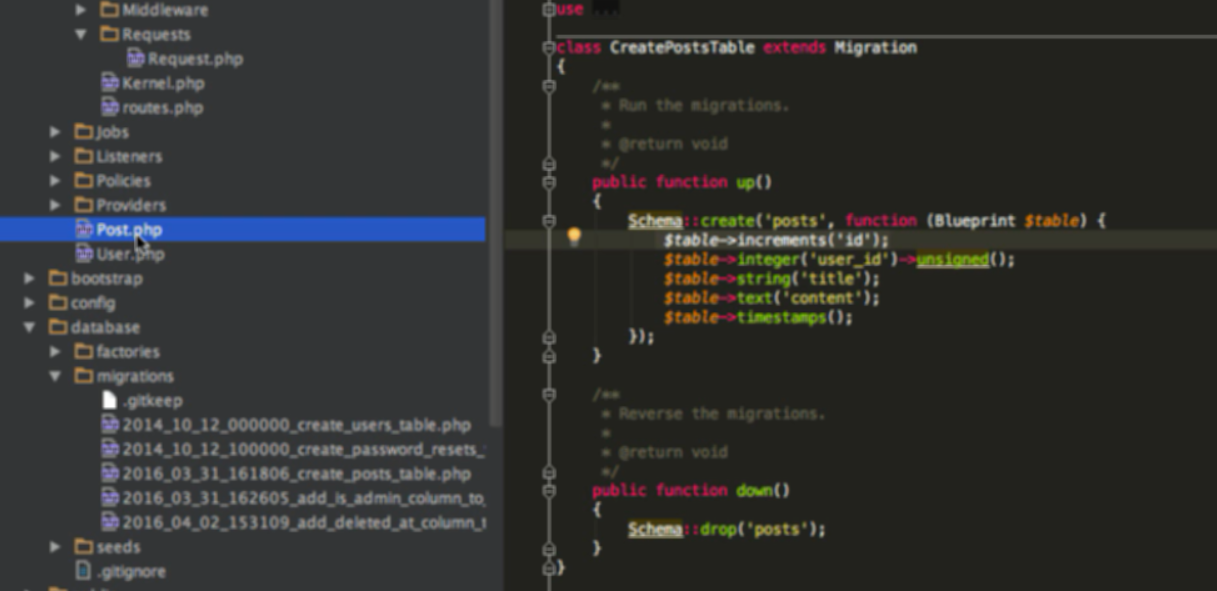
2. One to One relationship

Ví dụ mỗi người sở hữu 1 cái điện thoại, từ người muốn truy xuất lấy thông tin của điện thoại



+ Trên User phải có định nghĩa column primary key (default là id) xem doc để xem cách custom primary key

+ trên Phone phải có column user\_id (mặc định có thể là cái khác)



Ta có model User và Phone

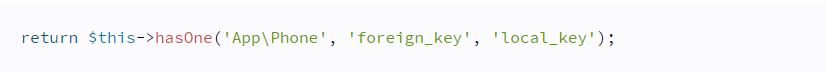
Trong model User, định nghĩa method phone: là method để truy cập và bảng phone, method này return như trên

hasOne ko có param2 thì mặc là user\_id

method này có tác dụng là nối khóa chính id (mặc định là id, có thể định nghĩa lại bởi người dùng) trên user với khóa ngoại trên Phone (mặc định là user\_id , có thể định nghĩa lại nếu pass param2 vào hàm hasOne)



