

```
<?php
```

```
use Illuminate\Http\Request;
```

```
/*
```

```
|-----
```

```
| API Routes
```

```
|-----
```

```
|
```

```
| Here is where you can register API routes for your application. These  
| routes are loaded by the RouteServiceProvider within a group which  
| is assigned the "api" middleware group. Enjoy building your API!
```

```
|
```

```
*/
```

```
//Route::middleware('auth:api')->get('/user', function (Request $request)
```

```
{
```

```
//    return $request->user();
```

```
//});
```

```
Route::group(['middleware' => ['json.response']], function () {
```

```
    Route::middleware('auth:api')->get('/user', function (Request $request)
```

```
{
```

```
        return $request->user();
```

```
});
```

```
// public routes
```

```
Route::post('/login', 'api\authcontroller@login')->name('login.api');
```

```
Route::post('/register',
```

```
'api\authcontroller@register')->name('register.api');
```

```
// private routes
```

```
Route::middleware('auth:api')->group(function () {
```

```
    Route::get('/logout', 'Api\AuthController@logout')->name('logout');
```

```
});
```

```
// json store routes list
Route::middleware('auth:api')->group(function () {

    Route::get('/user', 'Api\UserController@index')->name('user');
    Route::get('/ticket', 'Api\TicketController@index')->name('user');
    Route::get('/organization',
'Api\OrganizationController@index')->name('user');

    Route::get('/user/{id}',
'api\usercontroller@searchbyorganizationid')->name('user.search');
    Route::get('/ticket/{id}',
'api\ticketcontroller@searchbyorganizationid')->name('ticket.search');
    Route::get('/organization/{id}',
'api\organizationcontroller@searchbyorganizationid')->name('organization.search');

    Route::get('/organizations/{id}',
'api\organizationcontroller@search')->name('organization.search.ticket');

    Route::get('/users/{name}',
'api\usercontroller@search')->name('user.search.ticket');

    Route::get('/tickets/{subject}',
'api\ticketcontroller@search')->name('ticket.search.user');
});

});
```

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\User;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Laravel\Passport\HasApiTokens;

class AuthController extends Controller
{
    //
    public function register (Request $request) {

        $validator = Validator::make($request->all(), [
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:6|confirmed',
        ]);

        if ($validator->fails())
        {
            return response(['errors'=>$validator->errors()->all()], 422);
        }

        $request['password']=Hash::make($request['password']);
        $user = User::create($request->toArray());

        $token = $user->createToken('Laravel Password Grant
Client')->accessToken;
        $response = ['token' => $token];

        return response($response, 200);

    }
    public function login (Request $request) {

        $user = User::where('email', $request->email)->first();

```

```

    if ($user) {

        if (Hash::check($request->password, $user->password)) {
            $token = $user->createToken('Laravel Password Grant
Client')->accessToken;
            $response = ['token' => $token];
            return response($response, 200);
        } else {
            $response = "Password mismatch";
            return response($response, 422);
        }

    } else {
        $response = 'User does not exist';
        return response($response, 422);
    }

}

public function logout (Request $request) {

    $token = $request->user()->token();
    $token->revoke();

    $response = 'You have been succesfully logged out!';
    return response($response, 200);

}
}

```

```

<?php

```

```

namespace App\Http\Controllers\Api;

```

```

use App\Http\Controllers\Controller;

```

```
use App\Services\OrganizationService;
use Exception;
use Illuminate\Contracts\View\Factory;
use Illuminate\Http\Request;
use Illuminate\View\View;
use Throwable;
```

```
class OrganizationController extends Controller
{
    private $service;

    public function __construct(OrganizationService $service)
    {
        $this->service = $service;
    }

    /**
     * @param Request $request
     * @return array|Factory|View|string
     * @throws Throwable
     */
    public function index(Request $request)
    {
        try {
            $value = $this->service->index();
            return $value;
        } catch (Exception $exception) {
            return response()->json([$exception]);
        }
    }

    public function searchByOrganizationId($id)
    {
        try {
            $value = $this->service->searchByOrganizationId($id);
            return $value;
        }
    }
}
```

```

        } catch (Exception $exception) {
            return response()->json([$exception]);
        }
    }

    public function search($id)
    {
        try {
            $value = $this->service->search($id);
            return $value;
        } catch (Exception $exception) {
            return response()->json([$exception]);
        }
    }
}

```

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```

use App\Http\Controllers\Controller;
use App\Services\TicketService;
use Illuminate\Contracts\View\Factory;
use Illuminate\Http\Request;
use Illuminate\View\View;
use Throwable;

```

```

class TicketController extends Controller
{
    private $service;

    public function __construct(TicketService $service)
    {
        $this->service = $service;
    }
}

```

```
}
```

```
/**
```

```
 * @param Request $request
```

```
 * @return array|Factory|View|string
```

```
 * @throws Throwable
```

```
 */
```

```
public function index(Request $request)
```

```
{
```

```
    try {
```

```
        $value = $this->service->index();
```

```
        return $value;
```

```
    } catch (Exception $exception) {
```

```
        return response()->json([$exception]);
```

```
    }
```

```
}
```

```
public function searchByOrganizationId($id)
```

```
{
```

```
    try {
```

```
        $value = $this->service->searchByOrganizationId($id);
```

```
        return $value;
```

```
    } catch (Exception $exception) {
```

```
        return response()->json([$exception]);
```

```
    }
```

```
}
```

```
public function search($subject)
```

```
{
```

```
    try {
```

```
        $value = $this->service->search($subject);
```

```
        return $value;
```

```
    } catch (Exception $exception) {
```

```
        return response()->json([$exception]);
```

```
    }  
    }  
}
```

<?php

```
namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;  
use App\Services\UserService;  
use Illuminate\Contracts\View\Factory;  
use Illuminate\Http\Request;  
use Illuminate\View\View;  
use Throwable;
```

```
class UserController extends Controller
```

```
{  
    private $service;  
  
    public function __construct(UserService $service)  
    {  
        $this->service = $service;  
    }  
}
```

```
/**
```

```
 * @param Request $request  
 * @return array|Factory|View|string  
 * @throws Throwable  
 */
```

```
public function index(Request $request)  
{  
    try {
```



```

        $value = $this->service->index();
        return $value;
    } catch (Exception $exception) {
        return response()->json([$exception]);
    }
}

public function searchByOrganizationId($id)
{
    try {
        $value = $this->service->searchByOrganizationId($id);
        return $value;
    } catch (Exception $exception) {
        return response()->json([$exception]);
    }
}

public function search($name)
{
    try {
        $value = $this->service->search($name);
        return $value;
    } catch (Exception $exception) {
        return response()->json([$exception]);
    }
}
}

