in GIT bash backup extens VSC

$ code --list-extensions | xargs -L 1 echo code --install-extension

code --install-extension abusaidm.html-snippets

code --install-extension christian-kohler.npm-intellisense

code --install-extension christian-kohler.path-intellisense

code --install-extension CoenraadS.bracket-pair-colorizer

code --install-extension dbaeumer.vscode-eslint

code --install-extension esbenp.prettier-vscode

code --install-extension humao.rest-client

code --install-extension octref.vetur

code --install-extension ritwickdey.LiveServer

code --install-extension Shan.code-settings-sync

code --install-extension steoates.autoimport

code --install-extension streetsidesoftware.code-spell-checker

code --install-extension vscode-icons-team.vscode-icons

## Task Explorer

## Bracket Pair Colorizer ???

# **Cài đặt node.js socket.io lên heroku server**

heroku login

git init

heroku create – OK - <https://secret-sierra-98823.herokuapp.com>

[**https://secret-mesa-60672.herokuapp.com/**](https://secret-mesa-60672.herokuapp.com/)

* Chuẩn bị

git add . ( có dấu chấm )

git commit -m "heroku first commit"

* bắt đầu đẩy lên

git push heroku master

* Kiểm tra file đã tãi

heroku run bash

* Tạo file **Procfile** trong thư mục gốc - web: node start.js
* Úp lại

git add Procfile

git commit -m "up Procfile file"

git push heroku master

* kích thoạt ứng dụng và các lệnh trong Procfile sẽ thực thi

heroku ps:scale web=1

* cách đơn giản để fix lỗi ứng dụng

heroku restart

* Push lại tất cả các file đã sửa đổi lên app một lần nữa.

--------------------------

---------------FOR-LẦN ĐẦU

git init

heroku create || update (new version)

heroku login

--------------------------

git add .

git commit -m "up all" - git config --global user.email "nghiatv@live.com"

git push orrigin master

git push heroku master

--------------------------

heroku restart Nhớ khởi động lại app.

heroku update

heroku ps:scale web=1 kiểm tra trạng thái của app

heroku ps

heroku open --Mở thử app trên web, bằng lệnh

heroku releases ( show vers cache)

git init : đây là command được sử dụng khi chúng ta muốn bắt đầu một repository mới

git clone : command này dùng để clone lại một project từ môt url (thường sử dụng khi anh em pull code từ dự án về từ trên gitlab hay github ) :

git status : đây chắc là câu lệnh thần thánh mà tất cả đều dùng, command này kiểm tra các file đã thay đổi trong project của bạn

git add : ( . ,\* Tất cả) command này dùng để add những file thay đổi trong project

git commit -m “comment” : Command này sẽ commits bất kỳ file nào bạn đã thêm bằng lệnh git add và cũng commits bất kỳ file nào bạn đã thay đổi kể trước đó:

git branch : Command này liệt kê tất cả các nhánh hiện tại ở local của bạn (

+ [tên branch]: tạo + -d [Xóa] ) + checkout [chuyển] + checkout -b [chuyển & tạo]

git push : [heroku master – origin master ]được sử dụng để đẩy các commit mới ở local lên remote. Nguồn để đẩy lên là nhánh mà con trỏ HEAD đang trỏ tới (nhánh làm việc).

git pull : Command này tìm và kết hợp kéo các file thay đổi trên remote từ xa về thư mục làm việc của bạn

$ heroku plugins:install heroku-builds

$ heroku builds:cache:purge -a appname ( Xóa cache )

$ git commit --allow-empty -m "Purge cache"

$ git push heroku master

Sau khi đăng ký 1 TK Netify :

Git chạy trong thư mục dự án:

npm install netlify-cli -g

netlify login (Bạn sẽ được dẫn vào trang của Netlify để Authorize)

netlify deploy (Bạn chọn dự án mới hoặc củ – chọn tên dự án & account )

Chọn đường dẫn thư mục build (dist)

netlify deploy –prod (Bạn xác nhận sau deploy thành công)