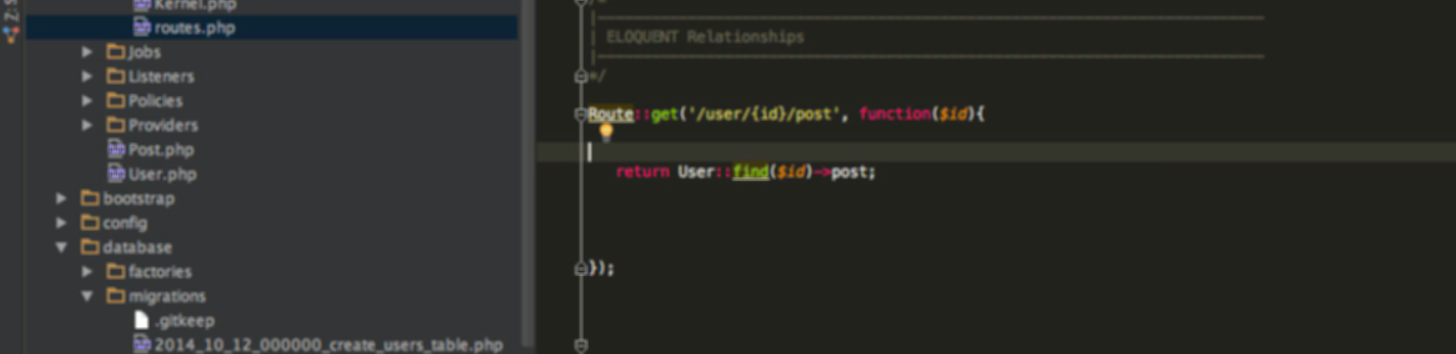
Trường hợp muốn local key là giá trị khác ngoài primary key (id hoặc cái đk định nghĩa lại):



Gọi Phone từ User



Khi đó lệnh này sẽ select ra phone có user\_id =1 ( nghĩa là lấy cái phone của user đó nhờ vào cặp khóa chính ngoại ) -> giống như . bên js

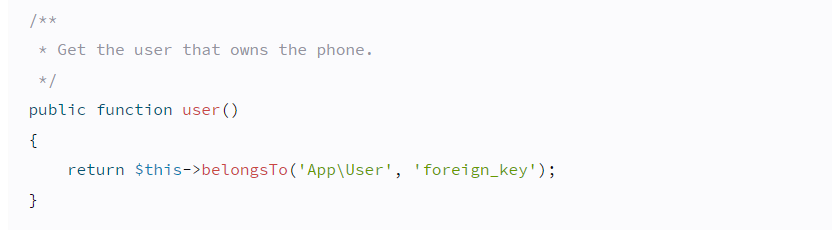


3. The inverse relation

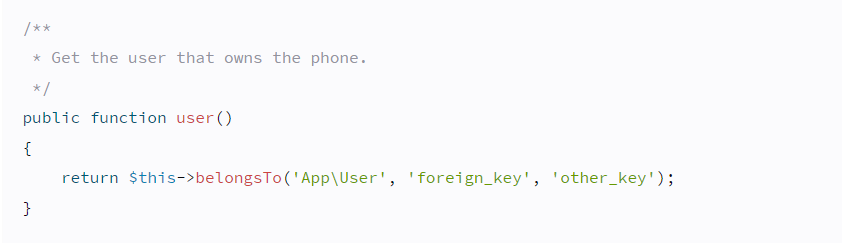
Điện thoại được sở hữu bởi 1 người



Mặc định user\_id của Phone sẽ map với id của User

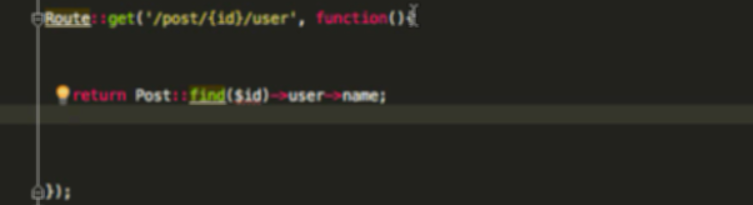


Tương tự nếu trên Phone, cái foreinkey mong muốn không phải là user\_id mà là cái khác thì định nghĩa như trên



Nếu không phải map vs primary key thì dùng thêm other\_key như trên

Lưu ý: đây là trường hợp one to one nên chỉ có 1 primary key vs 1 foreinkey nên đừng nghĩ về vụ nếu có nhiều primary hay foreinkey nữa, tào lao



Khi đó thì từ Post(Phone) có thể lấy dữ liệu từ User bằng method user vừa định nghĩa ở trên

Tên của khóa ngoại mặc định: tên model bị sở hữu\_id

Tên khóa chính mặc định: id

4. One to many relationship

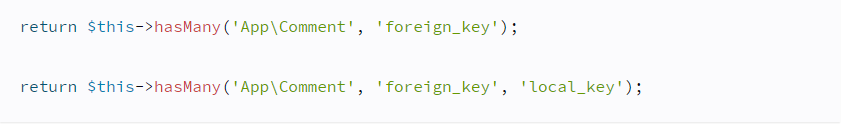
Là trường hợp một người sở hữu nhiều comment chả hạn



