Tổng kết

* Config

Chứa các file config về app, database

* Exception

Định nghĩa các exception

* BaseException: định nghĩa các phương thức chung
* ExceptionHandler: định nghĩa các xử lý exception
* Exception: khai báo với hệ thống sử dụng file ExceptionHandle khi có exception
* Contracts

Định nghĩa các interface

* Database
* AbstractConnection: Khai báo abstract class với hàm tạo và phương thức kiểm tra các tham số cần thiết để kết nối đến csdl
* PDOConnection, MySQLiConneciton: parse tham số đầu vào tuy theo kiểu kết nối PDO hay MySQLi và tạo connection tới csdl
* QueryBuilder: Sau khi đã khởi tạo kết nối đết csdl, ta sẽ sử dụng connection này để thực hiện các query. Đây là 1 abstract class, định nghĩa các phương thức để ta có thể sử dụng method chaining. Ví dụ : table(‘reports’)->where(‘…’)->select(\*)->get(‘’); Các method này sẽ thực hiện parse các tham số truyền vào vào các biến trong class, sau đó các biến này sẽ được đọc để tạo thành 1 câu query dạng text . Ví dụ: ‘select \* from reports where …‘ . Trong phần này, ta cũng sử dụng statement (tư tưởng giống reactjs, gửi template lên và các lần sau sẽ chỉ cần truyền các tham số vào template) để giảm thiểu nội dung cần gửi tới csdl.
* MySQLBuilder, PDOBuilder: định nghĩa các phương thức còn lại tuỳ vào kiểu csdl: excute câu lệnh, beginTransaction, prepare cho statement, count(), get() , …
* Logger: Ghi logger, tạo file và ghi nội dung vào file tuỳ theo level
* Models: Tạo model report tương ứng với bảng reports gồm thuộc tính, method set-get
* Repository: Sử dụng query builder để thao tác đến csdl
* Helper
* Config.php : đọc dữ liệu từ file
* App.php : đọc dữ liệu về file config App thông qua hàm được tạo từ file Config,php
* Tests:
  + Test Unit: test model, test repostitory
  + Test Functional: test khi gọi api