--------------------------

backup toàn bộ code có trên ứng dụng (heroku),

git clone git@heroku.com: secret-sierra-98823.git -o heroku

heroku logs --source app --tail => Kiểm tra lỗi ứng dụng

heroku logs --tail

--------------------------

heroku apps:destroy OR APPNAME : xóa remote hiện tại heroku apps:destroy mighty-reaches-78345

heroku git:clone -a cert-hust: clone repository về

heroku git:remote cert-hust: thêm 1 remote

heroku run php artisan migrate:refresh - git push heroku master

Create a file called **static.json** in the root of your project CLI-VUE

{ "root": "dist", "clean\_urls": true, "routes": { "/\*\*": "index.html" }

}

npm install -g @vue/cli - Nếu chưa có

git add .

git commit -m "Heroku config"

heroku local web

npm run build

heroku create

heroku buildpacks:add heroku/nodejs

heroku buildpacks:add https://github.com/heroku/heroku-buildpack-static

git push heroku master

heroku logs –-tail - Check error

heroku restart

---------------------------------------------------------

mkdir NewProject

CD NewProject

$ npm init –y

( Nếu có package.json chép vào, ngược lại thì )

$ npm install --save express body-parser cors mongoose nodemon

$ npm install --global @vue/cli -??? npm i @vue/cli

$ vue create frontend

$ CD dir frontend

$ npm install --save axios

$ vue add route

$ vue add vuetify

$ npm run serve

npm i @vue/cli

vue ui

git add .

git commit -m "up all"

git push heroku master

heroku open

Một số câu lệnh cơ bản IN **MONGO SHELL**

* **show dbs** : dùng để hiển thị danh sách các database đang tồn tại. Mặc định ban đầu sẽ có 3 database của hệ thống là: admin, config và local.
* **show collections** : hiển thị các collection đang tồn tại
* **db** : hiển thị danh sách database đang sử dụng. Vì bạn chưa có database nào nên mặc định database mặc định cho bạn sẽ là test.
* **use DATABASE\_NAME** : chuyển sang sử dụng database với tên là DATABASE\_NAME (tên này tùy bạn chọn).
* **db.dropDatabase()** : dùng để xóa database hiện tại
* **db.createCollection(name, options)** : dùng để tạo mới một Collection trong MongoDB (cái này tương tự như Table trong MySQL).

Cách select, insert, update collections .Mọi thứ bắt đầu từ db.<collections>...

* **db.COLLECTION\_NAME.insert(document)** : thêm mới một Document vào MongoDB (cái này tương như như Row trong MySQL)
* **db.COLLECTION\_NAME.save(document)** : thêm mới hoặc cập nhật một document trong collection

+ login mongo shell

mongo "mongodb+srv://cluster0-ql9ch.mongodb.net/test" --username nghiatv

+ login connect –java – mongo compass

MongoClient mongoClient = MongoClients.create(

"mongodb+srv://nghiatv:Tranmeji1@cluster0-ql9ch.mongodb.net/test?retryWrites=true&w=majority");

MongoDatabase database = mongoClient.getDatabase("test");

+ backup

mongodump --host Cluster0-shard-0/cluster0-shard-00-00-ql9ch.mongodb.net:27017,cluster0-shard-00-01-ql9ch.mongodb.net:27017,cluster0-shard-00-02-ql9ch.mongodb.net:27017 --ssl --username nghiatv --password Tranmeji1 --authenticationDatabase admin --db <test>

+ Export

mongoexport --host Cluster0-shard-0/cluster0-shard-00-00-ql9ch.mongodb.net:27017,cluster0-shard-00-01-ql9ch.mongodb.net:27017,cluster0-shard-00-02-ql9ch.mongodb.net:27017 --ssl --username nghiatv --password Tranmeji1 --authenticationDatabase admin --db <test> --collection <employees> --type <json> --out <employees.js>

mongostat --host Cluster0-shard-0/cluster0-shard-00-00-ql9ch.mongodb.net:27017,cluster0-shard-00-01-ql9ch.mongodb.net:27017,cluster0-shard-00-02-ql9ch.mongodb.net:27017 --ssl --username nghiatv --password Tranmeji1 --authenticationDatabase admin

mongostat --host Cluster0-shard-0/cluster0-shard-00-00-ql9ch.mongodb.net:27017,cluster0-shard-00-01-ql9ch.mongodb.net:27017,cluster0-shard-00-02-ql9ch.mongodb.net:27017 --ssl --username nghiatv --password <Tranmeji1> --authenticationDatabase admin

