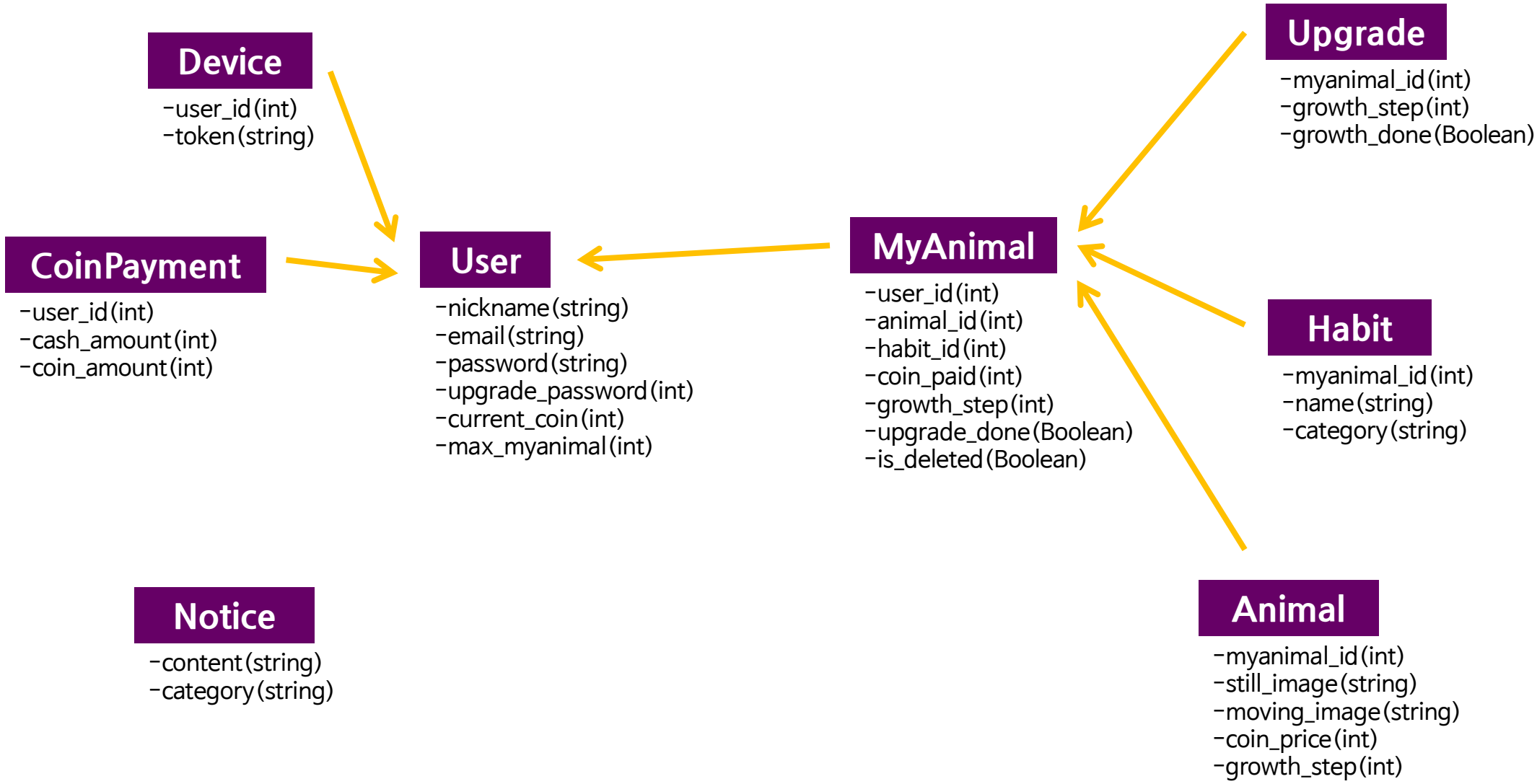

Device
-user_id(int)
-token(string)
CoinPayment
-user_id(int)
-cash_amount(int)
-coin_amount(int)
User
-nickname(string)
-email(string)
-password(string)
-upgrade_password(int)
-current_coin(int)
-max_myanimal(int)
MyAnimal
-user_id(int)
-animal_id(int)
-habit_id(int)
-coin_paid(int)
-growth_step(int)
-upgrade_done(Boolean)
-is_deleted(Boolean)
Upgrade
-myanimal_id(int)
-growth_step(int)
-growth_done(Boolean)
Habit
-myanimal_id(int)
-name(string)
-category(string)
Animal
-myanimal_id(int)
-still_image(string)
-moving_image(string)
-coin_price(int)
-growth_step(int)
Notice
-content(string)
-category(string)