Mặc định map primary key vs foreinKey

primary key mặc định: id, hoặc primary đã được custom

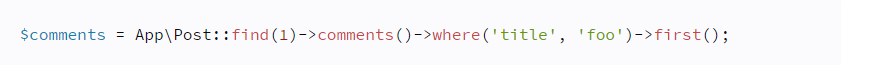
foreinKey: tên model owner \_id



Trường hợp muốn thay đổi



Nhắc lại méng: khi không làm gì thì dùng App/Post, còn khi trên đầu file có use App/Post thì dùng Post thôi



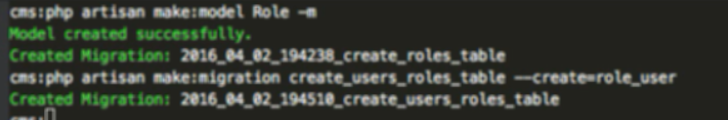
Gọi trên route hoặc controler

Defining The Inverse Of The Relation ( từ comment truy cập chủ của comment đó -tương tự có thể đọc doc)

6. Many to many relations

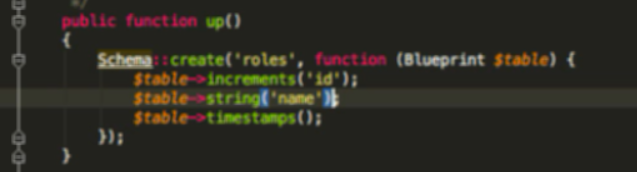
Một người có nhiều role, và một role có thể có nhiều người đảm nhiệm

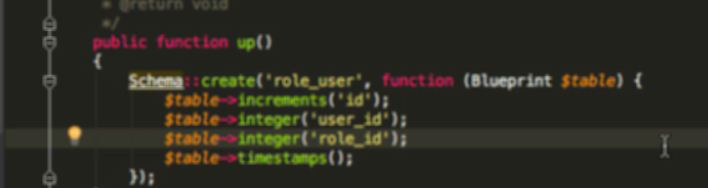
Có 3 table: user, role, role\_user



Tạo model role cùng migration

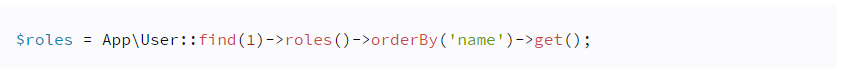
Tạo table role\_user cùng migrationg (table chứ ko có model, vì cái này là bảng trung gian, ko làm gì trong model cả)

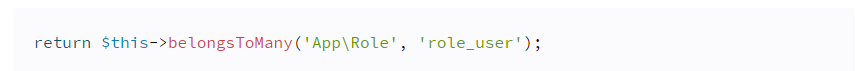




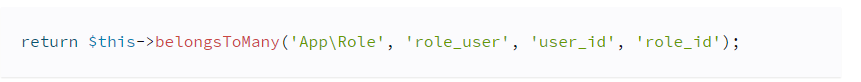






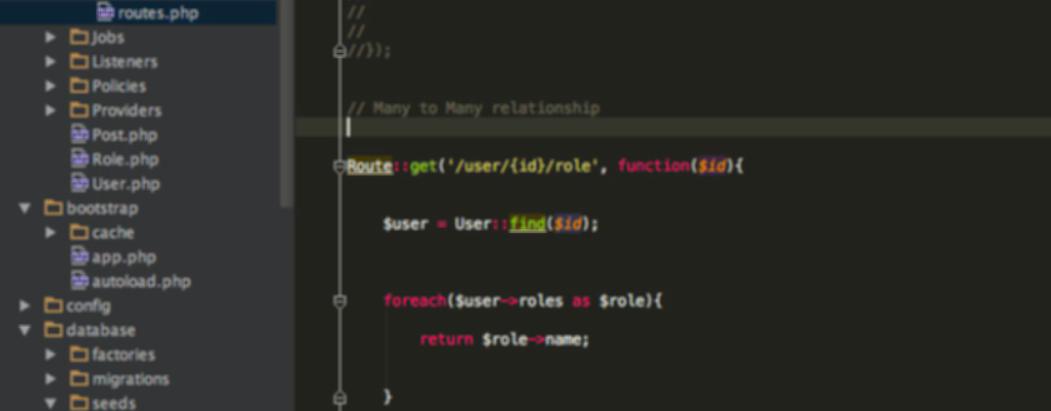


Tên của table trung gian sẽ tên của model 1 \_ tên model 2 theo thứ tự bảng chữ cái, tuy nhiên có thể tự chỉ ra nếu đặt tên khác

