제이프레임워크(JFramework)

https://github.com/javamon1174/jframe

개발 환경

- AWS(아마존 웹 서비스)
- AWS Route 53 – 도메인 연동(javamon.be)
- 운영체제 : Ubuntu 16.04
- 프로그래밍 언어 : PHP 7.0.15
- 데이터베이스 : Maria database 10.*
- 의존성 관리도구(Composer)

작업 환경

- macOS
- 개발 툴 : ATOM, Sequelpro
- VM : Virtual BOX
- BOX : ubuntu/trusty64
- VM 관리도구 : Vagrant
- Chrome developer tools

JframeWork

제이프레임워크?

- 동기

개인 프로젝트를 진행하면서 공통된 작업들을 반복적으로 하는데에 많은 시간이 소요되면서 이에 따른 "틀"의 필요성을 느꼈으며, CI와 라라벨을 사용하면서 느꼈던 필수적인 부분에 대한 집합을 고려하여 해당 프로젝트를 시작하게 되었습니다.

또한, 흔히 말하는 MVC 패턴을 직접 구현해보면서 PHP와 MVC 패턴에 대한 이해도를 높이고 싶었습니다.

- 준비물

1. PHP 7 이상
2. PHP 컴포저
3. PHP 모듈 - PDO
4. 마리아 DB 10.0.* 이상
5. APACHE - Rewrite, Override ON

ABOUT

제이 프레임워크는 굉장히 작습니다. 단순히, http 요청을 받고 그에 따른 MVC형태만 가지고 있지만,
추후에 리팩토링을 통해 "가장 필요함"에 중점을 두고 확장시킬 예정입니다.
또한, 작은 만큼 다양한 형태로 확장이 가능하다는 부분도 장점이라고 생각합니다.

- 제이프레임워크는

1. 디렉터리의 구조는 어플리케이션의 하위에 코어가 있는 구조인 라라벨과 유사합니다.

2. 라우트는 매우 심플하며, 아파치의 rewrite와 GET요청만을 활용하여 구성되었습니다.

3. 라라벨의 엘로퀀드 ORM 방식과 유사한 ORM을 가지고 있습니다.

4. 모델과 ORM은 PHP 모듈인 PDO로 구현되었으며, 트랜잭션을 활용한 commit, rollback이 가능합니다.

5. 누구든지 쉽게, 또는 바로 코어 클래스들을 수정/보완하여 커스터마이징이 가능합니다.

6. 내장된 함수를 통해, 몇줄 안되는 소스로 데이터베이스 테이블과의 입출력이 가능합니다.

발전 방향

- HTTP의 요청(POST, GET 등) 처리를 위한 클래스.

- 쿠키, 세션 등을 위한, 활용하기 위한 클래스.

- 배치프로세스(멀티 프로세싱 또는 멀티스레딩).

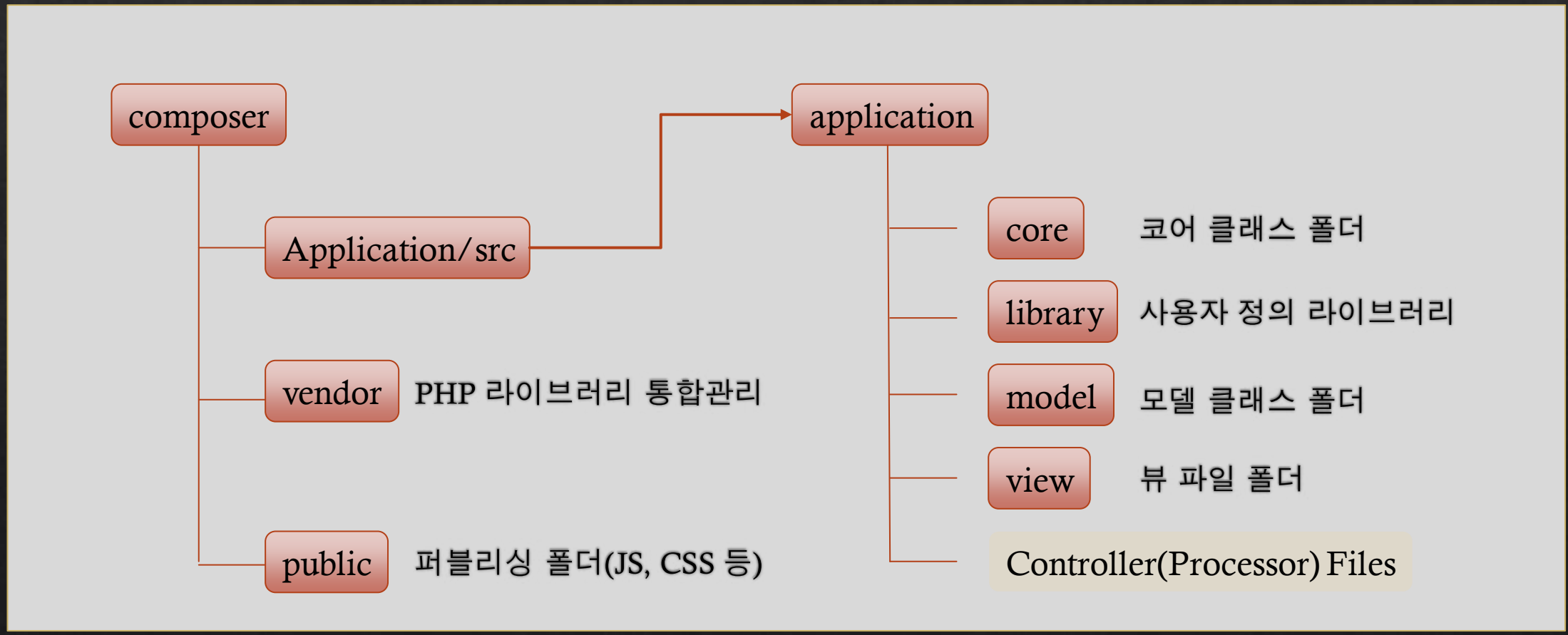
- 데이터베이스 스키마 마이그레이션.