```

<?php

```

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your application.
     *
     * @var array
     */
    protected $middleware = [
        \App\Http\Middleware\CheckForMaintenanceMode::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,

        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \App\Http\Middleware\TrustProxies::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array
     */
    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,

            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            // \Illuminate\Session\Middleware\AuthenticateSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \App\Http\Middleware\VerifyCsrfToken::class,

```

```

        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],

    'api' => [
        'throttle:60,1',
        'bindings',
    ],
];

/**
 * The application's route middleware.
 *
 * These middleware may be assigned to groups or used individually.
 *
 * @var array
 */
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' =>
        \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' =>
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' =>
        \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' =>
        \Illuminate\Auth\Middleware\EnsureEmailsVerified::class,
    'json.response' =>
        \App\Http\Middleware\ForceJsonResponse::class,
];

/**
 * The priority-sorted list of middleware.
 *
```

```

    * This forces non-global middleware to always be in the given order.
    *
    * @var array
    */
protected $middlewarePriority = [
    \Illuminate\Session\Middleware\StartSession::class,
    \Illuminate\View\Middleware\ShareErrorsFromSession::class,
    \App\Http\Middleware\Authenticate::class,
    \Illuminate\Session\Middleware\AuthenticateSession::class,
    \Illuminate\Routing\Middleware\SubstituteBindings::class,
    \Illuminate\Auth\Middleware\Authorize::class,
];
}

```

```
<?php
```

```
namespace App\Providers;
```

```

use Illuminate\Support\Facades\Gate;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as
ServiceProvider;
use Laravel\Passport\Passport;

```

```

class AuthServiceProvider extends ServiceProvider
{
    /**
     * The policy mappings for the application.
     *
     * @var array
     */
    protected $policies = [
        'App\Model' => 'App\Policies\ModelPolicy',
    ];

    /**

```

```

    * Register any authentication / authorization services.
    *
    * @return void
    */
    public function boot()
    {
        $this->registerPolicies();
        Passport::routes();

        //
    }
}

```

```
<?php
```

```
namespace App\Repositories\Interfaces;
```

```
interface IRepository
```

```

{
    //
}

```

```
<?php
```

```
namespace App\Repositories;
```

```
use App\Repositories\Interfaces\IRepository;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
abstract class BaseRepository implements IRepository
```

```
{
```

```
protected $model;
```

```
public function store(array $data): Model
{
    return $this->model->create($data);
}
```

```
public function update(int $id, array $data): Model
{
    $result = $this->model->find($id);
    $result->update($data);

    return $result;
}
```

```
public function delete(int $id): bool
{
    $result = $this->model->find($id);
    if (!$result) return false;
    return $result->delete();
}
```

```
public function getDataById(int $id): Model
{
    $model = $this->model->find($id);
    if (!$model) return $this->model;

    return $model;
}
}
```

```
<?php
```

```
namespace App\Repositories;
```

```
use Illuminate\Support\Facades\Storage;
```

```
class OrganizationRepository extends BaseRepository
```

```
{
```

```
    public function __construct()
```

```
    {
```

```
    }
```

```
    public function json()
```

```
    {
```

```
        $path = storage_path() . "/app/JsonStore/organizations.json";
```

```
        $json = json_decode(file_get_contents($path), true);
```

```
        return response()->json($json);
```

```
    }
```

```
    public function searchByOrganizationId($id)
```

```
    {
```

```
        $json = Storage::disk('local')->get('JsonStore/tickets.json');
```

```
        $json_user = Storage::disk('local')->get('JsonStore/users.json');
```

```
        $data = json_decode($json, true);
```

```
        $data_user = json_decode($json_user, true);
```

```
        $datas = array_filter($data);
```

```
        $data_users = array_filter($data_user);
```

```
        $data_ticket = collect($datas)->where('organization_id', $id);
```

```
        $data_user = collect($data_users)->where('organization_id', $id);
```

```
        $value = array(
```

```
            $data_ticket,
```

```
            $data_user,
```

```
        );
```

```
        return response()->json($value);
```

```

    }

    public function search($id)
    {
        $json_ticket = Storage::disk('local')->get('JsonStore/tickets.json');
        $json_user = Storage::disk('local')->get('JsonStore/users.json');
        $data = json_decode($json_ticket, true);
        $data_user = json_decode($json_user, true);

        $datas = array_filter($data);
        $data_users = array_filter($data_user);

        $data_ticket = collect($datas)->where('organization_id',
        $id)->pluck('subject');

        $data_user = collect($data_users)->where('organization_id',
        $id)->pluck('name');

        $value = array(
            $data_ticket,
            $data_user,
        );
        return response()->json($value);
    }
}

```

```

<?php

```

```

namespace App\Repositories;

```

```

use Illuminate\Support\Facades\Storage;

```

```

class TicketRepository extends BaseRepository
{

```



```
public function __construct()
{
}
```

```
public function json()
{

    $json = Storage::disk('local')->get('JsonStore/tickets.json');
    $json = json_decode($json, true);
    return response()->json($json);

}
```

```
public function searchByOrganizationId($id)
{
    $json = Storage::disk('local')->get('JsonStore/tickets.json');
    $data = json_decode($json, true);

    $datas = array_filter($data);

    $data = collect($datas)->where('organization_id', $id);

    return response()->json($data);

}
```

```
public function search($subject)
{
    $json_ticket = Storage::disk('local')->get('JsonStore/tickets.json');
    $json_user = Storage::disk('local')->get('JsonStore/users.json');
    $json_organization =
Storage::disk('local')->get('JsonStore/organizations.json');
    $data = json_decode($json_ticket, true);
    $data_user = json_decode($json_user, true);
    $data_organization = json_decode($json_organization, true);
```

```

    $datas = array_filter($data);
    $data_users = array_filter($data_user);
    $data_organizations = array_filter($data_organization);

    $data_ticket_assignee_id = collect($datas)->where('subject',
    $subject)->pluck('assignee_id');
    $data_ticket_submitter_id = collect($datas)->where('subject',
    $subject)->pluck('submitter_id');
    $data_organization_id = collect($datas)->where('subject',
    $subject)->pluck('organization_id');

    $data_user_assignee_id = collect($data_users)->where('_id',
    $data_ticket_assignee_id->first())->pluck('name');
    $data_user_submitter_id = collect($data_users)->where('_id',
    $data_ticket_submitter_id->first())->pluck('name');

    $data_user_organization =
    collect($data_organizations)->where('_id',
    $data_organization_id->first())->pluck('name');

    $value = array(

        $data_user_assignee_id->all(),
        $data_user_submitter_id->all(),
        $data_user_organization->all()
    );

    return response()->json($value);
}
}

