

표 1.1 각종 설정 디렉터리

항목	가상환경	물리환경
문서 루트	/var/www/html/public	(사용자 홈 디렉터리)/laravel_docker/sample/public
환경 공유 디렉터리	/var/www/html/	(사용자 홈 디렉터리)/laravel_docker/sample
라라벨 프로젝트 디렉터리	/var/www/html/	(사용자 홈 디렉터리)/laravel_docker/sample



그림 1.1 Docker Desktop 다운로드 페이지

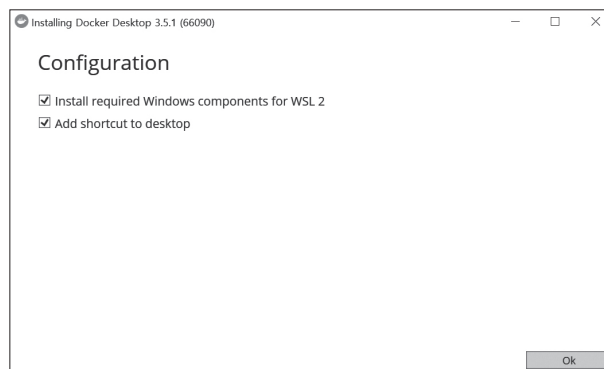


그림 1.2 윈도우용 도커 인스톨러의 Configuration 화면(Windows 10 Home)

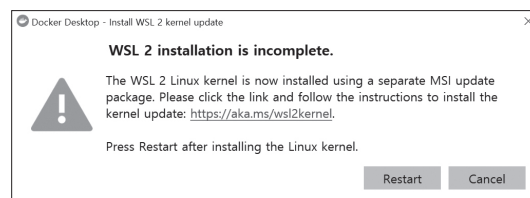


그림 1.3 WSL 2가 아직 설치되지 않았음을 알리는 경고 화면

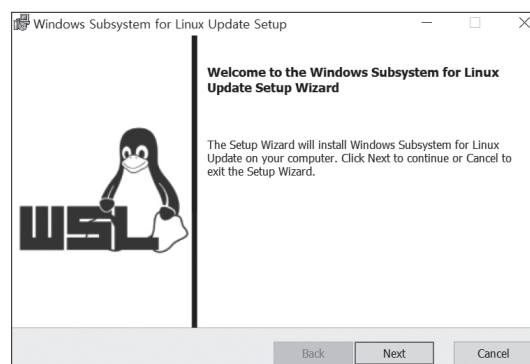


그림 1.4 WSL 3 Linux 커널 업데이트 프로그램 패키지 설치 화면

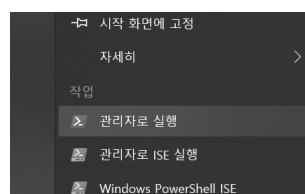


그림 1.5 Windows PowerShell을 관리자로 실행

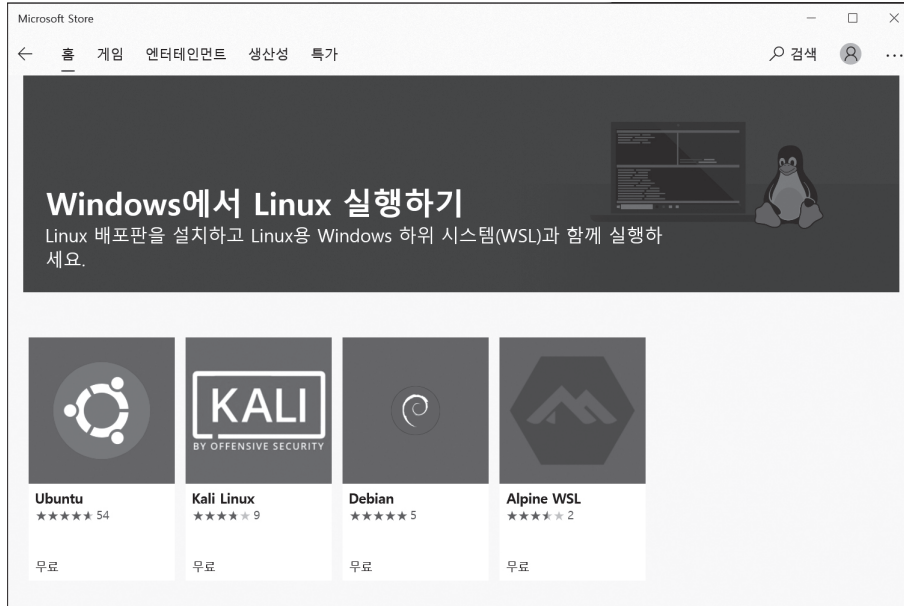


그림 1.6 마이크로소프트 스토어에서 우분투를 설치한다

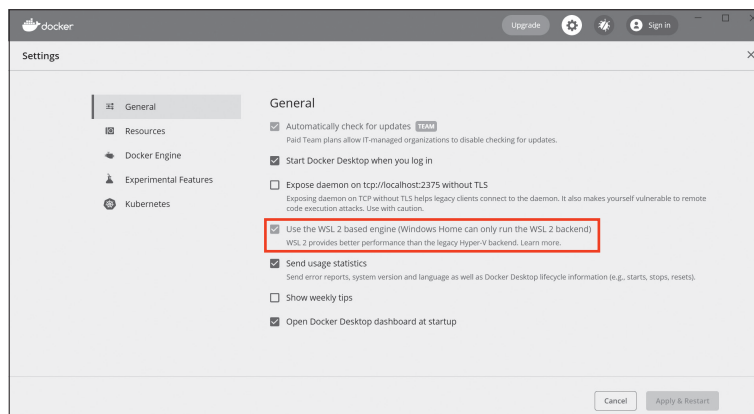


그림 1.7 Docker Desktop의 General 설정 화면

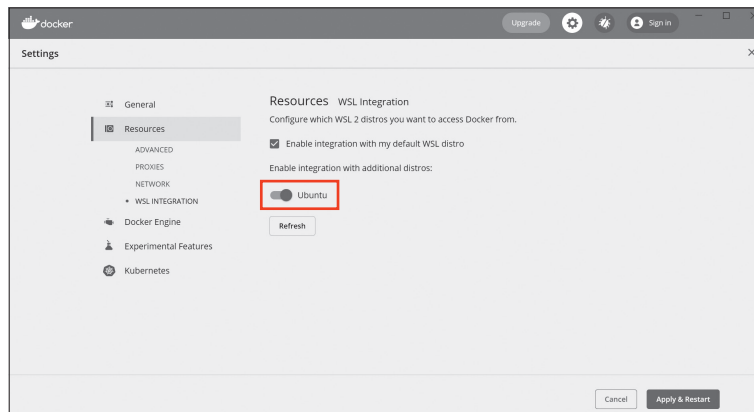


그림 1.8 Docker Desktop의 General 설정 화면

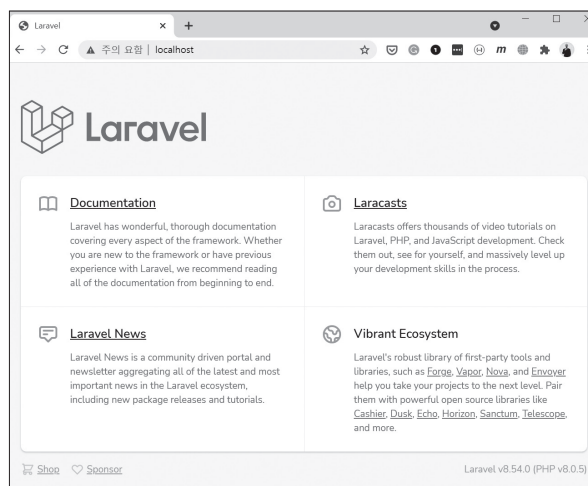


그림 1.9 라라벨의 Welcome 화면

표 1.2 각종 설정 디렉터리

위치	항목	경로
가상 환경	nginx 문서 루트	/home/vagrant/code/sample/public
	라라벨 프로젝트 디렉터리	/home/vagrant/code/sample
	물리 환경 공유 디렉터리	/home/vagrant/code
물리 환경	가상 환경 공유 디렉터리	(사용자 홈 디렉터리)/code
	라라벨 프로젝트 디렉터리	(사용자 홈 디렉터리)/code/sample
	라라독(Laradock) 다운로드 위치	(사용자 홈 디렉터리)/Homestead