Các trường số điện thoại bị format thành number và tự loại bỏ số 0 ở đầu, trong khi trường dữ liệu có dạng như 12/20 bị format thành định dạng ngày tháng.  
Để giữ nguyên dạng text cho các ô này, ta thêm thuộc tính CSS cho ô đó trong table như sau:

&lt;td style='mso-number-format:"\@"'&gt;0912222222&lt;/td&gt;

Tương tự như vậy ta có nhiều giá trị khác có thể thiết lập cho thuộc tính mso-number-format để định dạng excel cell theo ý muốn. Các bạn tham khảo bảng sau và có thể tự thử nghiệm để hiểu rõ hơn.

mso-number-format:"0" // NO Decimals

mso-number-format:"0\.000" // 3 Decimals

mso-number-format:"\#\,\#\#0\.000" // Comma with 3 dec

mso-number-format:"mm\/dd\/yy" // Date7

mso-number-format:"mmmm\ d\,\ yyyy" // Date9

mso-number-format:"m\/d\/yy\ h\:mm\ AM\/PM" // D -T AMPM

mso-number-format:"Short Date" // 01/03/1998

mso-number-format:"Medium Date" // 01-mar-98

mso-number-format:"d\-mmm\-yyyy" // 01-mar-1998

mso-number-format:"Short Time" // 5:16

mso-number-format:"Medium Time" // 5:16 am

mso-number-format:"Long Time" // 5:16:21:00

mso-number-format:"Percent" // Percent - two decimals

mso-number-format:"0%" // Percent - no decimals

mso-number-format:"0\.E+00" // Scientific Notation

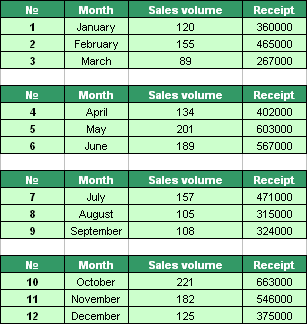
mso-number-format:"\@" // Text

mso-number-format:"\#\ ???\/???" // Fractions - up to 3 digits (312/943)

mso-number-format:"\0022£\0022\#\,\#\#0\.00" // £12.76

mso-number-format:"\#\,\#\#0\.00\_ \;\[Red\]\-\#\,\#\#0\.00\ " // 2 decimals, negative numbers in red and signed (1.56 -1.56)

Trong trường hợp bạn muốn table của bạn khi export ra excel màu mè một chút như hình dưới đây, giải pháp cho bạn chính là sử dụng inline CSS cho các thẻ th, td (tương ứng với các cell trong excel)



Ví dụ:

&lt;th width="1%" style="background-color: #E4FFFA; border: 1px solid #f4f4f4; font-size: 14px"&gt;

&lt;input type="checkbox" class="check-all"&gt;

&lt;/th&gt;

npm install -g @vue/cli

vue create demo

vue ui

npm run serve – vào browser at: localhost:8080 sẽ hiện logo VUEjs

(Đã cài xong VueJS cơ bản )

Cd demo

npm install bootstrap-vue --save

## **Some Mathematical Symbols Supported by HTML**

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| ∀ | &#8704; | &forall; | FOR ALL |
| ∂ | &#8706; | &part; | PARTIAL DIFFERENTIAL |
| ∃ | &#8707; | &exist; | THERE EXISTS |
| ∅ | &#8709; | &empty; | EMPTY SETS |
| ∇ | &#8711; | &nabla; | NABLA |
| ∈ | &#8712; | &isin; | ELEMENT OF |
| ∉ | &#8713; | &notin; | NOT AN ELEMENT OF |
| ∋ | &#8715; | &ni; | CONTAINS AS MEMBER |
| ∏ | &#8719; | &prod; | N-ARY PRODUCT |
| ∑ | &#8721; | &sum; | N-ARY SUMMATION |