- 라라벨의 artisan과 같은 스크립트 기반의 프로세스.

- 다양한 공격(인젝션 등)에 대응할 수 있는 보안.

구조

- 디렉터리 구조


composer
Application/src
vendor
PHP 라이브러리 통합관리
public
퍼블리싱 폴더(JS, CSS 등)
application
core
코어 클래스 폴더
library
사용자 정의 라이브러리
model
모델 클래스 폴더
view
뷰 파일 폴더
Controller(Processor) Files

코어 역할 (1/2)

- Config.php : 프레임워크 구동을 위한 환경설정 클래스
  - 데이터 베이스 기본정보
  - 상수 선언부

- Route.php : 요청된 주소에 따라 로더로 맵핑된 데이터 전달 클래스
  - Mathod Route() : 사용자 라우트 규칙 정의
  - Mathod Routing() : http 요청 매핑 및 라우팅
    코드이그나이터의 라우트-파싱 부분을 로직을 활용하여 사용자 정의 규칙 적용(정규표현식 매핑)
    내장 라우트 규칙 실행 로직 구현

- ORM.php : 라라벨의 엘로퀀트 ORM과 유사한 기능의 클래스
  - ORM 정적호출(static)시 해당 인스턴스를 반환하여, 내부 클래스 함수를 사용