```

<?php

namespace App\Repositories;

```
use Illuminate\Support\Facades\Storage;
```

```
class UserRepository extends BaseRepository
```

```
{
```

```
    public function __construct()
```

```
    {
```

```
    }
```

```
    public function json()
```

```
    {
```

```
        $path = storage_path() . "/app/JsonStore/users.json"; // ie:  
        /var/www/laravel/app/storage/json/filename.json
```

```
        $json = json_decode(file_get_contents($path), true);  
        return response()->json($json);
```

```
    }
```

```
    public function searchByOrganizationId($id)
```

```
    {
```

```
        $path = storage_path() . "/app/JsonStore/users.json";
```

```
        $json = json_decode(file_get_contents($path), true);
```

```
        $data = array_filter($json);
```

```
        $data = collect($data)->where('organization_id', $id);
```

```
        return response()->json($data);
```

```
    }
```

```
    public function search($name)
```

```
    {
```

```
        $json_ticket = Storage::disk('local')->get('JsonStore/tickets.json');
```

```

        $json_user = Storage::disk('local')->get('JsonStore/users.json');
        $json_organization =
Storage::disk('local')->get('JsonStore/organizations.json');
        $data = json_decode($json_ticket, true);
        $data_user = json_decode($json_user, true);
        $data_organization = json_decode($json_organization, true);

        $datas = array_filter($data);
        $data_users = array_filter($data_user);
        $data_organizations = array_filter($data_organization);

        $data_organization_id = collect($data_users)->where('name',
$name)->pluck('organization_id');
        $data_id = collect($data_users)->where('name',
$name)->pluck('_id');

        $data_ticket = collect($datas)->where('assignee_id',
$data_id->first())->pluck('subject');

        $data_ticket_submitter = collect($datas)->where('submitter_id',
$data_id->first())->pluck('subject');

        $data_user_organization =
collect($data_users)->where('organization_id',
$data_organization_id->first())->pluck('name');

        $value = array(
            $data_ticket_submitter->all(),
            $data_ticket->all(),
            $data_user_organization->all()
        );

        return response()->json($value);
    }
}

```

```
<?php
```

```
namespace App\Services\Interfaces;
```

```
interface IService
```

```
{
```

```
    //
```

```
}
```

```
<?php
```

```
namespace App\Services;
```

```
use App\Services\Interfaces\IService;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
abstract class BaseService implements IService
```

```
{
```

```
    // protected $repo;
```

```
    public function checkAuthenticate($id): Model
```

```
    {
```

```
        $model = $this->repo->getDataById($id);
```

```
        if (!$model->exists) abort(404, "Not Found");
```

```
        if ($model->user_id != request()->user()->id) abort(403,  
"Unauthorized action");
```

```
        return $model;
```

```
    }
```

```
    public function delete(int $id): bool
```

```
    {
```

```
        return $this->repo->delete($id);
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App\Services;
```

```
use App\Repositories\OrganizationRepository;
```

```
class OrganizationService
```

```
{
```

```
    protected $repo;
```

```
    public function __construct(OrganizationRepository $repo)
```

```
    {
```

```
        $this->repo = $repo;
```

```
    }
```

```
    public function index()
```

```
    {
```

```
        $json = $this->repo->json();
```

```
        return response()->json($json);
```

```
    }
```

```
    public function searchByOrganizationId($id)
```

```
    {
```

```
        $json = $this->repo->searchByOrganizationId($id);
```

```
        return response()->json($json);
```

```
    }
```

```
    public function search($id)
```

```
{
    $json = $this->repo->search($id);
    return response()->json($json);
}
}
```

```
<?php
```

```
namespace App\Services;
```

```
use App\Repositories\TicketRepository;
```

```
class TicketService
```

```
{
```

```
    protected $repo;
```

```
    public function __construct(TicketRepository $repo)
```

```
    {
```

```
        $this->repo = $repo;
```

```
    }
```

```
    public function index()
```

```
    {
```

```
        $json = $this->repo->json();
```

```
        return response()->json($json);
```

```
    }
```

```
    public function searchByOrganizationId($id)
```

```
    {
```

```
        $json = $this->repo->searchByOrganizationId($id);
        return response()->json($json);

    }

    public function search($subject)
    {
        $json = $this->repo->search($subject);
        return response()->json($json);

    }

}
```

```
<?php
```

```
namespace App\Services;
```

```
use App\Repositories\UserRepository;
```

```
class UserService
```

```
{
```

```
    protected $repo;
```

```
    public function __construct(UserRepository $repo)
```

```
    {
```

```
        $this->repo = $repo;
```

```
    }
```

```
    public function index()
```

```
    {
```

```
        $json = $this->repo->json();
```



```

        return response()->json($json);
    }

    public function searchByOrganizationId($id)
    {
        $json = $this->repo->searchByOrganizationId($id);
        return response()->json($json);
    }

    public function search($name)
    {
        $json = $this->repo->search($name);
        return response()->json($json);
    }
}

```

```

<?php

```

```

namespace App;

```

```

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Laravel\Passport\HasApiTokens;

```

```

class User extends Authenticatable
{
    use HasApiTokens, Notifiable;

```

```

/**

```

```

* The attributes that are mass assignable.
*
* @var array
*/
protected $fillable = [
    'name', 'email', 'password',
];

/**
 * The attributes that should be hidden for arrays.
 *
 * @var array
 */
protected $hidden = [
    'password', 'remember_token',
];

/**
 * The attributes that should be cast to native types.
 *
 * @var array
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];
}

```

```

<?php return array (
    'app' =>
        array (
            'name' => 'Laravel',
            'env' => 'local',
            'debug' => true,
            'url' => 'http://localhost',
            'asset_url' => NULL,