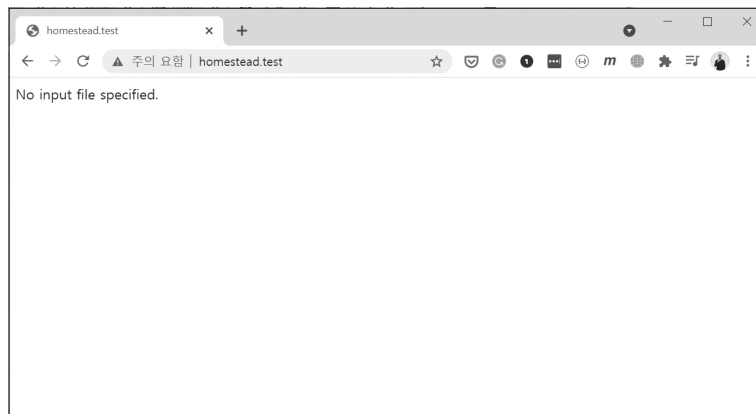


그림 1.10 Vagrant 실행 직후의 화면

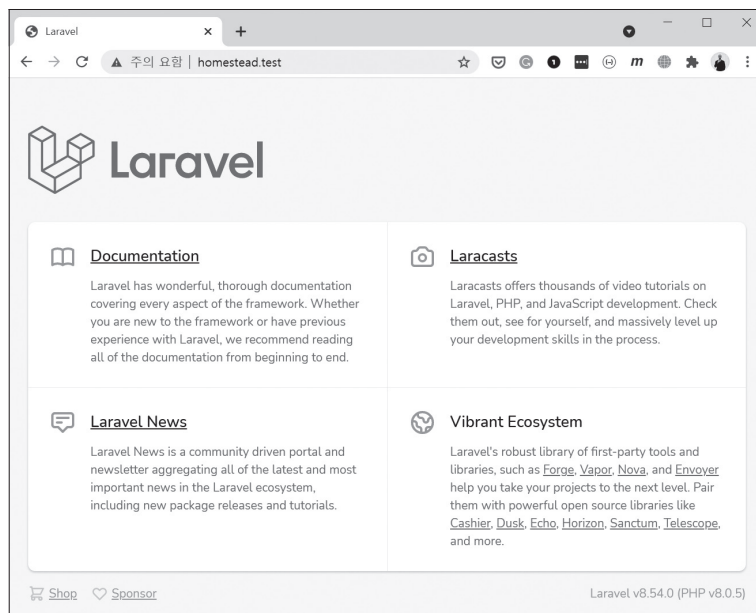


그림 1.11 라라벨의 Welcome 화면

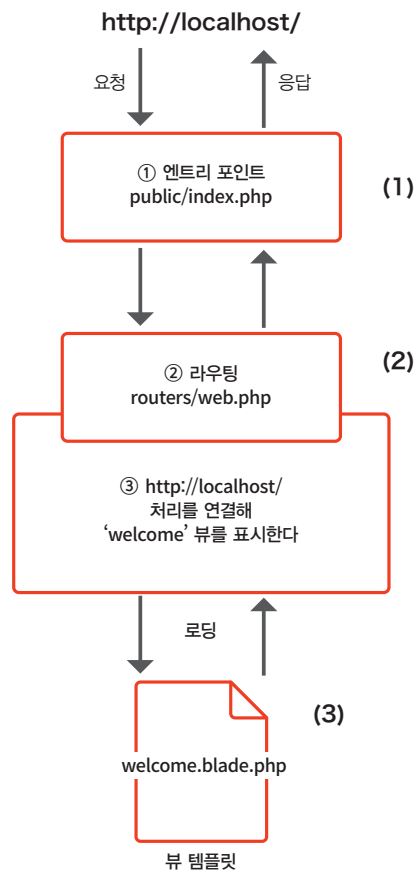


그림 1.12 Welcome 페이지 처리 흐름

표 1.3 작성할 화면/기능 리스트

ID	URI	설명
1	/home	첫 화면 표시
2	/register	사용자 등록 및 메일 송신
3	/login	로그인
4	/logout	로그아웃

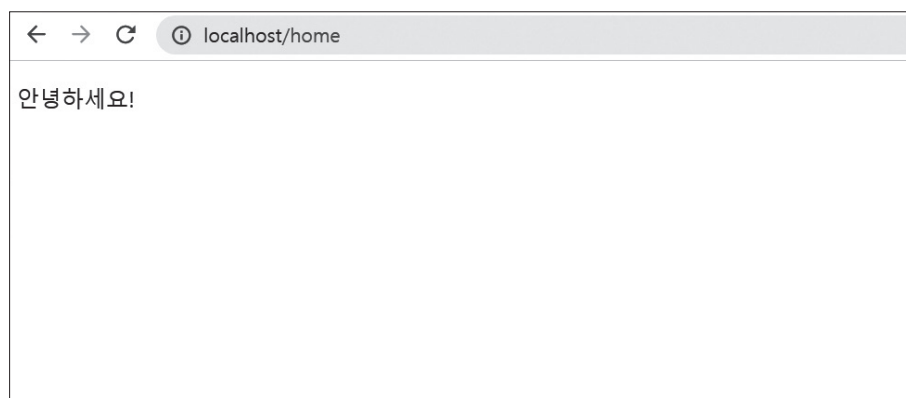


그림 1.13 첫 화면

표 1.4 users 테이블 정의

칼럼	타입	제약	비고
id	AUTO_INCREMENT	PRIMARY KEY	
name	VARCHAR(255)		사용자 이름
email	VARCHAR(255)	UNIQUE	이메일 주소
email_verified_at	TIMESTAMP	NULL 가능	이메일 확인 일시
password	VARCHAR(255)		비밀번호
remember_token	VARCHAR(100)	NULL 가능	자동 로그인용 토큰
created_at	TIMESTAMP		생성 일시
updated_at	TIMESTAMP		변경 일시

사용자 등록폼

localhost/register

이름:

메일 주소:

비밀번호:

비밀번호(확인):