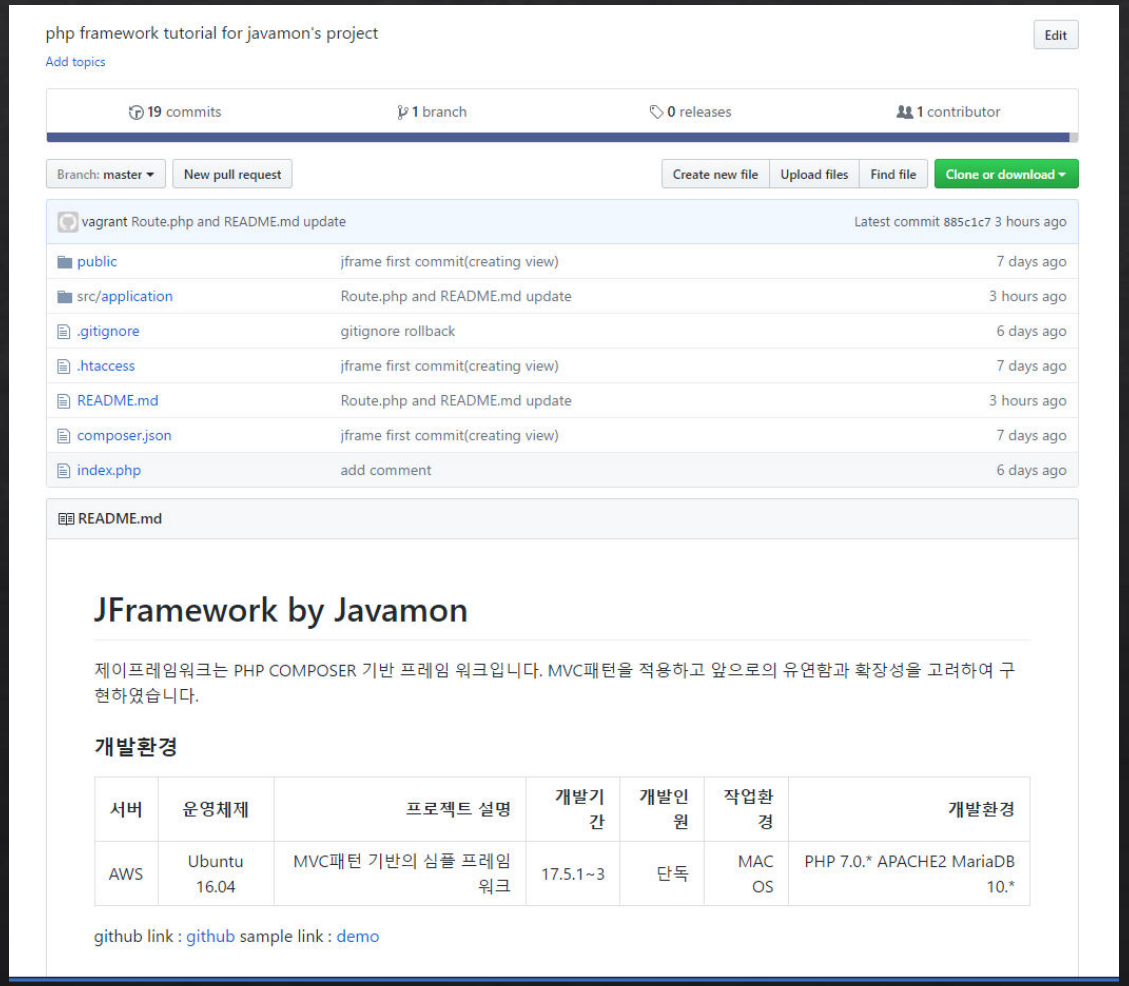
코어 역할 (2/2)

- Loader.php : 프레임워크간의 관계 정의 및 객체생성 클래스
  - 클래스를 인수로 받으며, 인스턴스를 리턴하는 클래스
- Processor.php : MVC의 컨트롤러 코어 클래스
  - 로더, 환경설정의 객체를 내부 변수에 저장
- Model.php : MVC의 모델 코어 클래스
  - 데이터 베이스 커넥터
  - 기본 동작 쿼리 함수 내장
- View.php : MVC의 뷰 코어 클래스
  - 컨트롤러(프로세서로)부터 요청 받은 페이지 로드 및 데이터 연관 배열 선언
  - 뷰 디렉토리 하위에 컨트롤러로 부터 요청된 파일이 존재하는지 확인 후 처리
  - 레이아웃의 형태를 지원하기 위해 배열의 형태로 파라미터를 받아 처리

구현

- GITHUB

1. 소스관리 URL : https://github.com/javamon1174/jframe



php framework tutorial for javamon's project

Edit

Add topics

19 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

vagrant Route.php and README.md update — Latest commit 885c1c7 3 hours ago

public	jframe first commit(creating view)	7 days ago
src/application	Route.php and README.md update	3 hours ago
.gitignore	gitignore rollback	6 days ago
.htaccess	jframe first commit(creating view)	7 days ago
README.md	Route.php and README.md update	3 hours ago
composer.json	jframe first commit(creating view)	7 days ago
index.php	add comment	6 days ago

README.md

JFramework by Javamon

제이프레임워크는 PHP COMPOSER 기반 프레임 워크입니다. MVC패턴을 적용하고 앞으로의 유연함과 확장성을 고려하여 구현하였습니다.

개발환경

서버	운영체제	프로젝트 설명	개발기간	개발인원	작업환경	개발환경
AWS	Ubuntu 16.04	MVC패턴 기반의 심플 프레임워크	17.5.1~3	단독	MAC OS	PHP 7.0.* APACHE2 MariaDB 10.*

github link : github sample link : demo

제이프레임워크 적용 프로젝트

- J_OverSearch
  - 이전에 진행했던 개인프로젝트 OverSearch에 Jframe을 적용한 샘플 프로젝트
  - Jframework의 라우트와 MVC 패턴을 통해 구현하였으며, 기존의 프로젝트 보다 퍼포먼스 향상
  - CSS, JS 일부 수정 및 JS의 AJAX를 View rendering 으로 대체
  - GITHUB : https://github.com/javamon1174/j_oversearch