```

```
'timezone' => 'UTC',
'locale' => 'en',
'fallback_locale' => 'en',
'faker_locale' => 'en_US',
'key' =>
'base64:/cjWNNov2I/oYm0PIC7vSCj1pV6xKszcgOlslj0ymD+U=',
'cipher' => 'AES-256-CBC',
'providers' =>
array (
    0 => 'Illuminate\Auth\AuthServiceProvider',
    1 => 'Illuminate\Broadcasting\BroadcastServiceProvider',
    2 => 'Illuminate\Bus\BusServiceProvider',
    3 => 'Illuminate\Cache\CacheServiceProvider',
    4 =>
'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
    5 => 'Illuminate\Cookie\CookieServiceProvider',
    6 => 'Illuminate\Database\DatabaseServiceProvider',
    7 => 'Illuminate\Encryption\EncryptionServiceProvider',
    8 => 'Illuminate\Filesystem\FilesystemServiceProvider',
    9 => 'Illuminate\Foundation\Providers\FoundationServiceProvider',
    10 => 'Illuminate\Hashing\HashServiceProvider',
    11 => 'Illuminate\Mail\MailServiceProvider',
    12 => 'Illuminate\Notifications\NotificationServiceProvider',
    13 => 'Illuminate\Pagination\PaginationServiceProvider',
    14 => 'Illuminate\Pipeline\PipelineServiceProvider',
    15 => 'Illuminate\Queue\QueueServiceProvider',
    16 => 'Illuminate\Redis\RedisServiceProvider',
    17 => 'Illuminate\Auth\Passwords>PasswordResetServiceProvider',
    18 => 'Illuminate\Session\SessionServiceProvider',
    19 => 'Illuminate\Translation\TranslationServiceProvider',
    20 => 'Illuminate\Validation\ValidationServiceProvider',
    21 => 'Illuminate\View\ViewServiceProvider',
    22 => 'App\Providers\AppServiceProvider',
    23 => 'App\Providers\AuthServiceProvider',
    24 => 'App\Providers\EventServiceProvider',
    25 => 'App\Providers\RouteServiceProvider',
),
```

```

'aliases' =>
array (
    'App' => 'Illuminate\Support\Facades\App',
    'Artisan' => 'Illuminate\Support\Facades\Artisan',
    'Auth' => 'Illuminate\Support\Facades\Auth',
    'Blade' => 'Illuminate\Support\Facades\Blade',
    'Broadcast' => 'Illuminate\Support\Facades\Broadcast',
    'Bus' => 'Illuminate\Support\Facades\Bus',
    'Cache' => 'Illuminate\Support\Facades\Cache',
    'Config' => 'Illuminate\Support\Facades\Config',
    'Cookie' => 'Illuminate\Support\Facades\Cookie',
    'Crypt' => 'Illuminate\Support\Facades\Crypt',
    'DB' => 'Illuminate\Support\Facades\DB',
    'Eloquent' => 'Illuminate\Database\Eloquent\Model',
    'Event' => 'Illuminate\Support\Facades\Event',
    'File' => 'Illuminate\Support\Facades\File',
    'Gate' => 'Illuminate\Support\Facades\Gate',
    'Hash' => 'Illuminate\Support\Facades\Hash',
    'Lang' => 'Illuminate\Support\Facades\Lang',
    'Log' => 'Illuminate\Support\Facades\Log',
    'Mail' => 'Illuminate\Support\Facades\Mail',
    'Notification' => 'Illuminate\Support\Facades\Notification',
    'Password' => 'Illuminate\Support\Facades>Password',
    'Queue' => 'Illuminate\Support\Facades\Queue',
    'Redirect' => 'Illuminate\Support\Facades\Redirect',
    'Redis' => 'Illuminate\Support\Facades\Redis',
    'Request' => 'Illuminate\Support\Facades\Request',
    'Response' => 'Illuminate\Support\Facades\Response',
    'Route' => 'Illuminate\Support\Facades\Route',
    'Schema' => 'Illuminate\Support\Facades\Schema',
    'Session' => 'Illuminate\Support\Facades\Session',
    'Storage' => 'Illuminate\Support\Facades\Storage',
    'URL' => 'Illuminate\Support\Facades\URL',
    'Validator' => 'Illuminate\Support\Facades\Validator',
    'View' => 'Illuminate\Support\Facades\View',
),
),

```

```
'auth' =>
array (
  'defaults' =>
    array (
      'guard' => 'web',
      'passwords' => 'users',
    ),
  'guards' =>
    array (
      'web' =>
        array (
          'driver' => 'session',
          'provider' => 'users',
        ),
      'api' =>
        array (
          'driver' => 'passport',
          'provider' => 'users',
        ),
    ),
  'providers' =>
    array (
      'users' =>
        array (
          'driver' => 'eloquent',
          'model' => 'App\\User',
        ),
    ),
  'passwords' =>
    array (
      'users' =>
        array (
          'provider' => 'users',
          'table' => 'password_resets',
          'expire' => 60,
        ),
    ),
)
```

```
),
'broadcasting' =>
array (
  'default' => 'log',
  'connections' =>
array (
  'pusher' =>
array (
  'driver' => 'pusher',
  'key' => "",
  'secret' => "",
  'app_id' => "",
  'options' =>
array (
  'cluster' => 'mt1',
  'encrypted' => true,
  ),
  ),
  'redis' =>
array (
  'driver' => 'redis',
  'connection' => 'default',
  ),
  'log' =>
array (
  'driver' => 'log',
  ),
  'null' =>
array (
  'driver' => 'null',
  ),
  ),
),
'cache' =>
array (
  'default' => 'file',
  'stores' =>
```

```
array (  
  'apc' =>  
    array (  
      'driver' => 'apc',  
    ),  
  'array' =>  
    array (  
      'driver' => 'array',  
    ),  
  'database' =>  
    array (  
      'driver' => 'database',  
      'table' => 'cache',  
      'connection' => NULL,  
    ),  
  'file' =>  
    array (  
      'driver' => 'file',  
      'path' =>  
'/media/user/Data/Projects/02/final/01/Final/blog/storage/framework/cache/data',  
    ),  
  'memcached' =>  
    array (  
      'driver' => 'memcached',  
      'persistent_id' => NULL,  
      'sasI' =>  
        array (  
          0 => NULL,  
          1 => NULL,  
        ),  
      'options' =>  
        array (  
        ),  
      'servers' =>  
        array (  
          0 =>
```

```

        array (
            'host' => '127.0.0.1',
            'port' => 11211,
            'weight' => 100,
        ),
    ),
),
'redis' =>
array (
    'driver' => 'redis',
    'connection' => 'cache',
),
),
'prefix' => 'laravel_cache',
),
'database' =>
array (
    'default' => 'mysql',
    'connections' =>
array (
    'sqlite' =>
array (
        'driver' => 'sqlite',
        'database' => 'login',
        'prefix' => "",
        'foreign_key_constraints' => true,
    ),
    'mysql' =>
array (
        'driver' => 'mysql',
        'host' => '127.0.0.1',
        'port' => '3306',
        'database' => 'login',
        'username' => 'root',
        'password' => '12345678',
        'unix_socket' => "",
        'charset' => 'utf8mb4',
    ),