SIMPLE GIT GUIDE(command)

1. Git hub에서 파일을 다운로드 할 건데, 파일 다운로드 할 위치에서 cmd창을 켠다 start command prompt with ruby
   cd /*local folder path*/

2. 로컬에서 작업한 후 git hub에 업데이트할 거니까 환경 설정하고 clone한다.
   git init
   git clone https://github.com/zzokokeic/barogagi1gibateam.git

3. 파일을 수정/업데이트한다!
   이 때, 실제 파일 다운로드는 /*local folder path*/ 안에 barogagi1gibateam 이름으로 완료됨

4. 파일 수정을 완료한 후 git hub에 push한다.
   /*local folder path*/barogagi1gibateam 위치에서 git status 하면 modified / new / delete 등으로 확인 가능,
   git add .
   git commit -m "메세지내용 - (최대한 작업한 내용 자세히 써줍시다)"
   git push origin master

DB Debugging

** rake db:drop

혹은 Schema.rb / dev.sqlite3 두개 파일 삭제

** rake db:migrate
** rake db:seed

이 때 seeds.rb는 barogagi1gibateam/db/ 안에 존재해야함
seeds.rb 는 공용으로 사용할 DB 미리 넣어 둘 파일

```
C:₩BRGG₩BRGG_BAR_TEAM₩barogagi1gibateam>irb
irb(main):001:0> require './controller.rb'
=> true
irb(main):002:0> h = Habit.first
D, [2018-11-12T00:42:55.062106 #8520] DEBUG -- :   Habit Load (0.5ms)  SELECT  "habits".*
","id" ASC LIMIT ?  [["LIMIT", 1]]
=> #<Habit id: 1, mission: "habitname", category: "habitcategory">
irb(main):003:0> h.category = "category_change_test"
=> "category_change_test"
irb(main):004:0> h.save
D, [2018-11-12T00:43:25.267721 #8520] DEBUG -- :    (0.0ms)  begin transaction
D, [2018-11-12T00:43:25.276209 #8520] DEBUG -- :   Habit Update (2.4ms)  UPDATE "habits"
s"."id" = ?  [["category", "category_change_test"], ["id", 1]]
D, [2018-11-12T00:43:25.286625 #8520] DEBUG -- :    (6.4ms)  commit transaction
=> true
```

** irb 로 db update 하려면

irb
require ‘./controller.rb’

타이핑 후 테스트
/* DB TEST */
DB.save필수!