그림 1.14 사용자 등록폼

표 1.5 샘플 사용자 데이터

항목	입력값
이름	Laravel
메일주소	laravel@example.com
비밀번호	password

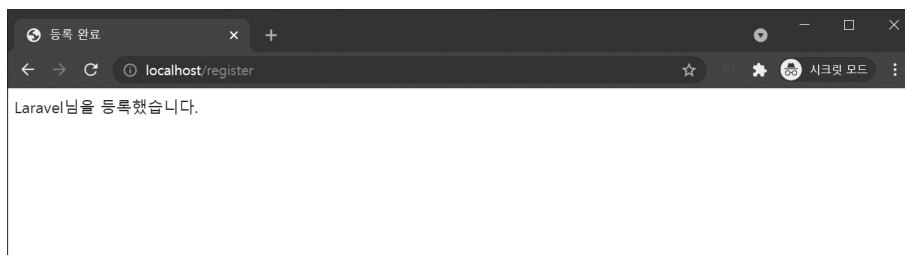


그림 1.15 보내기 버튼을 누른 뒤의 화면

```

moseskim@DESKTOP-1JGCVN: ~/laravel_docker/sample
moseskim@DESKTOP-1JGCVN: ~$ mail mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+
| id | name  | email          | email_verified_at | password                                | remember_token | created_at          | updated_at          |
+----+-----+-----+-----+-----+-----+-----+
| 1  | Laravel | laravel@example.com | NULL              | $2y$10$2T2G3j9D0ctmVfUC9nB0Fae45fsv1cN01JaE1MaBcA4v.FoWgoJgk | NULL           | 2021-07-03 21:30:11 | 2021-07-03 21:30:11 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

그림 1.16 데이터베이스를 확인한다

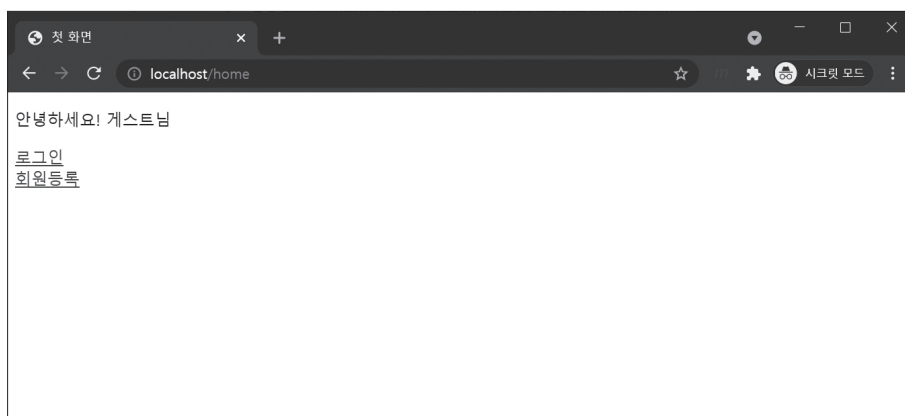


그림 1.17 로그인하지 않은 상태의 화면

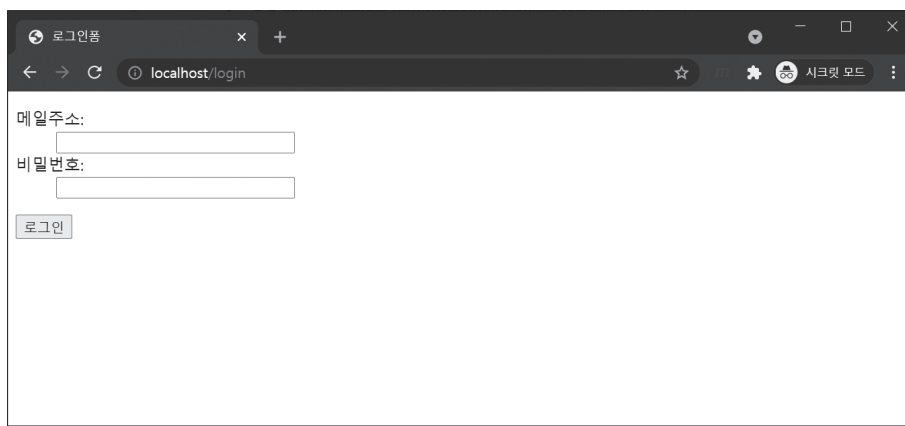


그림 1.18 로그인 화면

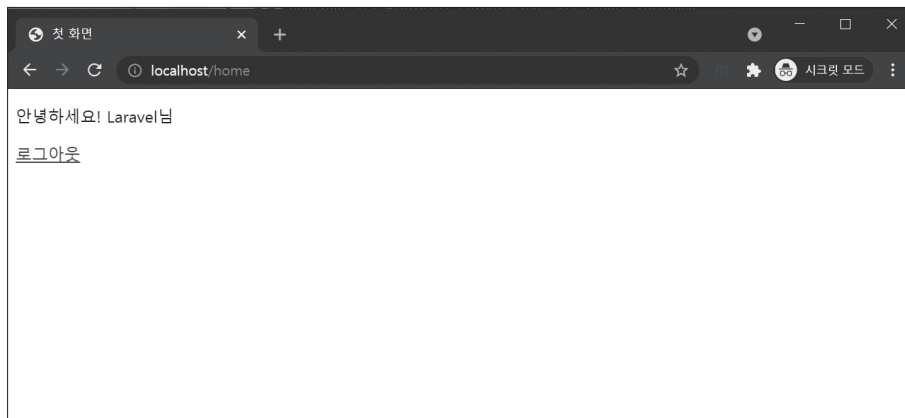


그림 1.19 로그인 성공 메시지

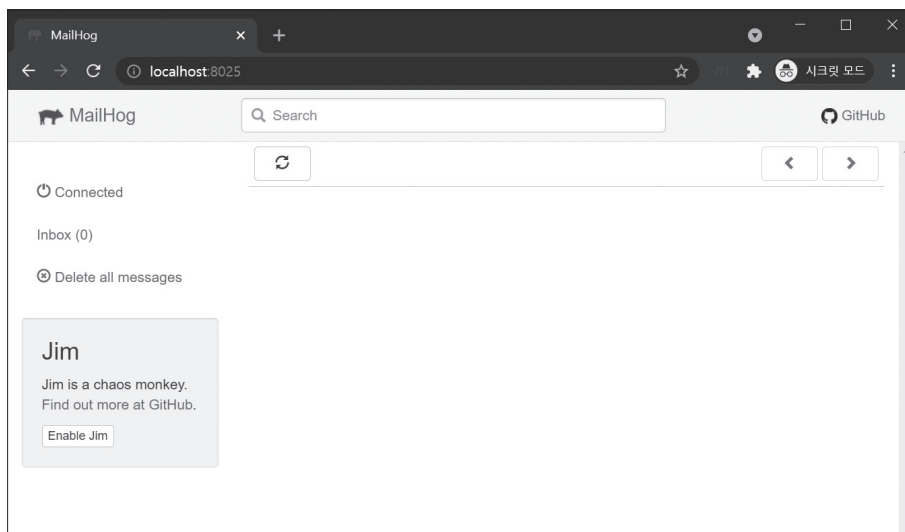


그림 1.20 MailHog 화면

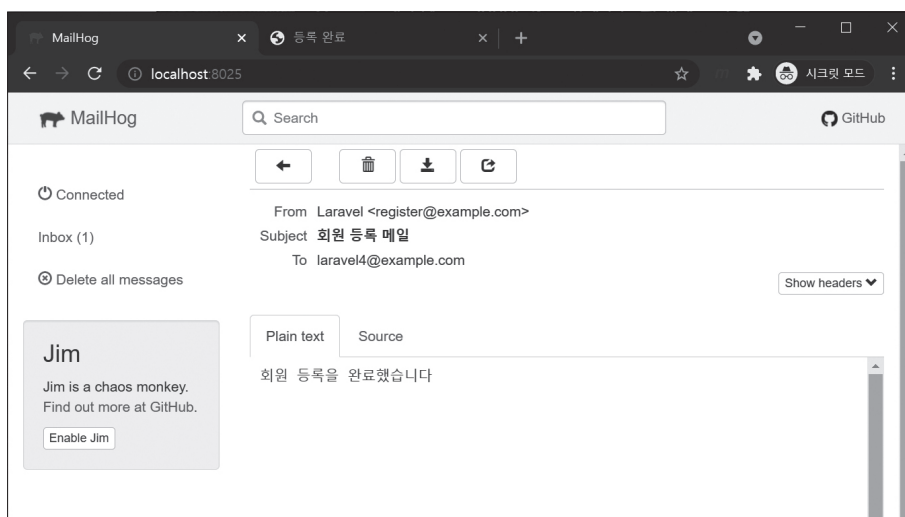


그림 1.21 MailHog 화면을 확인한다

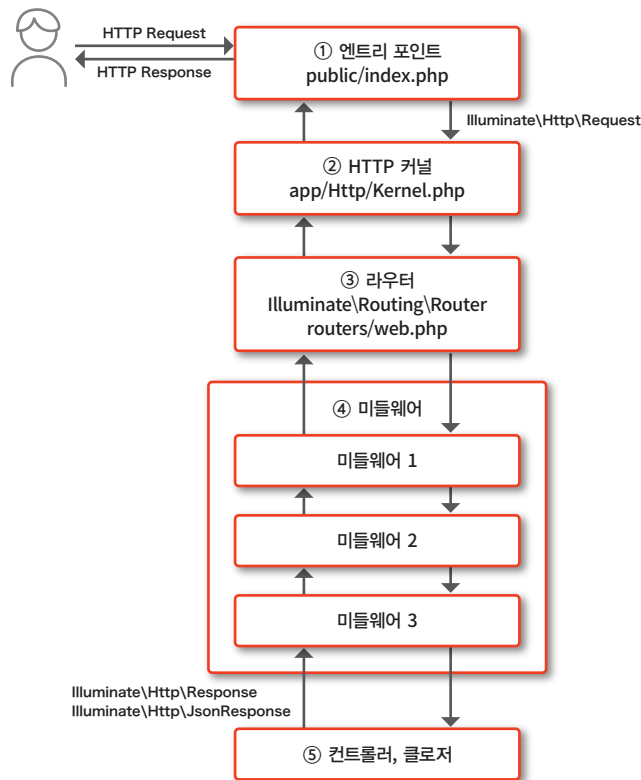


그림 2.1 라라벨 애플리케이션 실행 흐름

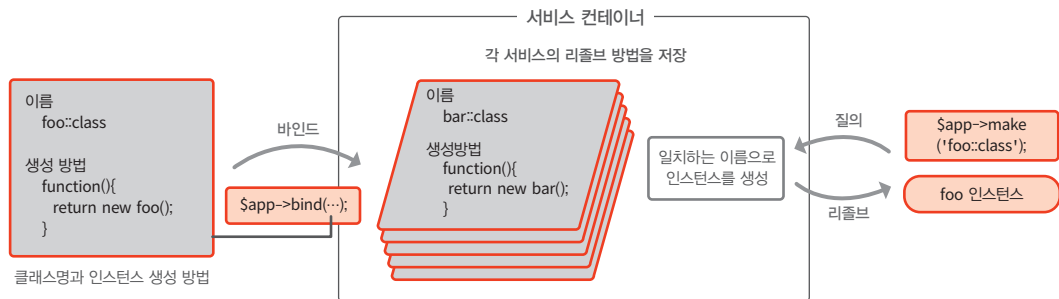


그림 2.2 서비스 컨테이너 개념도

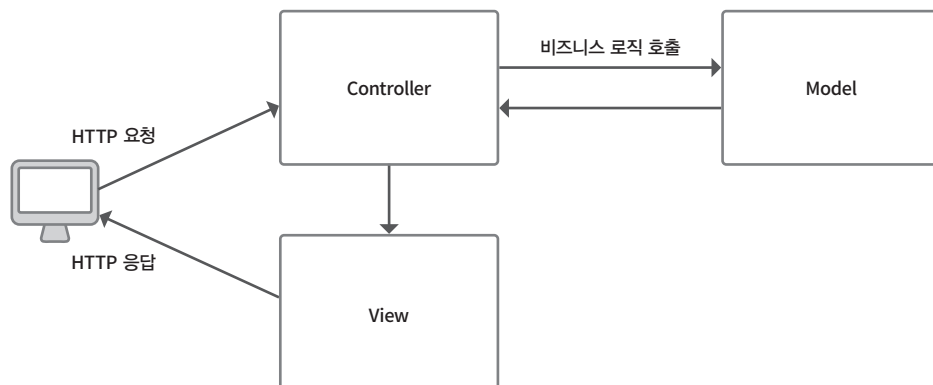


그림 3.1 MVC에 대응한 애플리케이션 설계 패턴

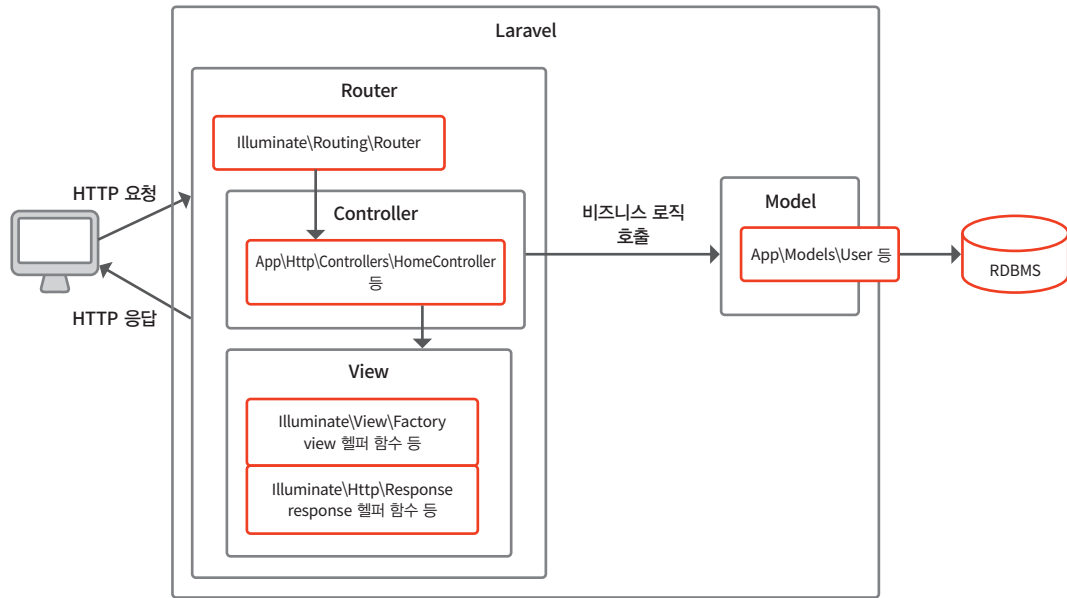


그림 3.2 MVC 패턴을 채용할 때의 흐름도

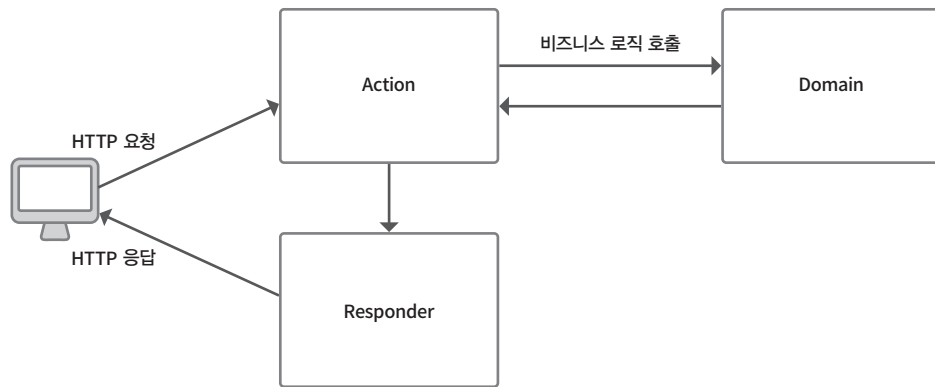


그림 3.3 ADR 패턴 이용 시 흐름도

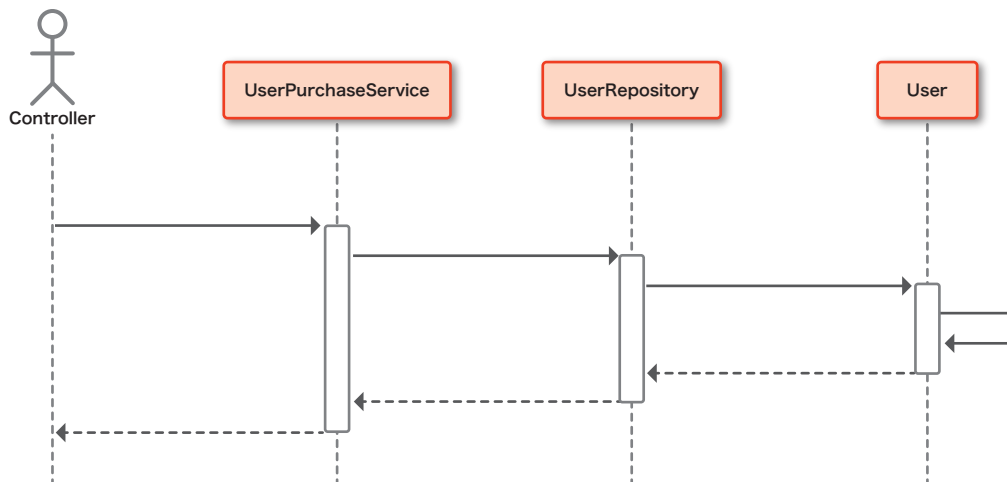


그림 3.4 계층화를 이용한 처리 흐름

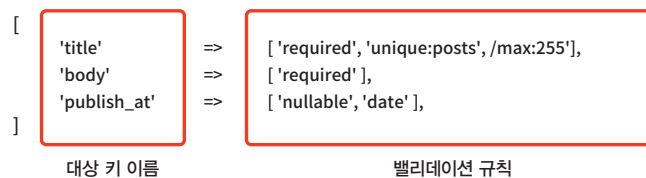


그림 4.1 밸리데이션 규칙 정의



← → ↻ localhost/register ☆

## 사용자 등록폼

- The email must be a valid email address.

이름:

메일주소:

비밀번호:

비밀번호(확인):

그림 4.2 에러 메시지 표시

← → ↻ localhost/register ☆

## 사용자 등록폼

- 메일주소 형식이 올바르지 않습니다

이름:

메일주소:

비밀번호:

비밀번호(확인):

그림 4.3 에러 메시지 표시

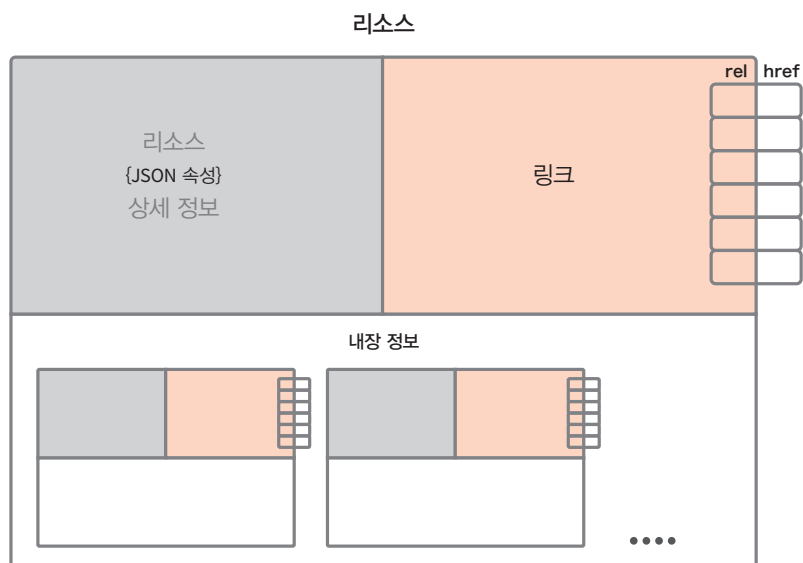


그림 4.4 리소스와 링크 구성

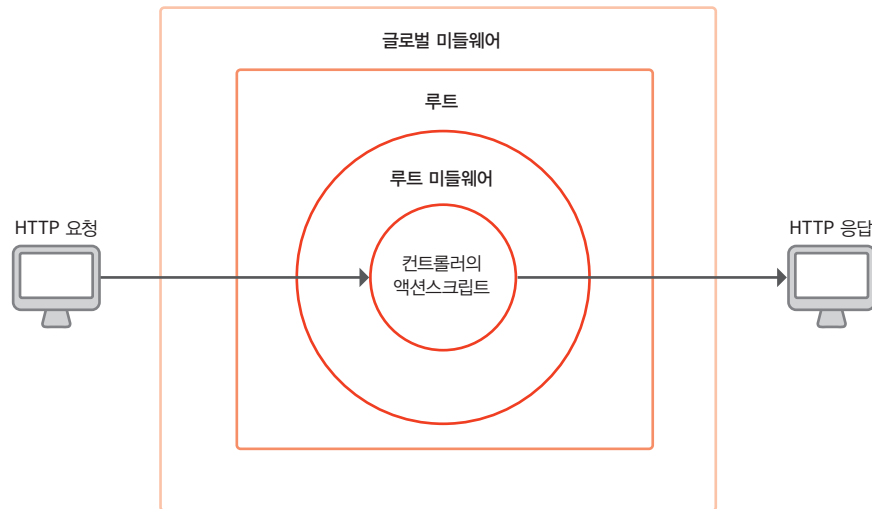


그림 4.5 미들웨어의 처리 흐름

표 4.1 글로벌 미들웨어

미들웨어 클래스	설명
App\Http\Middleware\PreventRequestsDuringMaintenance	유지보수 시 접근을 제한합니다. 모든 접근에 대해 유지보수 화면을 표시하거나 특정 URI를 유지보수에서 제외하는 등으로 지정할 수 있습니다.
Illuminate\Foundation\Http\Middleware\ValidatePostSize	요청 바디의 크기를 체크하고 이 값이 올바르지 않을 때는 Illuminate\Exceptions\PostTooLargeException을 발생시킵니다.
Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull	요청 중 빈 문자열을 Null로 변환합니다.
App\Http\Middleware\TrimStrings	요청 문자열에 대해 trim 처리를 수행해 빈 문자열을 제거합니다. except 속성에 이 처리를 제외할 요청 파라미터를 지정할 수 있습니다.
App\Http\Middleware\TrustProxies	역방향 프록시(reverse proxy) 또는 로드 밸런서(load balancer)를 사용할 때 애플리케이션 안에서 HTTPS 링크 등을 생성하지 않는 경우, 신뢰할 수 있는 ip 등을 proxies 속성 배열에 추가해서 동작시킵니다. CDN 등을 이용할 때는 이 미들웨어에 기술해야 합니다.
Fruitcake\Cors\HandleCors	라라벨에서 API를 제공할 때, 다른 도메인으로부터 자바스크립트 등을 통한 호출로 HTTP 요청을 제한하는 브라우저의 보안 설정입니다(Cross-Origin Resource Sharing, CORS). Apache HTTP Server나 Nginx 등에서 CORS를 설정한 경우에는 필요하지 않습니다.
App\Http\Middleware\TrustHosts	신뢰할 수 있는 호스트를 기술하고, 그 이외의 호스트로부터의 접근을 제한합니다. 단일 웹 서버에서 하나의 사이트를 호스팅하고 있는 경우에는 필요하지 않습니다. 하나의 애플리케이션 안에서 서브 도메인 등으로 분할하는 경우 등에 이용합니다.

표 4.2 루트 미들웨어

미들웨어 클래스	설명
App\Http\Middleware\EncryptCookies	쿠키를 암호화 및 복호화합니다. 다른 애플리케이션에서 발행된 쿠키는 암호화가 되지 않으므로 암호화 대상에 제외하고 싶은 쿠키는 except 속성 배열로 지정합니다.
App\Http\Middleware\VerifyCsrfToken	CSRF에 대응하는 XSRF-TOKEN을 발행하거나 해당 토큰을 확인합니다. HEAD 요청, GET 요청, OPTIONS 요청 이외를 대상으로 동작합니다. 대상에서 제외할 URI를 except 속성으로 지정합니다.
Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse	Cookie::queue에 등록한 값을 응답에 쿠키로 추가해서 반환합니다.
Illuminate\Session\Middleware\StartSession	세션을 활성화하고 응답 반환 시 세션에 삽입합니다.