```



```
'collation' => 'utf8mb4_unicode_ci',
'prefix' => '',
'prefix_indexes' => true,
'strict' => true,
'engine' => NULL,
),
'pgsql' =>
array (
    'driver' => 'pgsql',
    'host' => '127.0.0.1',
    'port' => '3306',
    'database' => 'login',
    'username' => 'root',
    'password' => '12345678',
    'charset' => 'utf8',
    'prefix' => '',
    'prefix_indexes' => true,
    'schema' => 'public',
    'sslmode' => 'prefer',
),
'sqlsrv' =>
array (
    'driver' => 'sqlsrv',
    'host' => '127.0.0.1',
    'port' => '3306',
    'database' => 'login',
    'username' => 'root',
    'password' => '12345678',
    'charset' => 'utf8',
    'prefix' => '',
    'prefix_indexes' => true,
),
),
'migrations' => 'migrations',
'redis' =>
array (
    'client' => 'predis',
```

```
'default' =>
array (
    'host' => '127.0.0.1',
    'password' => NULL,
    'port' => '6379',
    'database' => 0,
),
'cache' =>
array (
    'host' => '127.0.0.1',
    'password' => NULL,
    'port' => '6379',
    'database' => 1,
),
),
'filesystems' =>
array (
    'default' => 'local',
    'cloud' => 's3',
    'disks' =>
array (
    'local' =>
array (
    'driver' => 'local',
    'root' =>
'/media/user/Data/Projects/02/final/01/Final/blog/storage/app',
),
    'public' =>
array (
    'driver' => 'local',
    'root' =>
'/media/user/Data/Projects/02/final/01/Final/blog/storage/app/public',
    'url' => 'http://localhost/storage',
    'visibility' => 'public',
),
    's3' =>
```

```
array (  
    'driver' => 's3',  
    'key' => NULL,  
    'secret' => NULL,  
    'region' => NULL,  
    'bucket' => NULL,  
    'url' => NULL,  
),  
,  
,  
'hashing' =>  
array (  
    'driver' => 'bcrypt',  
    'bcrypt' =>  
    array (  
        'rounds' => 10,  
    ),  
    'argon' =>  
    array (  
        'memory' => 1024,  
        'threads' => 2,  
        'time' => 2,  
    ),  
,  
,  
'logging' =>  
array (  
    'default' => 'stack',  
    'channels' =>  
    array (  
        'stack' =>  
        array (  
            'driver' => 'stack',  
            'channels' =>  
            array (  
                0 => 'daily',  
            ),  
            'ignore_exceptions' => false,
```

```

),
'single' =>
array (
    'driver' => 'single',
    'path' =>
'/media/user/Data/Projects/02/final/01/Final/blog/storage/logs/laravel.log',
    'level' => 'debug',
),
'daily' =>
array (
    'driver' => 'daily',
    'path' =>
'/media/user/Data/Projects/02/final/01/Final/blog/storage/logs/laravel.log',
    'level' => 'debug',
    'days' => 14,
),
'slack' =>
array (
    'driver' => 'slack',
    'url' => NULL,
    'username' => 'Laravel Log',
    'emoji' => ':boom:',
    'level' => 'critical',
),
'papertrail' =>
array (
    'driver' => 'monolog',
    'level' => 'debug',
    'handler' => 'Monolog\\Handler\\SyslogUdpHandler',
    'handler_with' =>
array (
        'host' => NULL,
        'port' => NULL,
    ),
),
'stderr' =>
array (

```

```
'driver' => 'monolog',
'handler' => 'Monolog\\Handler\\StreamHandler',
'formatter' => NULL,
'with' =>
array (
    'stream' => 'php://stderr',
),
),
'syslog' =>
array (
    'driver' => 'syslog',
    'level' => 'debug',
),
'errorlog' =>
array (
    'driver' => 'errorlog',
    'level' => 'debug',
),
),
'mail' =>
array (
    'driver' => 'smtp',
    'host' => 'smtp.mailtrap.io',
    'port' => '2525',
    'from' =>
array (
    'address' => 'hello@example.com',
    'name' => 'Example',
),
    'encryption' => NULL,
    'username' => NULL,
    'password' => NULL,
    'sendmail' => '/usr/sbin/sendmail -bs',
    'markdown' =>
array (
    'theme' => 'default',
```

```
'paths' =>
array (
    0 =>
'/media/user/Data/Projects/02/final/01/Final/blog/resources/views/vendor
/mail',
),
),
'log_channel' => NULL,
),
'queue' =>
array (
    'default' => 'sync',
    'connections' =>
array (
        'sync' =>
array (
            'driver' => 'sync',
        ),
        'database' =>
array (
            'driver' => 'database',
            'table' => 'jobs',
            'queue' => 'default',
            'retry_after' => 90,
        ),
        'beanstalkd' =>
array (
            'driver' => 'beanstalkd',
            'host' => 'localhost',
            'queue' => 'default',
            'retry_after' => 90,
        ),
        'sqs' =>
array (
            'driver' => 'sqs',
            'key' => 'your-public-key',
            'secret' => 'your-secret-key',
```

```
'prefix' => 'https://sqs.us-east-1.amazonaws.com/your-account-id',
'queue' => 'your-queue-name',
'region' => 'us-east-1',
),
'redis' =>
array (
    'driver' => 'redis',
    'connection' => 'default',
    'queue' => 'default',
    'retry_after' => 90,
    'block_for' => NULL,
),
),
'failed' =>
array (
    'database' => 'mysql',
    'table' => 'failed_jobs',
),
),
'services' =>
array (
    'mailgun' =>
array (
    'domain' => NULL,
    'secret' => NULL,
    'endpoint' => 'api.mailgun.net',
),
    'ses' =>
array (
    'key' => NULL,
    'secret' => NULL,
    'region' => 'us-east-1',
),
    'sparkpost' =>
array (
    'secret' => NULL,
),
),
```

```

'stripe' =>
array (
    'model' => 'App\\User',
    'key' => NULL,
    'secret' => NULL,
    'webhook' =>
        array (
            'secret' => NULL,
            'tolerance' => 300,
        ),
    ),
),
'session' =>
array (
    'driver' => 'file',
    'lifetime' => '120',
    'expire_on_close' => false,
    'encrypt' => false,
    'files' =>
'/media/user/Data/Projects/02/final/01/Final/blog/storage/framework/sess
ions',
    'connection' => NULL,
    'table' => 'sessions',
    'store' => NULL,
    'lottery' =>
        array (
            0 => 2,
            1 => 100,
        ),
    'cookie' => 'laravel_session',
    'path' => '/',
    'domain' => NULL,
    'secure' => false,
    'http_only' => true,
    'same_site' => NULL,
),
'view' =>