[Full Math Reference](https://www.w3schools.com/charsets/ref_utf_math.asp)

## **Some Greek Letters Supported by HTML**

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| Α | &#913; | &Alpha; | GREEK CAPITAL LETTER ALPHA |
| Β | &#914; | &Beta; | GREEK CAPITAL LETTER BETA |
| Γ | &#915; | &Gamma; | GREEK CAPITAL LETTER GAMMA |
| Δ | &#916; | &Delta; | GREEK CAPITAL LETTER DELTA |
| Ε | &#917; | &Epsilon; | GREEK CAPITAL LETTER EPSILON |
| Ζ | &#918; | &Zeta; | GREEK CAPITAL LETTER ZETA |

[Full Greek Reference](https://www.w3schools.com/charsets/ref_utf_greek.asp)

## **Some Other Entities Supported by HTML**

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| © | &#169; | &copy; | COPYRIGHT SIGN |
| ® | &#174; | &reg; | REGISTERED SIGN |
| € | &#8364; | &euro; | EURO SIGN |
| ™ | &#8482; | &trade; | TRADEMARK |
| ← | &#8592; | &larr; | LEFTWARDS ARROW |
| ↑ | &#8593; | &uarr; | UPWARDS ARROW |
| → | &#8594; | &rarr; | RIGHTWARDS ARROW |
| ↓ | &#8595; | &darr; | DOWNWARDS ARROW |
| ♠ | &#9824; | &spades; | BLACK SPADE SUIT |
| ♣ | &#9827; | &clubs; | BLACK CLUB SUIT |
| ♥ | &#9829; | &hearts; | BLACK HEART SUIT |
| ♦ | &#9830; | &diams; | BLACK DIAMOND SUIT |

mysql://

b85c8cef50ea15

8f210f06

us-cdbr-east-02.cleardb.com

heroku\_cb325f7993da071

heroku addons:create cleardb:ignite

heroku config:get CLEARDB\_DATABASE\_URL

heroku addons:create jawsdb

heroku config:get JAWSDB\_URL

heroku addons:destroy jawsdb

select

s.schema\_name

,sp.grantee user

,cast(round(sum(coalesce(t.data\_length + t.index\_length, 0)) / 1024 / 1024, 3) as char) db\_size\_mb

,sp.has\_insert

from

information\_schema.schemata s

inner join

information\_schema.tables t on s.schema\_name = t.table\_schema

inner join (

select

spi.grantee

,spi.table\_schema

,max(

case

when spi.privilege\_type = 'INSERT' then 1

else 0

end

) has\_insert

from

information\_schema.schema\_privileges spi

group by

spi.grantee

,spi.table\_schema

) sp on s.schema\_name = sp.table\_schema

group by

s.schema\_name

,sp.grantee

,sp.has\_insert;

UPDATE customer SET company = "11CƠ SỞ KHẮC DẤU - PHÒNG CS QLHC VỀ TTXH", address = "459 Trần Hưng Đạo - Quận 1 - HCM", maso = "03001336422", ghichu = "Khắc dấu" WHERE id=17256

Copy Heroku Git URL<https://git.heroku.com/nghiatv.git>, trở lại git bash, clone nó về máy tính của mình theo đường link đó :

Git clone https://git.heroku.com/nghiatv.git

Truy cập vào thư mục vừa clone về, tạo file **composer.json** và copy cùng với hết source của bạn bỏ vào trong thư mục đó.

Gõ tiếp các lệnh để đưa code của bạn lên

**git add .**

**git commit -m "First commit"**

**git push**

Heroku sử dụng  **JawsDB MySQL**  
Trên Heroku bạn phải cài đặt addon này vào , chọn Tab Setting , gõ từ khóa vào khung tìm kiếm bên dưới và tiến hành cài đặt:

heroku config:get JAWSDB\_URL

mysql://

gez7ttzaugbqgm2s

hntfnck0shyhzjqj

bbj31ma8tye2kagi.cbetxkdyhwsb.us-east-1.rds.amazonaws.com

3306

hmjxh4of7e8f1fbc

| **Grants for gez7ttzaugbqgm2s@%** |  |
| --- | --- |
| GRANT USAGE ON \*.\* TO 'gez7ttzaugbqgm2s'@'%' |  |
| GRANT SELECT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON `hmjxh4of7e8f1fbc`.\* TO 'gez7ttzaugbqgm2s'@'%' |  |