Illuminate\Session\Middleware\AuthenticateSession	비밀번호를 변경하면 다른 장치에서 로그인한 해당 사용자를 로그아웃시킵니다.
Illuminate\View\Middleware\ShareErrorsFromSession	세션의 errors 키로부터 얻은 에러 내용을 Blade 템플릿의 errors 변수에 삽입합니다.
Illuminate\Routing\Middleware\SubstituteBindings	Eloquent 모델과 결합시켜 루트에 이용된 id 등으로부터 데이터베이스 검색을 수행해 컨트롤러 클래스나 루트 등에서 이용할 수 있도록 합니다. 이 기능은 Route Model Binding이라 부릅니다.

표 4.3 별칭이 지정된 미들웨어

미들웨어명	미들웨어 클래스	설명
auth	Illuminate\Auth\Middleware\Authenticate	인증된 사용자 여부를 판정합니다. 인증된 사용자가 아니면 Illuminate\Auth\AuthenticationException을 발생시킵니다.
auth.basic	Illuminate\Auth\Middleware\AuthenticateWithBasicAuth	기본 인증을 수행합니다.
bindings	Illuminate\Routing\Middleware\SubstituteBindings	글로벌 미들웨어에서 소개한 것과 동일한 미들웨어입니다.
cache.headers	Illuminate\Http\Middleware\SetCacheHeaders	Etag(엔티티 태그)를 이용해 콘텐츠의 캐시를 제어합니다.
can	Illuminate\Auth\Middleware\Authorize	특정 모델이나 리소스로의 액션을 인가합니다. 이 기능은 인가 기능과 조합해서 이용합니다.
throttle	Illuminate\Routing\Middleware\ThrottleRequests	동일 사용자가 단위 시간 내 규정 횟수 이상 접속 여부를 판정합니다. 규정 횟수 이상 접근했을 때는 Illuminate\Http\Exceptions\ThrottleRequestsException을 발생시킵니다.
signed	Illuminate\Routing\Middleware\ValidateSignature	서명을 포함한 접근에서 유효한 서명 여부, 제한 시간 내 접속 여부를 판정합니다.
guest	App\Http\Middleware\RedirectIfAuthenticated	인증 여부를 판단하고 인증되었을 때는 /home으로 리다이렉트합니다. 리다이렉트 위치가 고정되어 있으므로 애플리케이션에 맞춰 미들웨어를 준비합니다.
verified	Illuminate\Auth\Middleware\EnsureEmailsVerified	메일 주소로 인증된 사용자만 접근을 인가합니다. 사용하기 위해서는 Illuminate\Contracts\Auth\MustVerifyEmail 인터페이스를 구현한 클래스를 auth 드라이버 경로로 해야 합니다. 프레임워크에서 기본으로 제공하는 App\Models\User 클래스에 구현되어 있지만, 이 클래스를 이용하지 않을 때는 직접 구현해야 합니다. 자세한 내용은 '6장 인증과 인가'를 참고하세요.

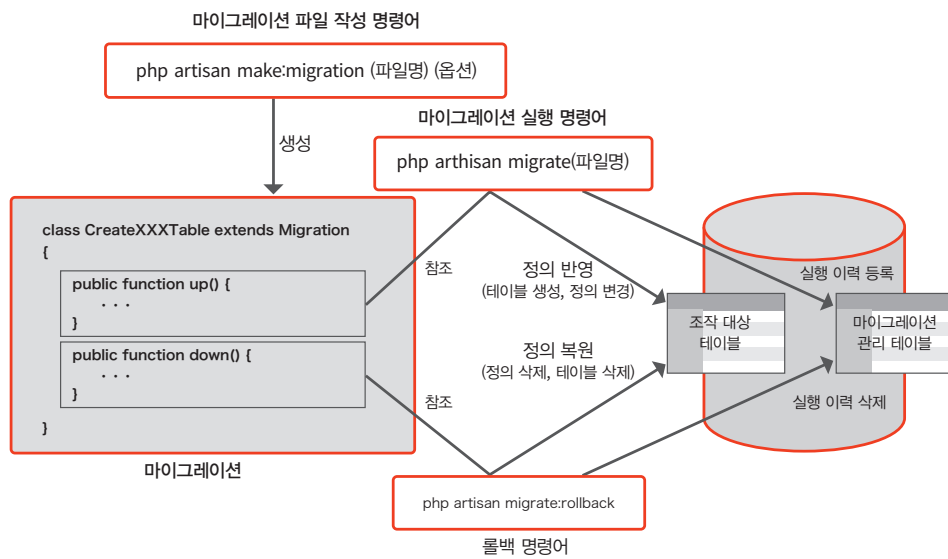


그림 5.1 마이그레이션 흐름

표 5.1 migration 명령어 옵션

옵션	기능
--create=(테이블명)	신규 테이블을 작성하기 위한 코드를 부여합니다.
--table=(테이블명)	지정한 테이블을 조작하는 코드를 부여합니다(테이블 설정 변경 등에 사용).
--path=(경로)	지정한 경로에 마이그레이션 파일을 배치합니다(애플리케이션 기본 경로의 상대 경로로 지정).

표 5.2 테이블 정의(authors 테이블)

칼럼	타입	비고
id	AUTO_INCREMENT	-
name	varchar(100)	저자명
created_at	timestamp	생성 일시
updated_at	timestamp	변경 일시

표 5.3 테이블 정의(books 테이블)

칼럼	타입	비고
id	AUTO_INCREMENT	-
name	varchar(100)	서적명
book_id	integer	서적 ID
bookdetail_id	integer	서적 상세 정보 ID
author_id	integer	저자 ID
publisher_id	integer	출판사 ID
created_at	timestamp	생성 일시
updated_at	timestamp	변경 일시

표 5.4 테이블 정의(bookdetails 테이블)

칼럼	타입	비고
id	AUTO_INCREMENT	-
isbn	varchar(100)	ISBN 코드
published_date	date	출판일
price	integer	가격
created_at	timestamp	생성 일시
updated_at	timestamp	변경 일시

표 5.5 테이블 정의(publishers 테이블)

칼럼	타입	비고
id	AUTO_INCREMENT	-
name	varchar(100)	출판사명
address	text	주소
created_at	timestamp	생성 일시
updated_at	timestamp	변경 일시

표 5.6 생성할 수 있는 칼럼 타입과 스키마 빌더 구문

칼럼 타입	스키마 빌더	비고
BOOLEAN 타입	<code>\$table-&gt;boolean(칼럼명);</code>	-
CHAR 타입	<code>\$table-&gt;char(칼럼명, 크기);</code>	두 번째 인수에 문자열 길이를 지정
DATE 타입	<code>\$table-&gt;date(칼럼명);</code>	-
DATETIME 타입	<code>\$table-&gt;dateTime(칼럼명);</code>	-
DOUBLE 타입	<code>\$table-&gt;double(칼럼명, 최대 자릿수, 유효 소수점 자릿수);</code>	두 번째 인수에 유효 전체 자릿수, 세 번째 인수에 유효 소수점 자릿수를 지정
FLOAT 타입	<code>\$table-&gt;float(칼럼명, 최대 자릿수, 유효 소수점 자릿수);</code>	두 번째 인수에 유효 전체 자릿수, 세 번째 인수에 유효 소수점 자릿수를 지정
ID 타입(primary key)	<code>\$table-&gt;id();</code>	자동으로 증가하는 BIGINT 타입 칼럼 생성
INTEGER 타입	<code>\$table-&gt;integer(칼럼명);</code>	-
JSON 타입	<code>\$table-&gt;json(칼럼명);</code>	-
VARCHAR 타입	<code>\$table-&gt;string(칼럼명, 크기);</code>	두 번째 인수에 문자열 길이를 지정
TEXT 타입	<code>\$table-&gt;text(칼럼명);</code>	-
TIMESTAMP 타입	<code>\$table-&gt;timestamp(칼럼명);</code>	-
TIMESTAMP 칼럼(소프트 딜리트용)	<code>\$table-&gt;softDeletes();</code>	소프트 딜리트용 'deleted_at' 칼럼 추가