```



```
array (
  'paths' =>
    array (
      0 =>
'/media/user/Data/Projects/02/final/01/Final/blog/resources/views',
    ),
    'compiled' =>
'/media/user/Data/Projects/02/final/01/Final/blog/storage/framework/view
s',
  ),
  'debug-server' =>
    array (
      'host' => 'tcp://127.0.0.1:9912',
    ),
  'passport' =>
    array (
      'private_key' => NULL,
      'public_key' => NULL,
    ),
  'trustedproxy' =>
    array (
      'proxies' => NULL,
      'headers' => 30,
    ),
  'tinker' =>
    array (
      'commands' =>
        array (
        ),
      'dont_alias' =>
        array (
          0 => 'App\\Nova',
        ),
    ),
);
```

```
<?php return array (
    'beyondcode/laravel-dump-server' =>
        array (
            'providers' =>
                array (
                    0 => 'BeyondCode\\DumpServer\\DumpServerServiceProvider',
                ),
            ),
    'fideloper/proxy' =>
        array (
            'providers' =>
                array (
                    0 => 'Fideloper\\Proxy\\TrustedProxyServiceProvider',
                ),
            ),
    'laravel/nexmo-notification-channel' =>
        array (
            'providers' =>
                array (
                    0 => 'Illuminate\\Notifications\\NexmoChannelServiceProvider',
                ),
            ),
    'laravel/passport' =>
        array (
            'providers' =>
                array (
                    0 => 'Laravel\\Passport\\PassportServiceProvider',
                ),
            ),
    'laravel/slack-notification-channel' =>
        array (
            'providers' =>
                array (
                    0 => 'Illuminate\\Notifications\\SlackChannelServiceProvider',
                ),
            ),
    'laravel/tinker' =>
```

```

array (
    'providers' =>
        array (
            0 => 'Laravel\\Tinker\\TinkerServiceProvider',
        ),
    ),
    'nesbot/carbon' =>
        array (
            'providers' =>
                array (
                    0 => 'Carbon\\Laravel\\ServiceProvider',
                ),
            ),
    'nunomaduro/collision' =>
        array (
            'providers' =>
                array (
                    0 =>
'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',
                ),
            ),
        );

```

```

<?php return array (
    'providers' =>
        array (
            0 => 'Illuminate\\Auth\\AuthServiceProvider',
            1 => 'Illuminate\\Broadcasting\\BroadcastServiceProvider',
            2 => 'Illuminate\\Bus\\BusServiceProvider',
            3 => 'Illuminate\\Cache\\CacheServiceProvider',
            4 =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
            5 => 'Illuminate\\Cookie\\CookieServiceProvider',
            6 => 'Illuminate\\Database\\DatabaseServiceProvider',
            7 => 'Illuminate\\Encryption\\EncryptionServiceProvider',
            8 => 'Illuminate\\Filesystem\\FilesystemServiceProvider',

```

```

9 => 'Illuminate\\Foundation\\Providers\\FoundationServiceProvider',
10 => 'Illuminate\\Hashing\\HashServiceProvider',
11 => 'Illuminate\\Mail\\MailServiceProvider',
12 => 'Illuminate\\Notifications\\NotificationServiceProvider',
13 => 'Illuminate\\Pagination\\PaginationServiceProvider',
14 => 'Illuminate\\Pipeline\\PipelineServiceProvider',
15 => 'Illuminate\\Queue\\QueueServiceProvider',
16 => 'Illuminate\\Redis\\RedisServiceProvider',
17 => 'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
18 => 'Illuminate\\Session\\SessionServiceProvider',
19 => 'Illuminate\\Translation\\TranslationServiceProvider',
20 => 'Illuminate\\Validation\\ValidationServiceProvider',
21 => 'Illuminate\\View\\ViewServiceProvider',
22 => 'BeyondCode\\DumpServer\\DumpServerServiceProvider',
23 => 'Fideloper\\Proxy\\TrustedProxyServiceProvider',
24 => 'Illuminate\\Notifications\\NexmoChannelServiceProvider',
25 => 'Laravel\\Passport\\PassportServiceProvider',
26 => 'Illuminate\\Notifications\\SlackChannelServiceProvider',
27 => 'Laravel\\Tinker\\TinkerServiceProvider',
28 => 'Carbon\\Laravel\\ServiceProvider',
29 =>
'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',
30 => 'App\\Providers\\AppServiceProvider',
31 => 'App\\Providers\\AuthServiceProvider',
32 => 'App\\Providers\\EventServiceProvider',
33 => 'App\\Providers\\RouteServiceProvider',
),
'eager' =>
array (
0 => 'Illuminate\\Auth\\AuthServiceProvider',
1 => 'Illuminate\\Cookie\\CookieServiceProvider',
2 => 'Illuminate\\Database\\DatabaseServiceProvider',
3 => 'Illuminate\\Encryption\\EncryptionServiceProvider',
4 => 'Illuminate\\Filesystem\\FilesystemServiceProvider',
5 => 'Illuminate\\Foundation\\Providers\\FoundationServiceProvider',
6 => 'Illuminate\\Notifications\\NotificationServiceProvider',
7 => 'Illuminate\\Pagination\\PaginationServiceProvider',

```

```

8 => 'Illuminate\\Session\\SessionServiceProvider',
9 => 'Illuminate\\View\\ViewServiceProvider',
10 => 'BeyondCode\\DumpServer\\DumpServerServiceProvider',
11 => 'Fideloper\\Proxy\\TrustedProxyServiceProvider',
12 => 'Illuminate\\Notifications\\NexmoChannelServiceProvider',
13 => 'Laravel\\Passport\\PassportServiceProvider',
14 => 'Illuminate\\Notifications\\SlackChannelServiceProvider',
15 => 'Carbon\\Laravel\\ServiceProvider',
16 => 'App\\Providers\\AppServiceProvider',
17 => 'App\\Providers\\AuthServiceProvider',
18 => 'App\\Providers\\EventServiceProvider',
19 => 'App\\Providers\\RouteServiceProvider',
),
'deferred' =>
array (
    'Illuminate\\Broadcasting\\BroadcastManager' =>
'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Contracts\\Broadcasting\\Factory' =>
'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Contracts\\Broadcasting\\Broadcaster' =>
'Illuminate\\Broadcasting\\BroadcastServiceProvider',
    'Illuminate\\Bus\\Dispatcher' => 'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Contracts\\Bus\\Dispatcher' =>
'Illuminate\\Bus\\BusServiceProvider',
    'Illuminate\\Contracts\\Bus\\QueueingDispatcher' =>
'Illuminate\\Bus\\BusServiceProvider',
    'cache' => 'Illuminate\\Cache\\CacheServiceProvider',
    'cache.store' => 'Illuminate\\Cache\\CacheServiceProvider',
    'memcached.connector' => 'Illuminate\\Cache\\CacheServiceProvider',
    'command.cache.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.cache.forget' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.clear-compiled' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'command.auth.resets.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

'command.config.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.config.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.down' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.environment' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.key.generate' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.fresh' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.install' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.refresh' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.reset' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.rollback' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.status' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.optimize' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.optimize.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.package.discover' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.preset' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.failed' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.flush' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

'command.queue.forget' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.listen' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.restart' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.retry' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.work' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.route.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.route.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.route.list' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.seed' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'Illuminate\\Console\\Scheduling\\ScheduleFinishCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'Illuminate\\Console\\Scheduling\\ScheduleRunCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.storage.link' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.up' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.view.cache' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.view.clear' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.app.name' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.auth.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.cache.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

'command.channel.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.console.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.controller.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.event.generate' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.event.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.exception.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.factory.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.job.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.listener.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.mail.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.middleware.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.migrate.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.model.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.notification.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.notification.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.observer.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.policy.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.provider.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

'command.queue.failed-table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.queue.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.request.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.resource.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.rule.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.seeder.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.session.table' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.serve' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.test.make' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'command.vendor.publish' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'migrator' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'migration.repository' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'migration.creator' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'composer' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
 'hash' => 'Illuminate\\Hashing\\HashServiceProvider',
 'hash.driver' => 'Illuminate\\Hashing\\HashServiceProvider',
 'mailer' => 'Illuminate\\Mail\\MailServiceProvider',
 'swift.mailer' => 'Illuminate\\Mail\\MailServiceProvider',
 'swift.transport' => 'Illuminate\\Mail\\MailServiceProvider',
 'Illuminate\\Mail\\Markdown' => 'Illuminate\\Mail\\MailServiceProvider',
 'Illuminate\\Contracts\\Pipeline\\Hub' =>
'Illuminate\\Pipeline\\PipelineServiceProvider',
 'queue' => 'Illuminate\\Queue\\QueueServiceProvider',

