Cannot read property 'connect' of undefined**

Cascade drop tables where error exists or remove the typeorm related errors in the entities

**Git Delete Branch**

*// Delete local branch*

>git branch -d <branch-name>

>git branch -D <branch-name>

:: -d aka –delete will delete branch only if the branch has no un-commited changes.

:: -D will forcefully delete the branch

*// Delete remote branch*

>git push origin:<branch-name>

>git push origin -d <branch-name>

:: Here, origin is the remote-name

**Git Create Branch and push to remote**

1. git checkout -b MyNewBranch
2. git push origin MyNewBranch //one time push
3. git push -u origin MyNewBranch // set upstream for re-use

## error: cannot lock ref 'refs/remotes/origin/ #5438

Hacky workaround I've been doing is a combination of:  
running: git gc --prune=now  
and deleteing .git/refs/remote/origin

Callback response data return issue.

Use callback response for other uses

//    const getData=()=>{

        //      return new Promise((resolve,reject)=>{

        //          sqs.getQueueUrl(params, async(err: any, data: any) => {

        //              if (err) {

        //                  // if queue not found create a new queue for the employee

        //                  resolve(await ctx.call('notification.createQueue', { queue\_name: params.QueueName }));

        //              } else {

        //                  resolve(data)

        //              }

        //          })

        //      })

        // }

        // return getData().then(function(data:any) {

        //    return data;

        //  }).catch(function(err) {

        //    console.log(err);

        //    return err

        //  });

        // async function myBackEndLogic() {

        //  try {

        //      const result = await getData();

        //      console.log(result);

        //      return result;

        //     //return JSON.parse(result) if you want

        //  } catch (error) {

        //      console.error(error);

        //      return error;

        //  }

        // }

        // return myBackEndLogic();

        //---------------------------------------------------------------

**Delete the redis db**

**Flushdb , flushall**

$ redis-cli

> select 0 # (or whichever namespace Sidekiq is using)

> keys \* # (just to get an idea of what you're working with)

> smembers queues

> lrange queue:queue\_name 0 -1 # (queue\_name must be your relevant queue)

> lrem queue:queue\_name -1 "payload"

**Merge Conflicts while merging Don’t want to continue…**

git merge –abort

**committed but not pushed**

git reset --hard HEAD~1

git init

**Log of all ref updates:**

git reflog