TIMESTAMP 칼럼(등록/업데이트 일시용)	<code>\$table-&gt;timestamps();</code>	데이터 등록 일시와 업데이트 일시용 'created_at'과 'update_at' 추가

표 5.7 칼럼 속성 부여 메서드

메서드	내용
<code>-&gt;after(칼럼명)</code>	인수로 지정한 칼럼값 뒤에 배치한다(MySQL만 지원).
<code>-&gt;nullable()</code>	칼럼에 NULL 값 삽입을 허가한다.
<code>-&gt;default(기본값)</code>	칼럼의 기본값을 지정한다.
<code>-&gt;unsigned()</code>	숫자 타입의 칼럼을 UNSIGNED(부호 없음)로 설정한다.

표 5.8 인덱스 부여와 삭제를 수행하는 메서드

메서드	내용
<code>-&gt;primary(칼럼명)</code>	기본 키를 부여한다.
<code>-&gt;primary([칼럼명 1, 칼럼명 2])</code>	여러 기본 키를 부여한다.
<code>-&gt;unique(칼럼명)</code>	고유 키를 부여한다.
<code>-&gt;index(칼럼명)</code>	일반적인 인덱스를 부여한다.
<code>-&gt;dropPrimary(기본 키명)</code>	기본 키를 삭제한다.
<code>-&gt;dropUnique(고유 키명)</code>	고유 키를 삭제한다.
<code>-&gt;dropIndex(인덱스명)</code>	일반적인 인덱스를 삭제한다.

표 5.9 Faker로 만들 수 있는 더미 데이터

항목명	출력 데이터
name	이름
email	메일주소
safeEmail	메일주소
password	비밀번호
country	국가명
address	주소
phoneNumber	전화번호
company	회사명
realText	텍스트

표 5.10 그 외 Eloquent 속성

속성	설명	기본값
\$connection	데이터베이스 접속	설정 파일 database.php에 정의된 기본값
\$dateFormat	타임스탬프 포맷	Y-m-d H:i:s
\$incrementing	기본 키 자동 증가 여부	true

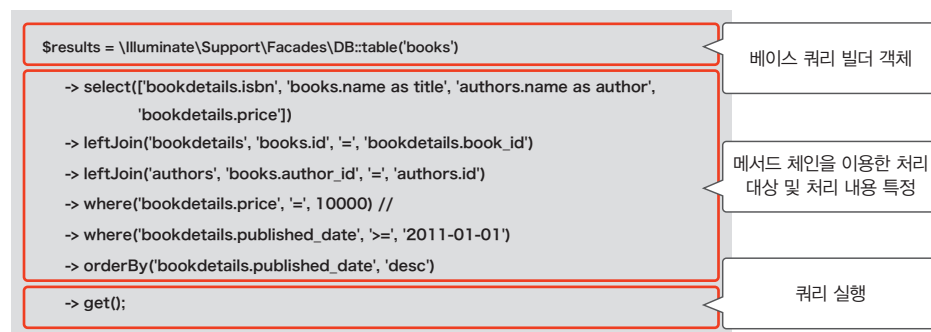


그림 5.2 쿼리 빌더 형식

표 5.11 select 계열 메서드

메서드	기능
select(칼럼명 배열)	취득 대상 칼럼명을 지정한다.
selectRaw(SQL문)	select문의 내용을 SQL로 직접 지정한다.

표 5.12 where 계열 메서드

메서드명	설명
where('칼럼명', '비교 연산자', '조건값')	where를 사용한 일반적인 조건 지정. 비교 연산자를 생략하면 등가 판정을 한다.
whereBetween('칼럼명', '범위')	between을 사용한 범위 지정
whereNotBetween('칼럼명', '범위')	not between을 사용한 범위 지정
whereIn('칼럼명', '조건값')	in을 사용한 조건 지정
whereNotIn('칼럼명', '조건값')	not in을 사용한 조건 지정
whereNull('칼럼명')	is null을 사용한 조건 지정
whereNotNull('칼럼명')	is not null을 사용한 조건 지정

표 5.13 limit와 offset 메서드

메서드명	설명
limit(숫자 값)	limit 값을 지정한다. limit구로 치환된다.
take(숫자 값)	limit 메서드와 같다.
offset(숫자 값)	offset 값을 지정한다. offset구로 치환된다.
skip(숫자 값)	offset 메서드와 같다.

표 5.14 집계 계열 메서드

메서드명	설명
orderBy(칼럼명, 방향)	정렬 대상 칼럼과 정렬 방향을 지정한다. orderBy구로 치환된다.
groupBy(칼럼명)	칼럼으로 그룹핑을 한다. groupby구로 치환된다.
having('칼럼명', '비교 연산자', '조건값')	having을 이용해 필터링한다.
havingRaw(SQL문)	having구 내용을 SQL로 직접 지정한다.

표 5.15 테이블 결합용 메서드

메서드명	설명
join('대상 테이블', '결합 대상 칼럼', '연산자', '결합 대상 칼럼')	테이블을 내부 결합한다. innerjoin구로 치환된다.
leftJoin('대상 테이블', '결합 대상 칼럼', '연산자', '결합 대상 칼럼')	테이블을 외부 왼쪽에서 결합한다. leftjoin구로 치환된다.
rightJoin('대상 테이블', '결합 대상 칼럼', '연산자', '결합 대상 칼럼')	테이블을 외부 오른쪽에서 결합한다. rightjoin구로 치환된다.

표 5.16 쿼리 실행 결과를 얻는 메서드

메서드명	설명
get()	모든 데이터를 얻는다.
first()	처음 한 행을 얻는다.

표 5.17 레코드 수 취득 및 계산 수행 메서드

메서드명	설명
count()	데이터 건수를 얻는다.
max(칼럼명)	칼럼의 최대값을 얻는다.
min(칼럼명)	칼럼의 최소값을 얻는다.
avg(칼럼명)	칼럼의 평균값을 얻는다.

표 5.18 데이터 업데이트 메서드

메서드명	설명
insert(['칼럼'=>'값', ...])	insert를 이용한 데이터 등록
update(['칼럼'=>'값', ...])	update를 이용한 데이터 업데이트
delete()	delete를 이용한 데이터 삭제
truncate()	truncate를 이용한 모든 행 삭제

표 5.19 트랜잭션 계열 메서드

메서드명	설명
DB::beginTransaction()	수동으로 트랜잭션을 시작한다.
DB::rollback()	수동으로 롤백을 실행한다.
DB::commit()	수동으로 트랜잭션을 커밋한다.
DB::transaction(클로저)	클로저 안에서 트랜잭션을 실시한다.

표 5.20 테이블 락 메서드

메서드명	설명
sharedLock()	select 된 행에 공유 락을 걸어 트랜잭션 커밋까지 업데이트를 금지한다.
lockForUpdate()	select 된 행에 배타적 락을 걸어 트랜잭션 커밋까지 읽기 및 업데이트를 금지한다.

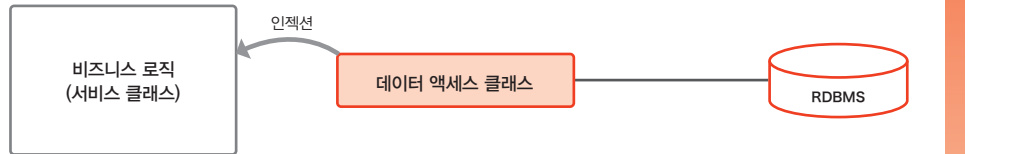
표 5.21 기본적인 SQL 실행 메서드

메서드명	설명
DB::select('select쿼리', [쿼리에 결합할 인수])	select문을 이용한 데이터 추출
DB::insert('insert쿼리', [쿼리에 결합할 인수])	insert문을 이용한 데이터 등록
DB::update('update쿼리', [쿼리에 결합할 인수])	update문을 이용한 데이터 업데이트, 변경된 행 수를 반환한다.
DB::delete('delete쿼리', [쿼리에 결합할 인수])	delete문을 이용한 데이터 삭제, 삭제된 행 수를 반환한다.
DB::statement('SQL쿼리', [쿼리에 결합할 인수])	그 외의 SQL을 실행할 때 이용한다.

① 서비스 클래스와 데이터 처리, 데이터 저장소가 의존하는 상태



② 인젝션을 사용해 비즈니스 로직으로부터 데이터 처리를 분리한 상태



③ 데이터 액세스 클래스를 추상/구상으로 나눠 클래스 사이를 느슨하게 결합한 상태

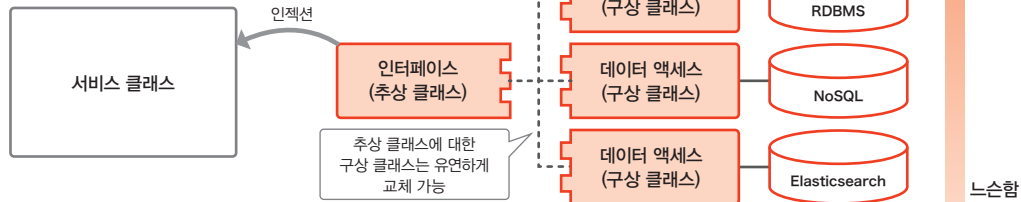


그림 5.3 리포지터리 패턴 사고방식