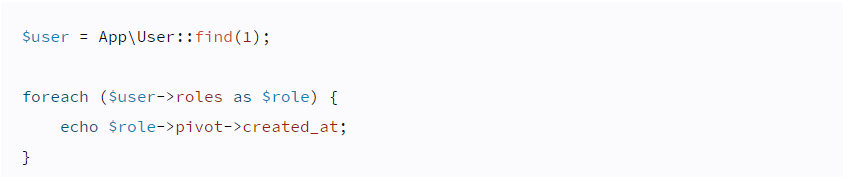


Tương tự với 2 cột trung gian tên model\_id

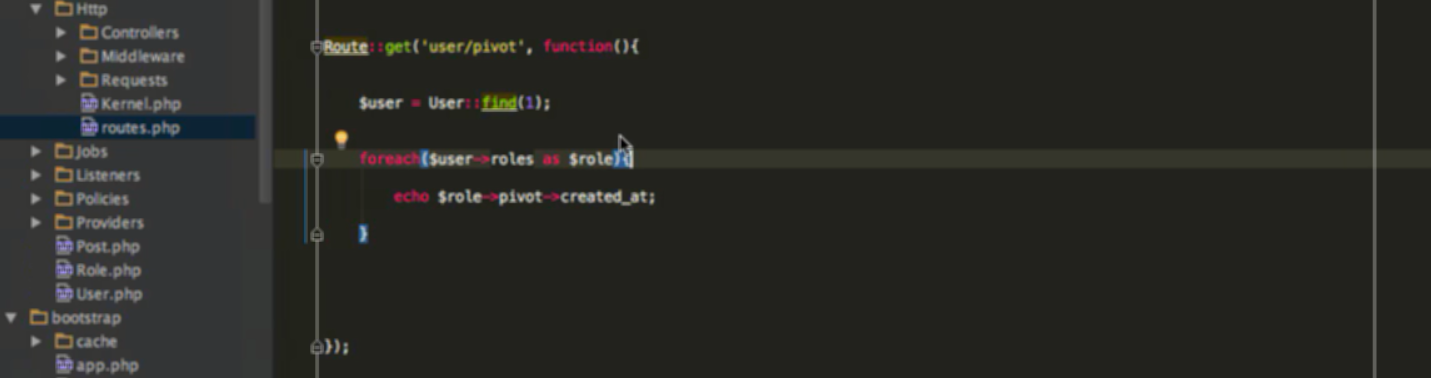
Một model có thể cùng có nhiều và nhiều loại relation 1 lúc



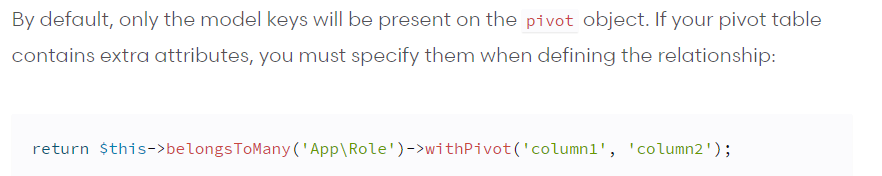
#### The Inverse Of The Relationship ( với many to many thì inverse giống cái thường, trên role định nghĩa giống method user tương tự role ở trên)