```

'queue.worker' => 'Illuminate\Queue\QueueServiceProvider',
'queue.listener' => 'Illuminate\Queue\QueueServiceProvider',
'queue.failer' => 'Illuminate\Queue\QueueServiceProvider',
'queue.connection' => 'Illuminate\Queue\QueueServiceProvider',
'redis' => 'Illuminate\Redis\RedisServiceProvider',
'redis.connection' => 'Illuminate\Redis\RedisServiceProvider',
'auth.password' =>
'Illuminate\Auth\Passwords\PasswordResetServiceProvider',
'auth.password.broker' =>
'Illuminate\Auth\Passwords\PasswordResetServiceProvider',
'translator' => 'Illuminate\Translation\TranslationServiceProvider',
'translation.loader' =>
'Illuminate\Translation\TranslationServiceProvider',
'validator' => 'Illuminate\Validation\ValidationServiceProvider',
'validation.presence' =>
'Illuminate\Validation\ValidationServiceProvider',
'command.tinker' => 'Laravel\Tinker\TinkerServiceProvider',
'NunoMaduro\Collision\Contracts\Provider' =>
'NunoMaduro\Collision\Adapters\Laravel\CollisionServiceProvider',
),
'when' =>
array (
'Illuminate\Broadcasting\BroadcastServiceProvider' =>
array (
),
'Illuminate\Bus\BusServiceProvider' =>
array (
),
'Illuminate\Cache\CacheServiceProvider' =>
array (
),
'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider'
=>
array (
),
'Illuminate\Hashing\HashServiceProvider' =>
array (

```

```

    ),
    'Illuminate\Mail\MailServiceProvider' =>
    array (
    ),
    'Illuminate\Pipeline\PipelineServiceProvider' =>
    array (
    ),
    'Illuminate\Queue\QueueServiceProvider' =>
    array (
    ),
    'Illuminate\Redis\RedisServiceProvider' =>
    array (
    ),
    'Illuminate\Auth\Passwords>PasswordResetServiceProvider' =>
    array (
    ),
    'Illuminate\Translation\TranslationServiceProvider' =>
    array (
    ),
    'Illuminate\Validation\ValidationServiceProvider' =>
    array (
    ),
    'Laravel\Tinker\TinkerServiceProvider' =>
    array (
    ),
    'NunoMaduro\Collision\Adapters\Laravel\CollisionServiceProvider'
=>
    array (
    ),
    ),
);

```

<?php

return [

```
/*
|-----
| Application Name
|-----
|
| This value is the name of your application. This value is used when
the
| framework needs to place the application's name in a notification or
| any other location as required by the application or its packages.
|
*/
```

```
'name' => env('APP_NAME', 'Laravel'),
```

```
/*
|-----
| Application Environment
|-----
|
| This value determines the "environment" your application is currently
| running in. This may determine how you prefer to configure various
| services the application utilizes. Set this in your ".env" file.
|
*/
```

```
'env' => env('APP_ENV', 'production'),
```

```
/*
|-----
| Application Debug Mode
|-----
|
| When your application is in debug mode, detailed error messages
with
| stack traces will be shown on every error that occurs within your
| application. If disabled, a simple generic error page is shown.
|
```

```
*/
```

```
'debug' => env('APP_DEBUG', false),
```

```
/*
```

```
|-----
```

```
| Application URL
```

```
|-----
```

```
|
```

| This URL is used by the console to properly generate URLs when
using

| the Artisan command line tool. You should set this to the root of
| your application so that it is used when running Artisan tasks.

```
|
```

```
*/
```

```
'url' => env('APP_URL', 'http://localhost'),
```

```
'asset_url' => env('ASSET_URL', null),
```

```
/*
```

```
|-----
```

```
| Application Timezone
```

```
|-----
```

```
|
```

| Here you may specify the default timezone for your application,
which

| will be used by the PHP date and date-time functions. We have gone
| ahead and set this to a sensible default for you out of the box.

```
|
```

```
*/
```

```
'timezone' => 'UTC',
```

```
/*
```

```
|-----
```

```
| Application Locale Configuration
```

```
|-----  
|  
| The application locale determines the default locale that will be used  
| by the translation service provider. You are free to set this value  
| to any of the locales which will be supported by the application.  
|  
*/
```

```
'locale' => 'en',
```

```
/*  
|-----  
| Application Fallback Locale  
|-----  
|  
| The fallback locale determines the locale to use when the current  
one  
| is not available. You may change the value to correspond to any of  
| the language folders that are provided through your application.  
|  
*/
```

```
'fallback_locale' => 'en',
```

```
/*  
|-----  
| Faker Locale  
|-----  
|  
| This locale will be used by the Faker PHP library when generating  
fake  
| data for your database seeds. For example, this will be used to get  
| localized telephone numbers, street address information and more.  
|  
*/
```

```
'faker_locale' => 'en_US',
```

```
/*
|-----
| Encryption Key
|-----
|
| This key is used by the Illuminate encrypter service and should be
set
| to a random, 32 character string, otherwise these encrypted strings
| will not be safe. Please do this before deploying an application!
|
*/
```