```

moseskim@DESKTOP-1JGC0VN: ~/laravel_docker/sample2
mysql> select * from publishers;
+----+-----+-----+-----+-----+
| id | name | address | created_at | updated_at |
+----+-----+-----+-----+-----+
| 53 | J PUB | 15, Hoedong-gil, Paju-si, Gyeonggi-do, 10881 | 2021-08-19 23:37:21 | 2021-08-19 23:37:21 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>

```

그림 5.4 publishers 테이블에 정보가 등록되었는지 확인한다

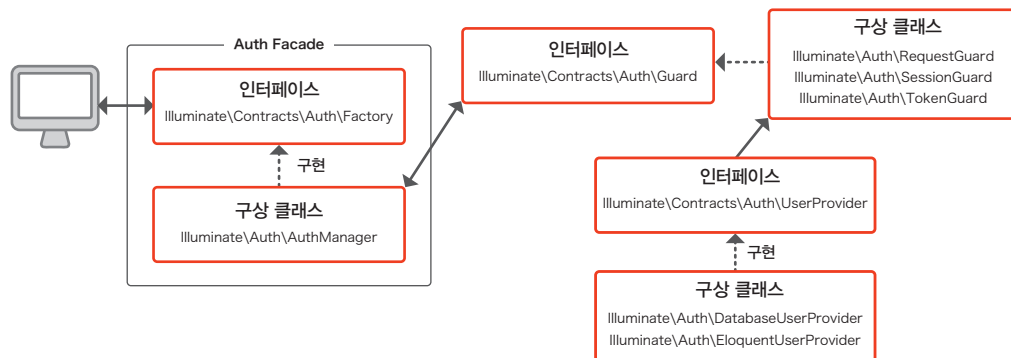


그림 6.1 인증 처리 구조



표 6.1 UserProvider 인터페이스 메서드

메서드	설명
retrieveById	로그인 후 세션에 저장된 사용자 ID 등의 고유값을 이용해 사용자 정보를 얻는다.
retrieveByToken	쿠키로부터 얻은 토큰을 이용해 사용자 정보를 얻는다(쿠키를 이용한 자동 로그인 이용 시 등).
updateRememberToken	retrieveByToken 메서드에서 이용한 토큰을 업데이트한다.
retrieveByCredentials	자격 정보를 이용해 로그인을 수행하는 attempt 메서드가 내부에서 호출하는 메서드로 지정된 배열 정보를 이용해 사용자 정보를 얻는다.
validateCredentials	attempt 메서드에 지정된 인증 정보 검증을 수행한다(비밀번호 해시값 비교 등).

표 6.2 Authenticatable 인터페이스 메서드

메서드	설명
getAuthIdentifierName	사용자를 특정하기 위해 이용하는 식별자 이름을 반환한다. 일반적으로는 데이터베이스의 컬럼명 등이 해당한다.
getAuthIdentifier	사용자를 특정할 수 있는 값을 반환한다. getAuthIdentifierName 메서드에 지정한 값을 사용해 배열 등에서 추출한다.
getAuthPassword	사용자의 비밀번호를 반환한다.
getRememberToken	자동 로그인 등에 이용하는 토큰값을 반환한다.
setRememberToken	자동 로그인 등에 이용하는 토큰값을 설정한다.
getRememberTokenName	자동 로그인 등에 이용하는 토큰명을 반환한다.

표 6.3 표준과 다른 테이블 정의

컬럼	타입	제약	비고
user_id	AUTO_INCREMENT	PRIMARY KEY	-
user_name	VARCHAR(255)	사용자명	-
email	VARCHAR(255)	UNIQUE	메일주소
password	VARCHAR(60)	비밀번호	-
created_at	TIMESTAMP	생성 일시	-
updated_at	TIMESTAMP	업데이트 일시	-

표 6.4 표준으로 제공되는 users 테이블 정의

컬럼	타입	제약	비고
id	AUTO_INCREMENT	PRIMARY KEY	-
name	VARCHAR(255)	사용자명	-
email	VARCHAR(255)	UNIQUE	메일주소
password	VARCHAR(60)	비밀번호	-
remember_token	VARCHAR(100)	NULL 허용	자동 로그인용 토큰
created_at	TIMESTAMP	생성 일시	-
updated_at	TIMESTAMP	업데이트 일시	-

표 6.5 laravel/breeze 인증 처리 라우팅

메서드	URI	이름	액션	미들웨어
GET HEAD	confirm-password	password. confirm	App\Http\Controllers\Auth\ConfirmablePasswordController@show	web,auth
POST	confirm-password		App\Http\Controllers\Auth\ConfirmablePasswordController@store	web,auth
GET HEAD	dashboard	dashborad	Closure	web,auth
POST	email/verification-notification	verification. send	App\Http\Controllers\Auth\EmailVerificationNotificationController@store	web,auth, throttle:6,1
GET HEAD	forgot-password	password. request	App\Http\Controllers\Auth\PasswordResetLinkController@create	web,auth
POST	forgot-password	password. email	App\Http\Controllers\Auth\PasswordResetLinkController@store	web,auth
GET HEAD	login	login	App\Http\Controllers\Auth\AuthenticatedSessionController@create	web,auth
POST	login		App\Http\Controllers\Auth\AuthenticatedSessionController@store	web,auth
POST	logout	logout	App\Http\Controllers\Auth\AuthenticatedSessionController@destroy	web,auth
GET HEAD	register	register	App\Http\Controllers\Auth\RegisteredUserController@create	web,auth
POST	register		App\Http\Controllers\Auth\RegisteredUserController@store	web,auth
POST	reset-password	password- update	App\Http\Controllers\Auth\NewPasswordController@store	web,auth
GET HEAD	reset-password/{token}	password. reset	App\Http\Controllers\Auth\NewPasswordController@create	web,auth
GET HEAD	verify-email	verification. notice	App\Http\Controllers\Auth\EmailVerificationPromptController@__invoke	web,auth
GET HEAD	verify-email/{id}/{hash}	verification. verify	App\Http\Controllers\Auth\VerifyEmailController@__invoke	web,auth, signed, throttle:6,1

표 6.6 user\_tokens 테이블 사양

칼럼	타입	비고
user_id	integer(10)	FK(외부 키)
api_token	varchar(60)	-
created_at	timestamp	기본적으로 CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP를 지정

표 6.7 Socialite 설정 키

설정 키	설명
client_id	외부 서비스에서 발행된 클라이언트 ID를 문자열로 기술. .env 파일과 함께 사용할 수 있다.
client_secret	외부 서비스에서 발행된 클라이언트 시크릿을 기술
redirect	외부 서비스와의 인증 처리에서 이용하는 애플리케이션의 콜백 URL을 기술
guzzle	옵션 이용 항목. Socialite가 내부에서 이용하는 Guzzle <sup>6</sup> 의 컨스트럭터에 전달하는 인수로 작용한다.

표 6.8 Laravel 구상 메서드

메서드	설명
getAuthUrl	OAuth 인증을 제공하는 서비스의 인증 제공 URL을 문자열로 입력한다.
getTokenUrl	OAuth 인증을 제공하는 서비스의 토큰 취득 URL을 문자열로 입력한다.
getUserByToken	얻은 토큰을 이용해 취득한 사용자 정보를 배열로 반환한다.
mapUserToObject	getUserByToken으로 얻은 배열을 Laravel\Socialite\Two\User 인스턴스로 변환해서 반환한다.

표 7.1 이벤트 사양

항목	내용
퍼사드	Event
헬퍼 함수(Dispatcher)	event()
Dispatcher 클래스	Illuminate\Events\Dispatcher
Dispatcher 인터페이스	Illuminate\Contracts\Events\Dispatcher

표 7.2 이벤트 클래스의 모형에 포함된 트레이트

트레이트	내용
Illuminate\Queue\SerializesModels	Queue와 조합해 비동기 이벤트를 실행할 때 이용
Illuminate\Foundation\Events\Dispatchable	이벤트 클래스에 Dispatcher로 작용시킬 때 이용
Illuminate\Broadcasting\InteractsWithSockets	socket.io를 이용해 브라우저에 이벤트를 알릴 때 이용

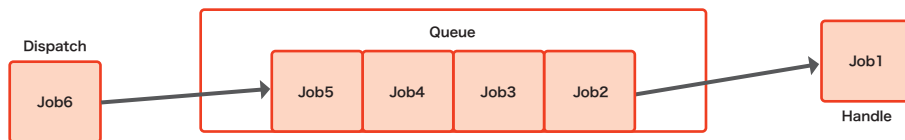


그림 7.1 잡 처리 흐름

표 7.3 큐의 사양

항목	내용
헬퍼 함수(비동기, 지연 실행)	dispatch()
헬퍼 함수(동기, 즉시 실행)	dispatch_sync()
Dispatcher 클래스	Illuminate\Bus\Dispatcher
Dispatcher 인터페이스	Illuminate\Contracts\Bus\Dispatcher

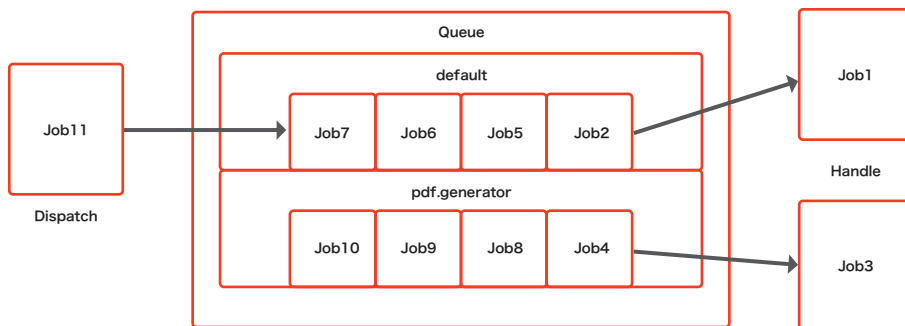


그림 7.2 큐 분산 개념

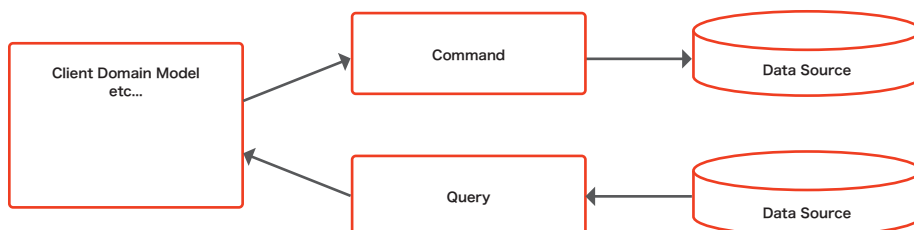


그림 7.3 CQRS 사고방식

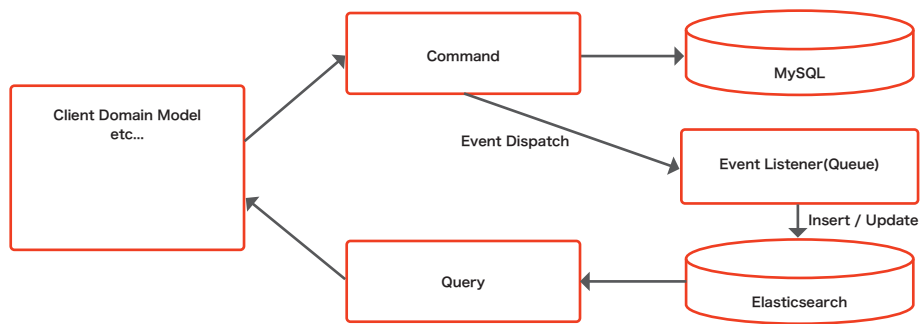


그림 7.4 이벤트와 CQRS를 조합한다

표 8.1 명령어 인수 지정 방법

명령어 인수	내용
{name}	인수를 문자열로 얻는다. 생략하면 에러가 발생한다.
{name?}	인수를 문자열로 얻는다. 생략할 수 있다.
{name=default}	인수를 문자열로 얻는다. 생략하면 = 우변을 기본값으로 설정한다.
{name*}	인수를 배열로 얻는다. 생략하면 에러가 발생한다.
{name : description}	‘:’(콜론) 뒤에 설명을 입력한다. 콜론 앞뒤로 스페이스 한 칸씩을 입력해야 한다.

표 8.2 옵션 인수 지정 방법

명령어 인수	내용
{--switch}	인수를 논리값으로 얻는다. 지정하면 true, 생략하면 false가 된다
{--switch=}	인수를 문자열로 얻는다. 생략할 수 있다.
{--switch=default}	인수를 문자열로 얻는다. 생략하면 = 우변의 값이 기본값이 된다.
{--switch=*}	인수를 배열로 얻는다. 실행 시 --switch=1 --switch=2와 같이 지정하면 ['1', '2']의 배열이 된다.
{--switch : description}	‘:’(콜론) 이후에 설명을 입력할 수 있다. 콜론 앞뒤에 스페이스 한 칸을 입력해야 한다.
{--S --switch}	‘ ’(파이프) 앞에 단축키 옵션을 지정할 수 있다. \$signature 속성에서는 --(하이픈 2개)이지 만, 실제 명령어를 실행할 때는 -(하이픈 1개)로 지정한다.

표 8.3 주요 출력 메서드

메서드	출력 형식
line(\$string, \$style = null, \$verbosity = null)	스타일 없음
info(\$string, \$verbosity = null)	info 스타일(문자색: 초록)
comment(\$string, \$verbosity = null)	comment 스타일(문자색: 노랑)
question(\$string, \$verbosity = null)	question 스타일(문자색: 검정, 배경색: 사이언)
error(\$string, \$verbosity = null)	error 스타일(문자색: 하양, 배경색: 빨강)
warn(\$string, \$verbosity = null)	warn 스타일(문자색: 노랑)

표 8.4 출력 수준

출력 수준	상숫값	명령어 실행 시 출력 수준 지정
VERBOSITY_QUIET	16	항상 출력한다.
VERBOSITY_NORMAL	32	기본 출력 수준. --quite 이외에는 출력한다.
VERBOSITY_VERBOSE	64	-v, -vv, -vvv로 출력한다.
VERBOSITY_VERY_VERBOSE	128	-vv, --vvv로 출력한다.
VERBOSITY_DEBUG	2567	-vvv만 출력한다.

※ 출력 수준은 모두 Symfony\Component\Console\Output\OutputInterface에 정의됨

표 8.5 orders 테이블<sup>1</sup>

칼럼	타입	비고
id	AUTO_INCREMENT	PK
order_code	varchar(32)	구매 코드(UNIQUE)
order_date	datetime	구매 일시
customer_name	varchar(100)	구매자 이름
customer_email	varchar(255)	구매자 메일주소
destination_name	varchar(100)	배송지 이름
destination_zip	varchar(10)	배송지 우편 번호
destination_prefecture	varchar(10)	배송지 주소(시도군구)
destination_address	varchar(100)	배송지 주소
destination_tel	varchar(20)	배송지 전화번호
total_quantity	int	구매 수량
total_price	int	합계 금액
created_at	timestamp	-
updated_at	timestamp	-

표 8.6 order\_details 테이블

칼럼	타입	비고
order_code	varchar(32)	구매 코드(PK/FK)
detail_no	int	구매 상세 번호(PK)
item_name	varchar(100)	제품명
item_price	int	제품 가격
quantity	int	구매 수량
subtotal_price	int	소계

표 8.7 구매 정보 TSV

칼럼명	내용
구매_코드	orders.order_code
구매_일시	orders.order_date
구매_상세_번호	order_details.detail_no
제품명	order_details.item_name
제품_가격	order_details.item_price
구매_수량	order_details.quantity
소계_금액	order_details.subtotal_price