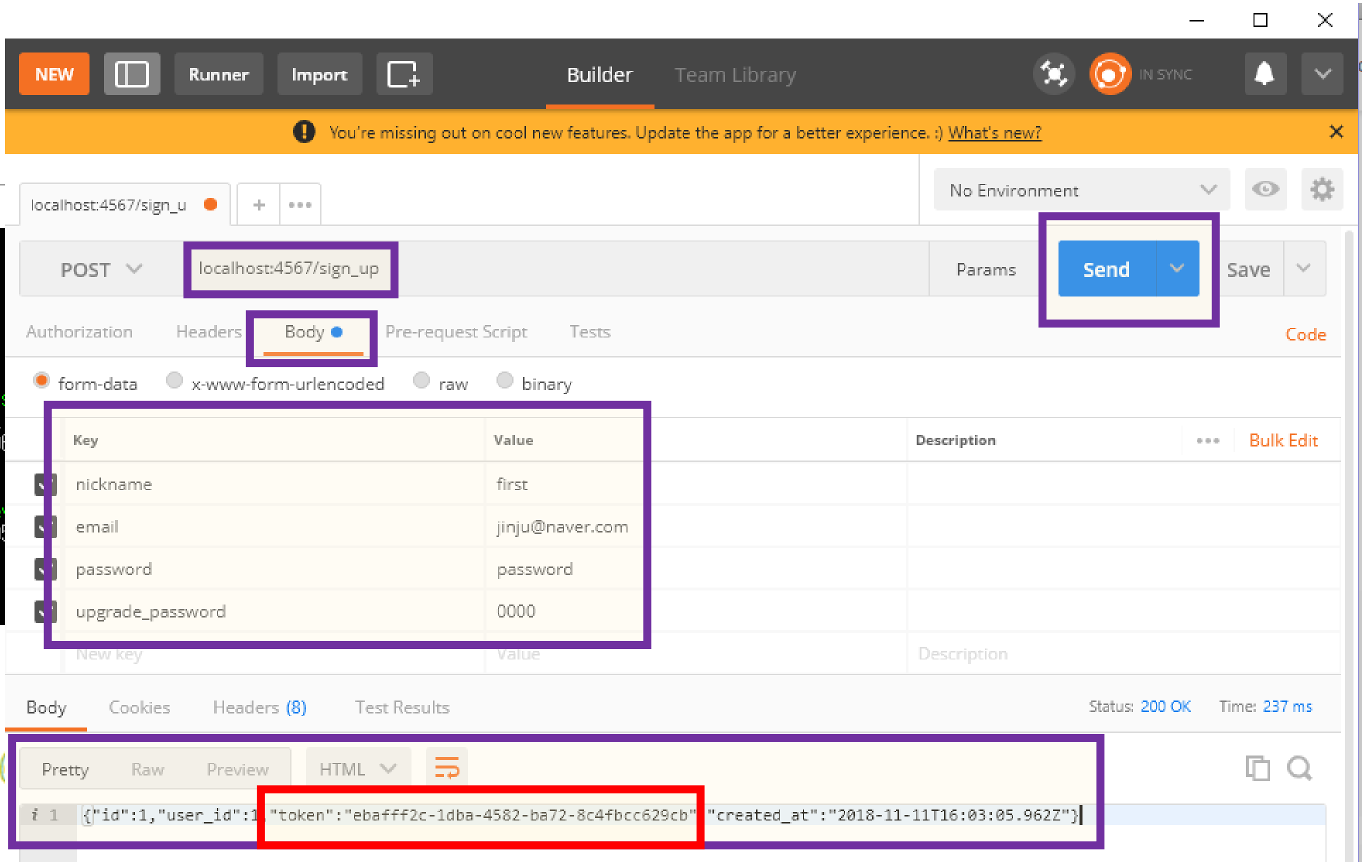
DB Debugging

Chrome에서 postman이라는 확장앱 사용해서 DB값 동작확인
유저시나리오 대로 확인도 반드시 해야 하지만 예외처리를 위한 다른 값을 넣어보는 작업도 필수 -〉 controller.rb에 반영

먼저 ruby controller.rb 커맨드로 local host 동작시키고
Postman 앱에서 params 값들을 지정
Send!

아래처럼 controller.rb 에 puts를 넣어서 테스트해볼 수 도 있음

```
puts "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
puts params["upgrade_password"]
puts user.upgrade_password
```



TOKEN -〉 Device테스트시 복사붙여넣기로 사용

DB Oriented Development

```ruby
get '/device' #로그인된 사람 가져오기

    user = USER.
    device = user.device
    device.to_json
end

post '/device' #로그인
    user = User.where(id pwd)
    device = Device.new
    device.user = user
    device.save
    device.to_json

end

delete '/device' #로그아웃
end

get '/aminal'
    # options = parmas["option"]
    Aniaml.where(option => params["option"]) GRAPHQL
end
```

RESTFUL 방식
: 미팅앱엔 이 방식으로
서버와 클라이언트의 로드를 분산하는 게 좋을 것 같다고 하심

서버에서 DB관련 method를 아래 방식으로 매칭시켜
최대한 로드를 줄임

- GET / POST / PUT / DELETE

대신 클라이언트에서 로직까지 함께 구현해야 함
(DB로직을 모두 알아야함 -> 옵션으로 바로 params전달)