

## [k-holdem-rails] local server

hoyoul

<2023-10-16 Mon>

### 개요

k-holdem을 처음 설치하고 local에서 실행하는 것까지 다룬다. 대충 rough하게 정리한 note지만, gira가 만들어지면 다시 정리할것이다. 여튼 처음 설치하는 사람들에게 guide가 될 수 있을 것이다.

### git project clone

해당 repo를 clone한다.

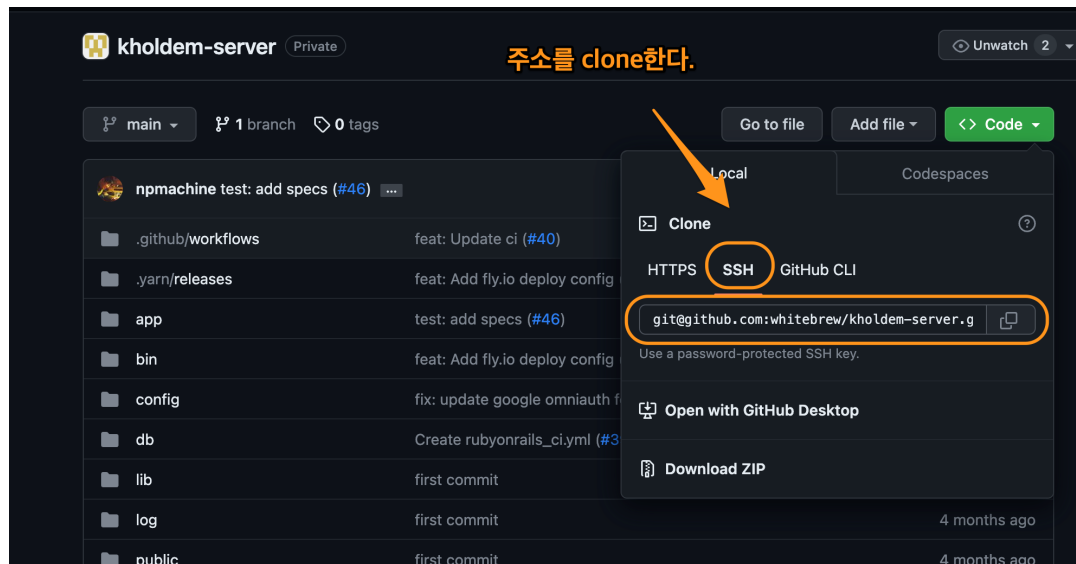


그림 1: clone

아래 처럼 clone한다.

```
git clone git@github.com:whitebrew/kholdem-server.git
```

PS: 나는 postfix multiple github을 사용하기 때문에, github.com에 postfix를 붙여준다. email에 postfix붙여주는 것과 비슷하다. github.com은 하나의 host 주소인데, 여러개의 account를 사용하기 때문에 postfix를 붙여줘서 구분한다.

```
git clone git@github.com:hoyoul:whitebrew/kholdem-server.git
```

## local실행을 위한 prerequisites

### asdf 설치

참고, asdf는 node.js와 ruby의 version관리를 통합적으로 할 수 있다. rvm이나 rbenv 을 해도 된다. 나는 이번에 asdf로 ruby버전을 관리하기로 했다. python이나 node도 관리할 수 있기 때문이다. asdf 공식 page에 나온 방식으로 설치할 수도 있고, gorails에서 설치한 방법으로 설치할 수도 있다. 둘 중 하나의 방식을 선택한다. 나는 asdf공식 page방식으로 설치했다.

### asdf 공식 page

prerequisite이 있다.

```
brew install coreutils curl git
```

그리고 asdf를 설치한다.

```
brew install asdf
```

asdf를 shell path에 등록한다.

```
echo -e "\n. $(brew --prefix asdf)/libexec/asdf.sh" >> ${ZDOTDIR:-~}/.zshrc
```

### gorails

```
cd
git clone https://github.com/excid3/asdf.git ~/.asdf
echo '. "$HOME/.asdf/asdf.sh"' >> ~/.zshrc
echo '. "$HOME/.asdf/completions/asdf.bash"' >> ~/.zshrc
echo 'legacy_version_file = yes' >> ~/.asdfrc
exec $SHELL
```

### ruby 설치

#### plugin 설정

asdf에서 plugin을 설치한다. asdf가 언어에 대한 통합버전 관리라서 언어를 plugin으로 지정해줘야 한다.

```
asdf plugin add ruby
```

## **ruby version 설치**

이제 asdf로 ruby를 설치할 수 있다. gorails에선 3.2.2를 추천한다. 왜냐면 rails 7.0.6을 안정화 버전으로 보고, 해당 rails에 맞는 ruby버전이 3.2.2기 때문이다. gemfile에 보면, k-holdem project는 3.2.1이다. 3.2.1, 3.2.2 둘다 설치하자.

```
asdf install ruby 3.2.2
asdf install ruby 3.2.1
```

- 사용할 ruby를 system에 설정하기로 하자. 어차피 3.2.1만 사용할거라서 system에 설치하기로 한다.

```
asdf global ruby 3.2.1
# Update to the latest Rubygems version
gem update --system
```

## **ruby version 확인**

제대로 설치되었는지 ruby version을 확인한다.

```
which ruby
#=> /Users/username/.asdf/shims/ruby
ruby -v
#=> 3.2.2
```

## **rails 설치**

gorails는 7.0.6을 추천하지만, project는 7.0.8 이상을 설치하기 때문에 맞춰서 설치하자.

```
gem install rails -v 7.0.8
```

만일 permission문제가 있다면, sudo를 붙여서 사용한다. 나는 문제가 없었다.

```
rails -v
```

제대로 설치되었는지 확인 해보자. 7.0.8이 나오면 된다.