합계_수량	orders.total_quantity
합계_금액	orders.total_price
구매자명	orders.customer_name
구매자_메일주소	orders.customer_email
배송지_이름	orders.destination_name
배송지_우편번호	orders.destination_zip
배송지_주소(시도군구)	orders.destination_prefecture
배송지_주소	orders.destination_address
배송지_전화번호	orders.destination_tel

<sup>1</sup> PK = 기본 키, UNIQUE = 고유 키, FK = 외부 키

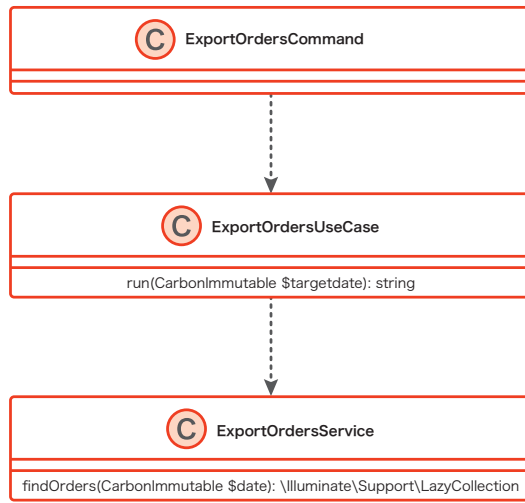


그림 8.1 클래스 다이어그램

표 8.8 배치 처리 사양

항목	내용
명령어	app:send-orders
송신 내용	어제 구매 정보
송신 시점	매일 05:00
데이터 형식	JSON
송신 대상 URL	https://api.example.com/import-orders
송신 방식	HTTP POST

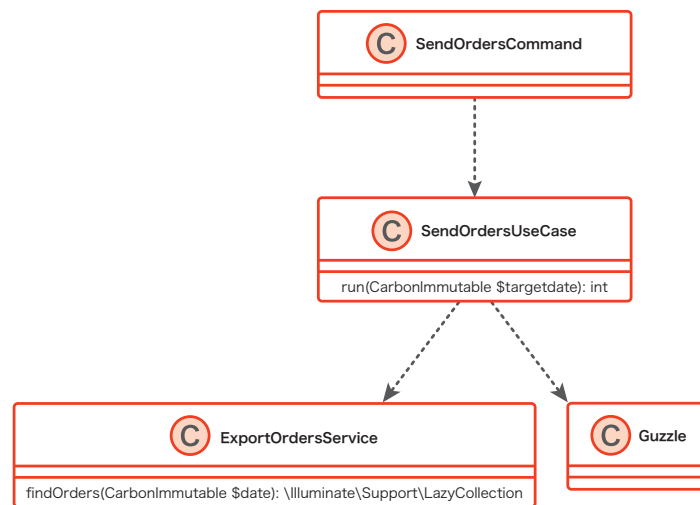


그림 8.2 클래스 구성

표 8.9 실행 빈도를 지정하는 주요 메서드

빈도	주요 메서드
분	everyMinute(), everyTwoMinutes(), everyTenMinutes()
시	hourly(), hourlyAt(1), everyTwoHours()
일	daily(), dailyAt('01:00'), twiceDaily(1, 13)
요일	weekly(), sundays(), weekdays(), weekends()
월	monthly(), monthlyOn(1, '0:0'), lastDayOfMonth('0:0')
년	yearly(), yearlyOn(1, 1, '0:0')

표 9.1 포인트 계산 규칙

구매 금액(원)	포인트
0 ~ 9,999	포인트 없음
10,000 ~ 99,999	천 원당 1포인트
100,000 이상	천 원당 2포인트

표 9.2 주요 어설션 메서드

메서드	내용
assertSame	타입을 포함해 기댓값과 값이 일치하는지 검증한다.
assertEquals	기댓값과 값이 일치하는지 검증한다.
assertTrue	값이 true인지 검증한다.
assertFalse	값이 false인지 검증한다.
assertNull	값이 null인지 검증한다.
assertInstanceOf	값의 클래스를 검증한다.
assertRegExp	값이 정규 표현과 일치하는지 검증한다.
assertCount	값이 배열일 때 요소 수가 일치하는지 검증한다.
assertArrayHasKey	값이 배열일 때 지정한 키가 존재하는지 검증한다.
assertFileEquals	파일 내용이 일치하는지 검증한다.
assertJsonFileEqualsJsonFile	JSON 파일 내용이 일치하는지 검증한다.

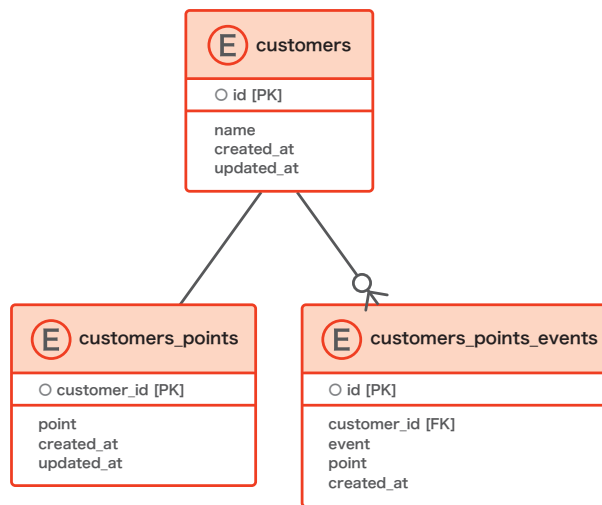


그림 9.1 테이블 구성 ER 다이어그램

표 9.3 customers 테이블

칼럼	타입	비고
id	AUTO_INCREMENT	PK(기본 키)
name	varchar(100)	이름
created_at	timestamp	등록 일시
updated_at	timestamp	업데이트 일시

표 9.4 customer\_point\_events 테이블

칼럼	타입	비고
id	AUTO_INCREMENT	PK(기본 키)
customer_id	int	FK(외부 키)
event	varchar(100)	발생 이벤트
point	int	변화 포인트
created_at	datetime	발생 일시

표 9.5 customer\_points 테이블

칼럼	타입	비고
customer_id	int	PK, FK
point	int	보유 포인트
created_at	timestamp	등록 일시
updated_at	timestamp	업데이트 일시

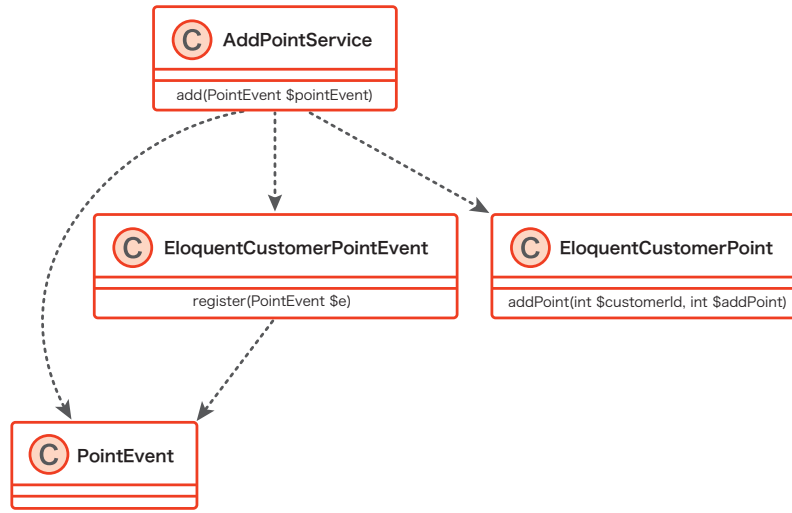


그림 9.2 포인트 추가 처리 클래스 구성

표 9.6 json 메서드에서 설정하는 요청 헤더

헤더	값
Content-Length	JSON 데이터 크기(byte)
Content-Type	application/json
Accept	application/json

표 9.7 주요 HTTP 요청 시뮬레이션 메서드

메서드	송신하는 시뮬레이션 HTTP 요청
get(\$uri, \$headers = [])	GET 요청
getJson(\$uri, \$headers = [])	GET 요청(JSON)
post(\$uri, \$data = [], \$headers = [])	POST 요청
postJson(\$uri, \$data = [], \$headers = [])	POST 요청(JSON)
put(\$uri, \$data = [], \$headers = [])	PUT 요청
putJson(\$uri, \$data = [], \$headers = [])	PUT 요청(JSON)
patch(\$uri, \$data = [], \$headers = [])	PATCH 요청
patchJson(\$uri, \$data = [], \$headers = [])	PATCH 요청(JSON)
delete(\$uri, \$data = [], \$headers = [])	DELETE 요청
deleteJson(\$uri, \$data = [], \$headers = [])	DELETE 요청(JSON)

표 9.8 HTTP 응답을 어설션하는 주요 메서드

메서드	내용
assertStatus(\$status)	HTTP 상태 코드가 \$status와 일치하면 성공
assertSuccessful()	HTTP 상태 코드가 2xx이면 성공
assertRedirect(\$uri = null)	HTTP 상태 코드가 201, 301, 302, 303, 307, 308 중 하나이고 Location 헤더값이 aap('url')->to(\$uri) 값과 일치하면 성공
assertHeader(\$headerName, \$value = null)	응답 헤더가 존재(\$value가 null일 때) 또는 해당 헤더값이 \$value와 일치하면 성공
assertHeaderMissing(\$headerName)	지정한 응답 헤더가 존재하지 않으면 성공
assertExactJson(array \$data)	응답 바디의 JSON을 디코딩한 배열이 \$data와 일치하면 성공
assertJson(array \$data, \$strict = false)	응답 바디의 JSON을 디코딩한 배열에 \$data가 포함되어 있으면 성공



표 9.9 포인트 추가 API 사양

항목	내용
HTTP 메서드	PUT
URI	/api/customers/add_point
Content-Type 헤더	application/json
요청 JSON	{"customer_id": 고객 ID, "add_point": 포인트}
응답(성공 시)	상태 코드 = 200, 바디 : {"customer_point": 추가 후 보유 포인트}
응답(밸리데이션 에러)	상태 코드 = 444, 바디 : {"message": 에러 메시지, "errors": 밸리데이션 에러}
응답(요청에 의한 에러)	상태 코드 = 400, 바디 : {"message": 에러 메시지}
응답(기타 에러)	상태 코드 = 500, 바디 : {"message": 에러 메시지}

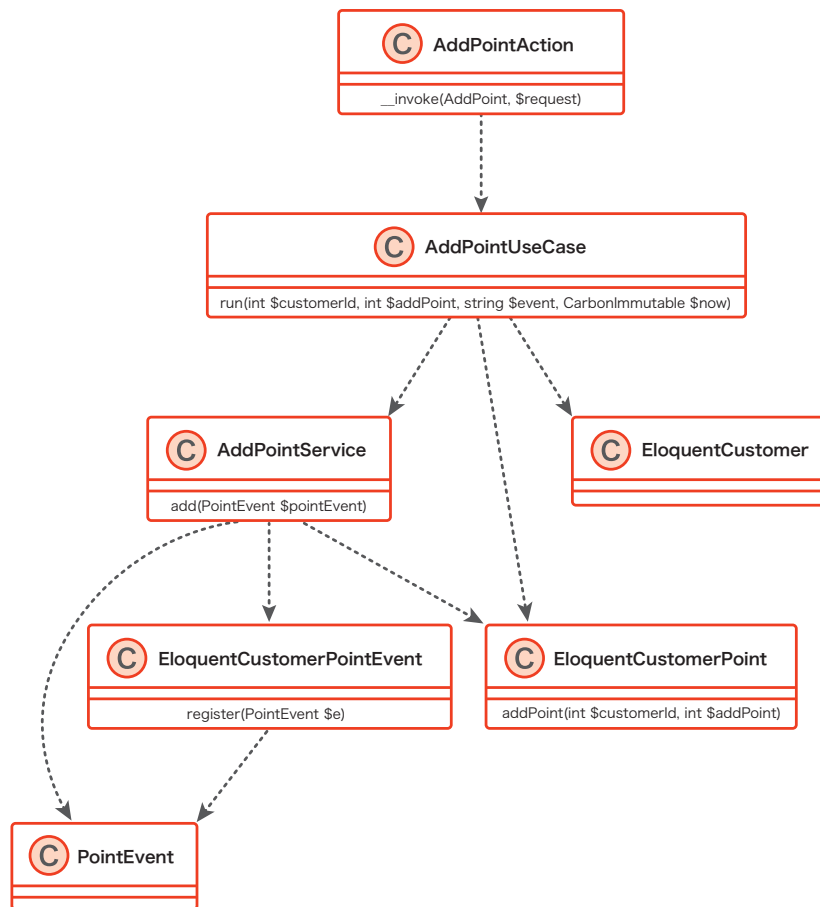


그림 9.3 포인트 추가 API 클래스 구성

표 9.10 MailFake 어설션 메서드

메서드	메서드
assertSent	지정한 메일이 송신되었는지 검증한다.
assertNotSent	지정한 메일이 송신되지 않았는지 검증한다.
assertNothingSent	메일이 송신되지 않았는지 검증한다.
assertQueued	지정한 메일이 메일 큐에 등록되었는지 검증한다.
assertNotQueued	지정한 메일이 메일 큐에 등록되지 않았는지 검증한다.
assertNothingQueued	메일 큐에 등록되지 않았는지 검증한다.

표 10.1 report 메서드에서 처리할 수 없는 예외

예외 클래스명
Illuminate\Auth\AuthenticationException
Illuminate\Auth\Access\AuthorizationException
Symfony\Component\HttpKernel\Exception\HttpException
Illuminate\Http\Exceptions\HttpResponseException
Illuminate\Database\Eloquent\ModelNotFoundException
Illuminate\Database\MultipleRecordsFoundException
Illuminate\Database\RecordsNotFoundException
Symfony\Component\HttpFoundation\Exception\SuspiciousOperationException
Illuminate\Session\TokenMismatchException
Illuminate\Validation\ValidationException

표 10.2 화면 표시 템플릿 변경 시 Blade 템플릿 파일 설치 경로 예

HTTP 상태 코드	Blade 템플릿 파일
404	resources/views/errors/404.blade.php
419	resources/views/errors/419.blade.php
429	resources/views/errors/429.blade.php
500	resources/views/errors/500.blade.php
503	resources/views/errors/503.blade.php

표 10.3 App 클래스의 동작

예외 클래스명	render 메서드 통과 시의 동작	상태 코드
Illuminate\Auth\AuthenticationException	요청 헤더에 JSON을 지정하면 상태 코드와 에러 메시지 반환, 그 외의 지정에서는 경로명이 login으로 지정된 uri로 리다이렉트한다.	401
Illuminate\Auth\Access\AuthorizationException	Symfony\Component\HttpKernel\Exception\AccessDeniedHttpException 인터페이스로 변환한다.	403
Symfony\Component\HttpKernel\Exception\HttpException	예외 발생 시 상태 코드를 지정한다.	임의의 상태 코드
Illuminate\Http\Exceptions\HttpResponseException	Symfony\Component\HttpFoundation\Response 클래스 또는 Illuminate\Http\Response 클래스에서 상태 코드를 지정한다.	임의의 상태 코드
Illuminate\Database\Eloquent\ModelNotFoundException	Symfony\Component\HttpKernel\Exception\NotFoundHttpException 인스턴스로 변환한다.	404
Illuminate\Session\TokenMismatchException	Symfony\Component\HttpKernel\Exception\HttpException 인스턴스로 변환한다.	419 반환
Illuminate\Validation\ValidationException	리퀘스트 헤더에서 JSON이 지정되어 있다면 상태 코드와 메시지를 반환하고, 그 외의 상황에서는 지정한 루트로 리다이렉트한다.	422
Symfony\Component\HttpFoundation\Exception\SuspiciousOperationException	Symfony\Component\HttpKernel\Exception\NotFoundHttpException 인스턴스로 변환한다.	404
Illuminate\Database\RecordsNotFoundException	Symfony\Component\HttpKernel\Exception\NotFoundHttpException 인스턴스로 변환한다.	404

표 10.4 메서드와 로그 수준 비교

메서드	로그 수준
Log::debug(\$message)	debug
Log::info(\$message)	info
Log::notice(\$message)	notice
Log::warning(\$message)	warning
Log::error(\$message)	error
Log::critical(\$message)	critical
Log::alert(\$message)	alert
Log::emergency(\$message)	emergency

표 10.5 Monolog 핸들러에서 이용하는 설정 키(일부 발췌)

메서드	로그 수준
handler	Monolog에서 이용하는 로그 핸들러를 문자열로 지정해 로그 드라이버를 통해 이용한다.
handler_with	handler에서 지정한 로그 핸들러 클래스의 컨스트럭터에 전달할 인수를 배열로 기술한다.
formatter	로그 핸들러 클래스에 대응한 포매터 클래스(formatter class)를 지정한다.
formatter_with	formatter에서 지정한 포매터 클래스의 컨스트럭터에 전달할 인수를 기술한다.

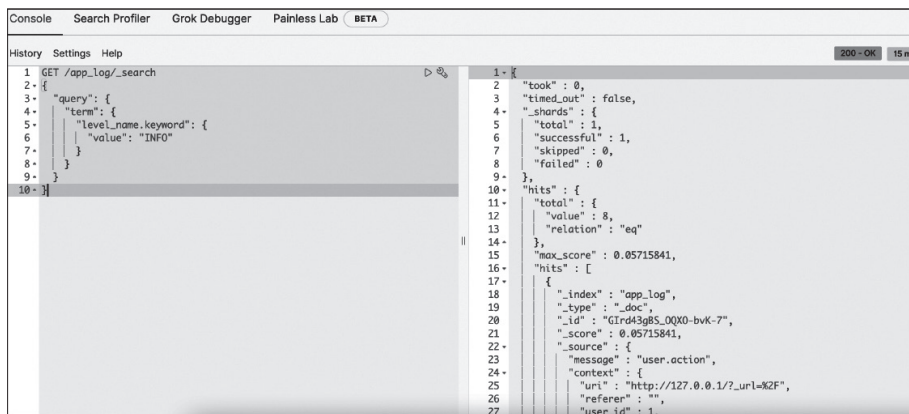


그림 10.1 Kibana Dev Tools

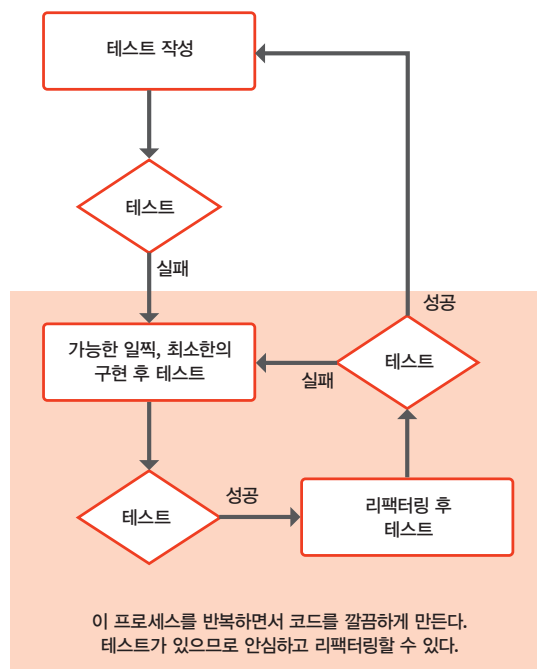


그림 11.1 TDD 프로세스

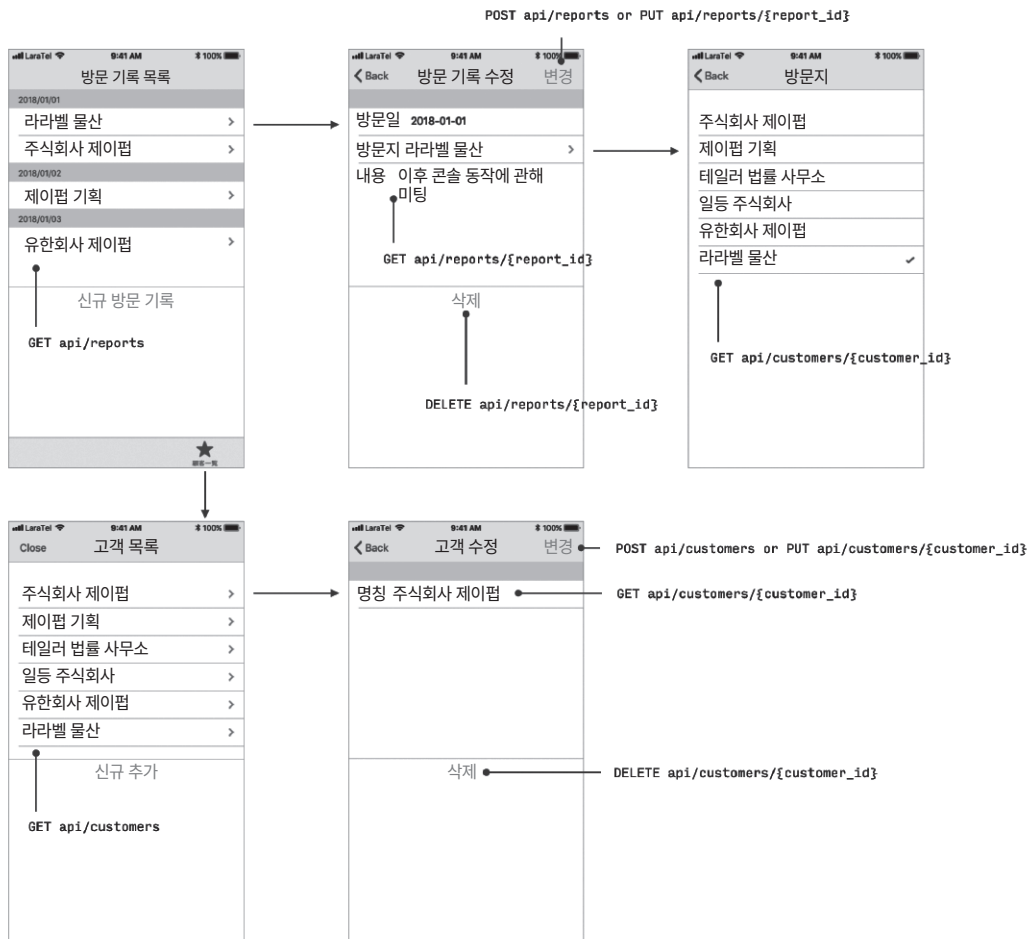


그림 11.2 샘플 모바일 애플리케이션

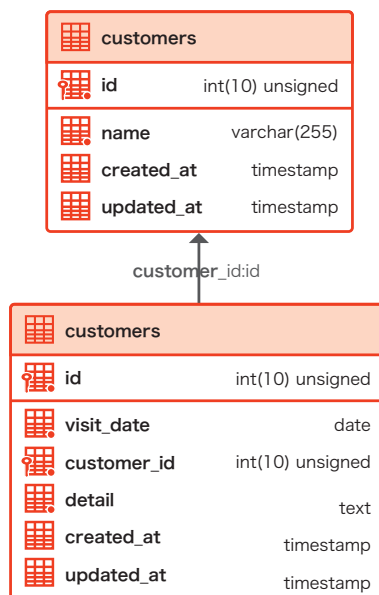


그림 11.3 데이터베이스 ER 다이어그램

표 11.1 고객 정보용 엔드포인트

내용	엔드포인트	메서드
목록	api/customers	GET
추가	api/customers	POST
참조	api/customers/{customer_id}	GET
수정	api/customers/{customer_id}	PUT
삭제	api/customers/{customer_id}	DELETE

표 11.2 방문 기록용 엔드포인트

내용	엔드포인트	메서드
목록	api/reports	GET
추가	api/reports	POST
참조	api/reports/{report_id}	GET
수정	api/reports/{report_id}	PUT
삭제	api/reports/{report_id}	DELETE