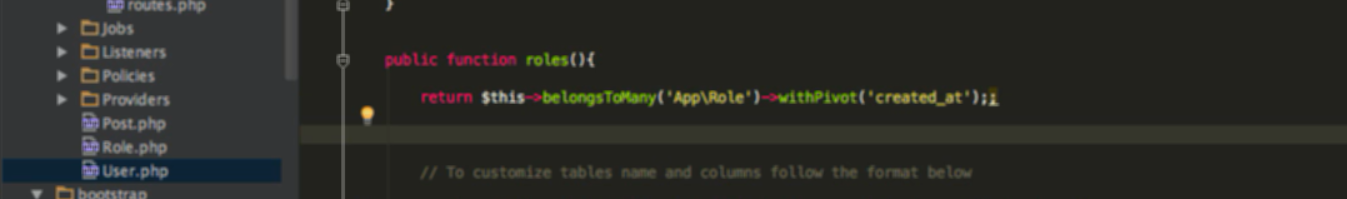


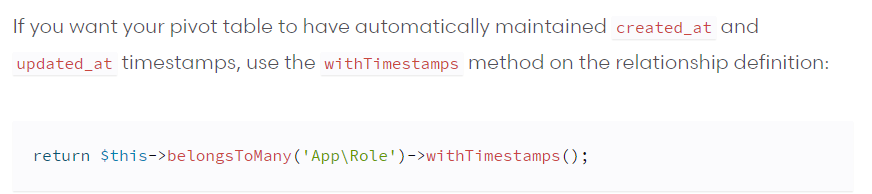
Truy cập table trung gian (gọi là pivot) , ở đây là many to many nên mỗi user sẽ có nhiều row -> lặp



Thay echo = return







9. Has many through relation



Dùng migrate để tạo các cột nha mai phen

Contries liên quan với user bới contry\_id

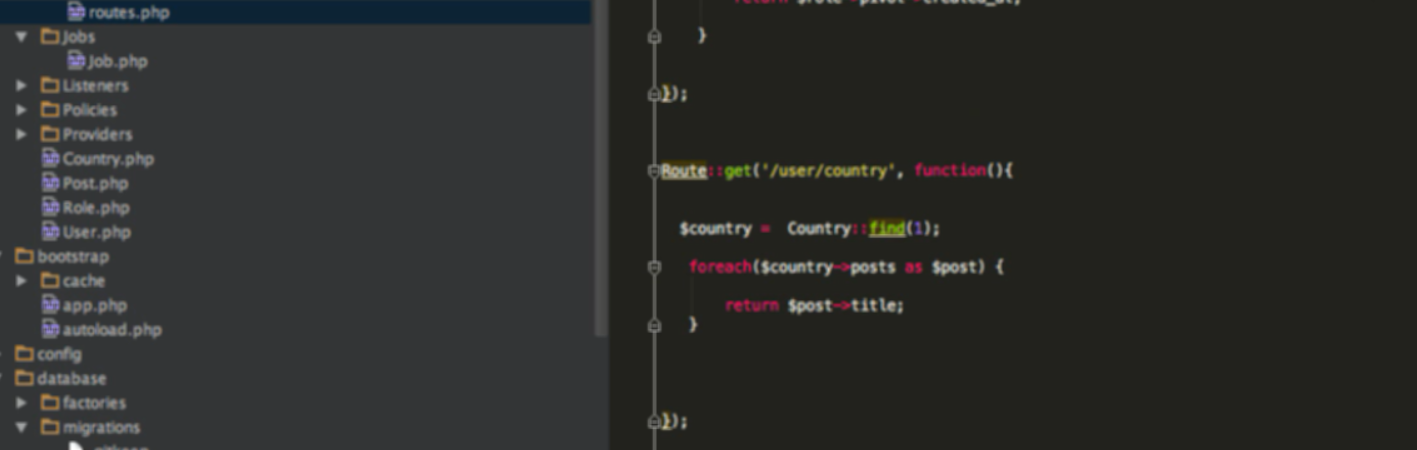
User liên quan post bởi user\_id

Tuy nhiên contries lại ko liên quan đến post, nhưng qua trung gian users, có thể từ contries truy cập post



Mặc định về các tên khóa liên kết như trên ( tên mode \_id ) tuy nhiên nếu muốn thay đổi thì pass giá trị vào theo thứ tự như sau