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO `customer` (`id`, `fullname`, `name`, `company`, `address`, `email`, `phone1`, `phone2`, `fax1`, `ghichu`, `group\_id`, `maso`, `account`, `bank`, `citibank`, `makhach`) [VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) (NULL, '', '', 'CƠ SỞ KHẮC DẤU - PHÒNG CS QLHC VỀ TTXH', '459 Trần Hưng Đạo - Quận 1 - HCM', '', '', '', '', 'Khắc dấu', '', '03001336422', '', '', '', '');

heroku run bash

df -h /tmp

**Hướng dẫn 1: Hãy thử sử dụng branch!**

# 0. Chuẩn bị trước

Đầu tiên sẽ tạo thư mục mới, tạo repository trống ở đó. Ở đây, đã tạo thư mục gọi là tutorial.

$ **mkdir tutorial**

$ **cd tutorial**

$ **git init**

Initialized empty Git repository in /Users/eguchi/Desktop/tutorial/.git/

Trong thư mục tutorial, tạo file với tên myfile.txt, rồi commit lên.

**Đến cả con khỉ cũng hiểu được lệnh Git**

$ **git add myfile.txt**

$ **git commit -m "first commit"**

[master (root-commit) a73ae49] first commit

1 files changed, 1 insertions(+), 0 deletions(-)

create mode 100644 myfile.txt

**Hướng dẫn 1: Hãy thử sử dụng branch!**

# 1. Tạo branch

Tại đây, tạo branch với tên issue1.

Branch có thể tạo bằng lệnh branch.

$ git branch <branchname>

Ở đây, hãy thử tạo branch với tên issue1.

$ **git branch issue1**

Khi thực hiện lệnh branch mà không chỉ định tham số, thì có thể hiển thị danh sách các branch. Ở đầu có dấu \* là branch hiện tại.

$ **git branch**

issue1

\* master

Lịch sử tại thời điểm này như sau.



**Hướng dẫn 1: Hãy thử sử dụng branch!**

# 2. Chuyển đổi branch

Để thêm commit vào branch issue1 đã tạo mới, thì cần checkout branch issue1.

Checkout branch sẽ thực hiện bằng lệnh checkout.

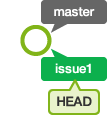
$ git checkout <branch>

Sẽ checkout branch issue1.

$ **git checkout issue1**

Switched to branch 'issue1'

Lịch sử tại thời điểm này như sau.



Note

Nếu chỉ định lựa chọn -b trong lệnh checkout rồi thực hiện, thì có thể tóm gọn thực hiện cả tạo branch và checkout.

$ git checkout -b <branch>

Khi commit với trạng thái đã checkout branch issue1, lịch sử sẽ được ghi vào branch issue1. Sau khi thêm giải thích lệnh add vào myfile.txt hãy thử commit lên.

Đến cả con khỉ cũng hiểu được lệnh Git

**add: Đăng ký thay đổi trong index**

$ **git add myfile.txt**

$ **git commit -m "Thêm giải thích add"**

[issue1 b2b23c4] Thêm giải thích add

1 files changed, 1 insertions(+), 0 deletions(-)

Lịch sử tại thời điểm này như sau.



**Hướng dẫn 1: Hãy thử sử dụng branch!**

# 4. Xóa branch

Vì nội dung branch issue1 đã được tích hợp thành công vào master nên hãy thử xóa đi.

Để xóa branch thì hãy chỉ định lựa chọn -d trong lệnh branch rồi thực hiện.

$ git branch -d <branchname>

Để xóa branch issue1, hãy thực hiện lệnh sau.

$ **git branch -d issue1**

Deleted branch issue1 (was b2b23c4).

Như thế, branch issue1 đã được xóa. Bằng lệnh branch hãy kiểm tra thử xem branch đã được xóa chưa.

$ **git branch**

\* master