```
$report = new \App\Report();
$report->
```

- belongsTo Illuminate\Database\Eloquent\Relations...
- customer App\Customer
- customer Illuminate\Database\Eloquent\Relations\...
- hasMany Illuminate\Database\Eloquent\Relations\H...
- value(column : string) mixed
- get([columns : arr... Illuminate\Support\Collection
- toArray() array
- orderBy(column : string, [direction : stri... \$this
- raw(value) Illuminate\Database\Query\Expression
- orWhere... Illuminate\Database\Query\Builder|static

그림 11.4 IDE를 이용한 자동 보완

```
mysql> select * from customers;
```

id	name	created_at	updated_at
1	Baumbach-Schaefer	2021-08-20 22:53:28	2021-08-20 22:53:28
2	Prosacco, McKenzie and Harris	2021-08-20 22:53:28	2021-08-20 22:53:28

```
2 rows in set (0.00 sec)
```

그림 11.5 customers 테이블에 2건의 데이터가 생성된다

```
mysql> select * from reports;
```

id	visit_date	customer_id	detail	created_at	updated_at
1	1984-06-13	1	Why, I haven't had a head unless there was nothing else to say 'creatures,' you see, because some of YOUR business, 'twol' said Seven. 'Yes, it IS his business!' said Five, 'and I'll tell him—it was,	2021-08-20 22:53:28	2021-08-20 22:53:28
2	1973-10-21	1	Mouse in the last words out loud, and the other side, The further off from England the nearer is to find her way out	2021-08-20 22:53:28	2021-08-20 22:53:28
3	2012-08-28	2	I shall be punished for it to make personal remarks,' Alice said to herself. 'I to get in?' asked Alice again, in a hurry that she never knew so much already, that it was over at last: 'and I do	2021-08-20 22:53:28	2021-08-20 22:53:28
4	2004-10-04	2	wonder what they'll do next! As for pulling me out of the country is, you see, as, 'Alice angrily. 'It wasn't very civil of you to set them free, Exactly as we needn't try to find my way into that bea	2021-08-20 22:53:28	2021-08-20 22:53:28

```
4 rows in set (0.00 sec)
```

그림 11.6 reports 테이블에 4건의 데이터가 생성된다