11. Polymorphic relation

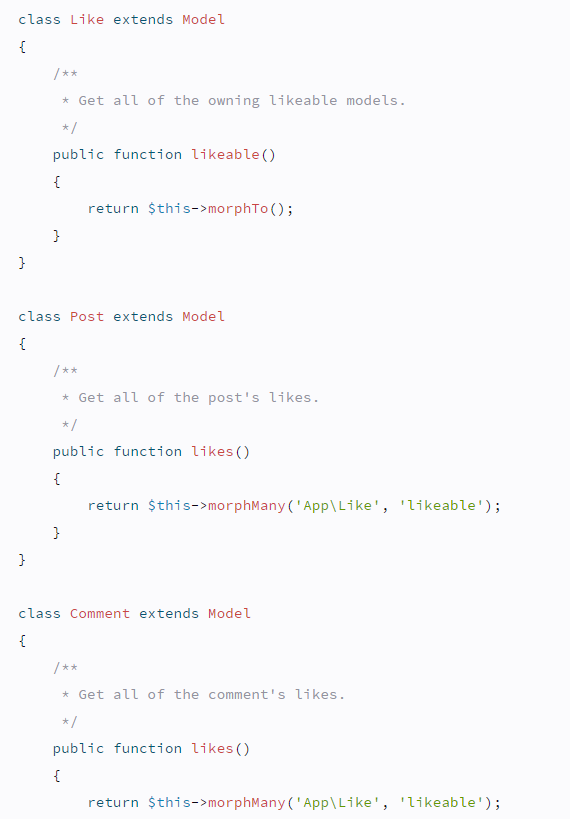


Khi một model có 2 model sở hữu ( bình thường là 1)

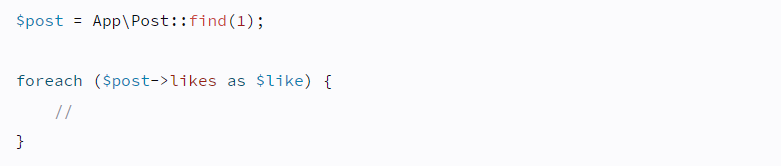
Ví dụ trong bài: bảng like sở hữu bởi posts, comments -> dùng bảng like để chứa like của post và comment

Likeable\_id là id của post hoặc comment được like tên này là tên của method ở belonger\_id

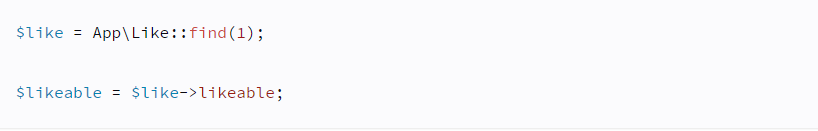
Likeable\_type là tên của class sở hữu like (Post hoặc Comment) like tên này là tên của method ở belonger\_id



Tương tự tạo các method

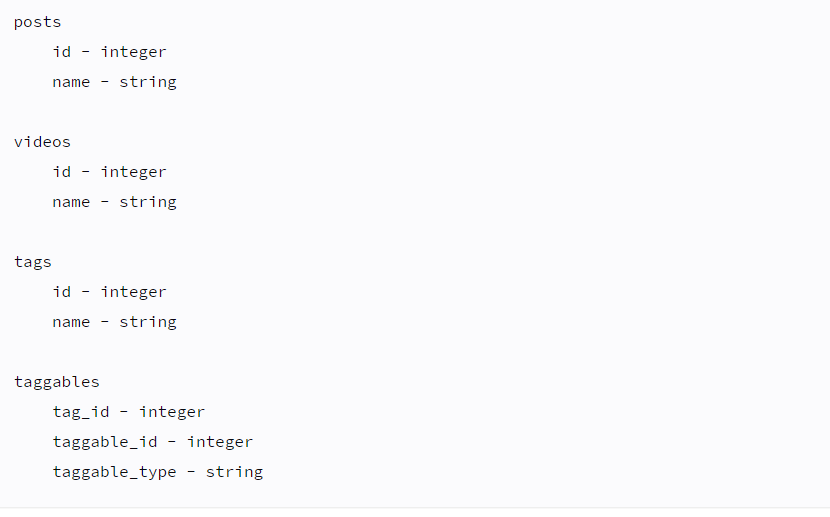


Từ owner truy cập belonger quá method likes



Từ belonger truye cập owner qua method likeable, việc truye cập comment hay post phụ thuộc vào likeable\_type

14. Polymorphic relation many to many



Tags được sở hữu bởi cả nhiều posts và nhiều videos

Mỗi post hoặc video lại có nhiều tag

Many to many dùng thêm bảng trung gian taggables

Phần còn lại đọc doc nha mai fen, ko khó đâu, nhưng mà coi video của cha nôi edwin mà ko đọc doc là khó ah