## **database 설정**

### **postgresql**

- postgresql 설치와 실행 gemfile을 보면 pg가 설치되어 있고, version은 1.0 이상이어서 상관없이 설치해도 된다. 현재 14가 stable버전인듯 하다.

```
brew tap homebrew/core
brew install postgresql
```

server는 db를 이용하기 때문에 시작과 함께 실행 시킨다. 나는 원래 직접 필요할 때만 실행 시켰지만, 상관없다. 아래와 같이 하는게 편하다.

```
brew services start postgresql
```

## local에서 실행

local에서 rails s로 바로 server를 실행시키면 안된다. bundler후에 db처리를 해 줘야 한다. db가 텅비워져 있기 때문이다.

## bundler

project의 root에서

```
bundle install
```

하면 된다.

## db 처리

### database 생성


db에 database를 만든다. 만드는데 수동으로 만들어도 되지만, rails db:create로 자동으로 만들자. rake를 써도 상관없다. db:create는 rails root project의 config/database.yml에 지정된 database를 참조해서 database를 만든다. database 이름은 kholdem<sub>serverdevelopment</sub>다.

```
rails db:create
```

### schema load

처음 db를 setting하면 schema를 load하면 된다. 기존의 table이 있다면 migrate만 하면 되겠지만, 나는 어차피 database만 있고 table이 하나도 없기 때문에 schema:load를 해준다. schema는 table layout의 최종버전으로 보면 된다.

```
rails db:schema:load
```



```
vi database.yml

# Configure Using Gemfile
# gem "pg"
#
default: &default
  adapter: postgresql
  encoding: unicode
  # For details on connection pooling, see Rails configuration guide
  # https://guides.rubyonrails.org/configuring.html#database-pooling
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>

development:
  <<: *default
  database: kholdem_server_development
```

그림 2: db1

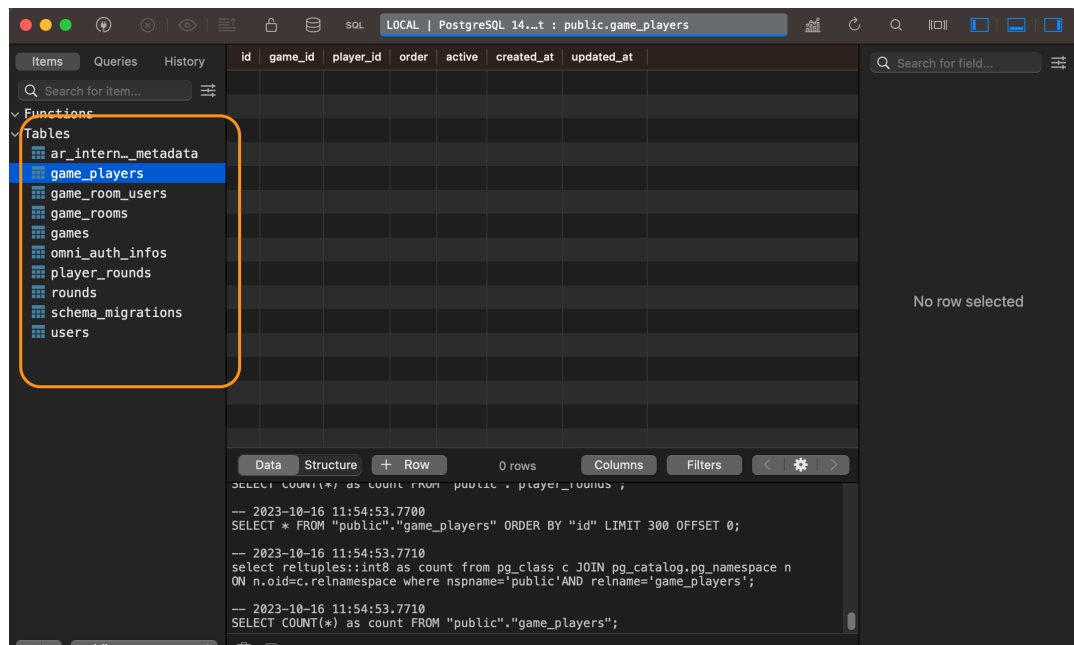


그림 3: table1

## db 확인

나는 mysql을 사용했기 때문에 querious라는 mysql전용 db client를 썼다. postgresql이라서 TablePlus로 확인했다.

record는 없지만, table이 만들어진것을 확인할 수 있다. fixture를 사용할 수도 있지만, 우선은 이것만 하고 넘어간다.

## 실행

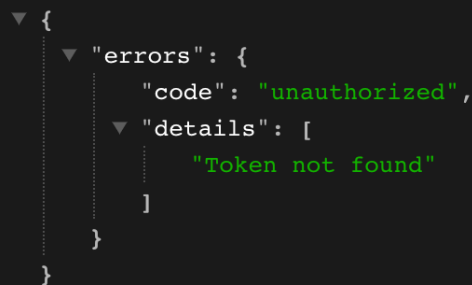
rails s



```
▶ rails s
=> Booting Puma
=> Rails 7.0.8 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.6.7 (ruby 3.2.1-p31) ("Birdie's Version")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 21966
* Listening on http://127.0.0.1:3000
* Listening on http://[::]:3000
Use Ctrl-C to stop
```

그림 4: server1

server가 local에서 실행되었음을 알 수 있다. browser에서 연결을 해보자.



```
{
  "errors": {
    "code": "unauthorized",
    "details": [
      "Token not found"
    ]
  }
}
```

그림 5: error1

Oauth문제 때문이다. routes.rb에 보면 omni-auth처리를 해주고 있다. rails 와 flutter에서 이부분을 좀더 봐야 할듯하다.