```
'key' => env('APP_KEY'),
```

```
'cipher' => 'AES-256-CBC',
```

```
/*
|-----
| Autoloaded Service Providers
|-----
|
| The service providers listed here will be automatically loaded on the
| request to your application. Feel free to add your own services to
| this array to grant expanded functionality to your applications.
|
*/
```

```
'providers' => [
```

```
    /*
    * Laravel Framework Service Providers...
    */
    Illuminate\Auth\AuthServiceProvider::class,
    Illuminate\Broadcasting\BroadcastServiceProvider::class,
    Illuminate\Bus\BusServiceProvider::class,
    Illuminate\Cache\CacheServiceProvider::class,
```

```
Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,  
    Illuminate\Cookie\CookieServiceProvider::class,  
    Illuminate\Database\DatabaseServiceProvider::class,  
    Illuminate\Encryption\EncryptionServiceProvider::class,  
    Illuminate\Filesystem\FilesystemServiceProvider::class,  
    Illuminate\Foundation\Providers\FoundationServiceProvider::class,  
    Illuminate\Hashing\HashServiceProvider::class,  
    Illuminate\Mail\MailServiceProvider::class,  
    Illuminate\Notifications\NotificationServiceProvider::class,  
    Illuminate\Pagination\PaginationServiceProvider::class,  
    Illuminate\Pipeline\PipelineServiceProvider::class,  
    Illuminate\Queue\QueueServiceProvider::class,  
    Illuminate\Redis\RedisServiceProvider::class,  
    Illuminate\Auth\Passwords>PasswordResetServiceProvider::class,  
    Illuminate\Session\SessionServiceProvider::class,  
    Illuminate\Translation\TranslationServiceProvider::class,  
    Illuminate\Validation\ValidationServiceProvider::class,  
    Illuminate\View\ViewServiceProvider::class,
```

```
/*
```

```
 * Package Service Providers...
```

```
*/
```

```
/*
```

```
 * Application Service Providers...
```

```
*/
```

```
App\Providers\AppServiceProvider::class,  
App\Providers\AuthServiceProvider::class,  
// App\Providers\BroadcastServiceProvider::class,  
App\Providers\EventServiceProvider::class,  
App\Providers\RouteServiceProvider::class,
```

```
],
```

```
/*
```

```
|-----
```


| Class Aliases

| -----
|
| This array of class aliases will be registered when this application
| is started. However, feel free to register as many as you wish as
| the aliases are "lazy" loaded so they don't hinder performance.
|

*/

'aliases' => [

'App' => Illuminate\Support\Facades\App::class,
'Artisan' => Illuminate\Support\Facades\Artisan::class,
'Auth' => Illuminate\Support\Facades\Auth::class,
'Blade' => Illuminate\Support\Facades\Blade::class,
'Broadcast' => Illuminate\Support\Facades\Broadcast::class,
'Bus' => Illuminate\Support\Facades\Bus::class,
'Cache' => Illuminate\Support\Facades\Cache::class,
'Config' => Illuminate\Support\Facades\Config::class,
'Cookie' => Illuminate\Support\Facades\Cookie::class,
'Crypt' => Illuminate\Support\Facades\Crypt::class,
'DB' => Illuminate\Support\Facades\DB::class,
'Eloquent' => Illuminate\Database\Eloquent\Model::class,
'Event' => Illuminate\Support\Facades\Event::class,
'File' => Illuminate\Support\Facades\File::class,
'Gate' => Illuminate\Support\Facades\Gate::class,
'Hash' => Illuminate\Support\Facades\Hash::class,
'Lang' => Illuminate\Support\Facades\Lang::class,
'Log' => Illuminate\Support\Facades\Log::class,
'Mail' => Illuminate\Support\Facades\Mail::class,
'Notification' => Illuminate\Support\Facades\Notification::class,
'Password' => Illuminate\Support\Facades>Password::class,
'Queue' => Illuminate\Support\Facades\Queue::class,
'Redirect' => Illuminate\Support\Facades\Redirect::class,
'Redis' => Illuminate\Support\Facades\Redis::class,
'Request' => Illuminate\Support\Facades\Request::class,
'Response' => Illuminate\Support\Facades\Response::class,

```

        'Route' => Illuminate\Support\Facades\Route::class,
        'Schema' => Illuminate\Support\Facades\Schema::class,
        'Session' => Illuminate\Support\Facades\Session::class,
        'Storage' => Illuminate\Support\Facades\Storage::class,
        'URL' => Illuminate\Support\Facades\URL::class,
        'Validator' => Illuminate\Support\Facades\Validator::class,
        'View' => Illuminate\Support\Facades\View::class,

    ],

];



---


<?php

return [

    /*
    |-----
    | Authentication Defaults
    |-----
    |
    | This option controls the default authentication "guard" and password
    | reset options for your application. You may change these defaults
    | as required, but they're a perfect start for most applications.
    |
    */

    'defaults' => [
        'guard' => 'web',
        'passwords' => 'users',
    ],

    /*
    |-----
    | Authentication Guards
    |-----

```

```

|-----
|
| Next, you may define every authentication guard for your application.
| Of course, a great default configuration has been defined for you
| here which uses session storage and the Eloquent user provider.
|
| All authentication drivers have a user provider. This defines how the
| users are actually retrieved out of your database or other storage
| mechanisms used by this application to persist your user's data.
|
| Supported: "session", "token"
|
| */

```

```

'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],

    'api' => [
//        'driver' => 'token',
        'driver' => 'passport',
        'provider' => 'users',
    ],
],

```

```

/*
|-----
| User Providers
|-----
|
| All authentication drivers have a user provider. This defines how the
| users are actually retrieved out of your database or other storage
| mechanisms used by this application to persist your user's data.
|

```

| If you have multiple user tables or models you may configure multiple
| sources which represent each model / table. These sources may then
| be assigned to any extra authentication guards you have defined.
|
| Supported: "database", "eloquent"
|
*/

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\User::class,  
    ],
```

```
    // 'users' => [  
    //     'driver' => 'database',  
    //     'table' => 'users',  
    // ],  
],
```

```
/*
```

```
|-----  
| Resetting Passwords  
|-----  
|
```

| You may specify multiple password reset configurations if you have more
| than one user table or model in the application and you want to have
| separate password reset settings based on the specific user types.
|
| The expire time is the number of minutes that the reset token should be
| considered valid. This security feature keeps tokens short-lived so
| they have less time to be guessed. You may change this as needed.
|

```

        */

        'passwords' => [
            'users' => [
                'provider' => 'users',
                'table' => 'password_resets',
                'expire' => 60,
            ],
        ],
    ],
];

```

```
<?php
```

```

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

```

```

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();

```

```

    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('users');
}
}

```

<?php

```

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('password_resets', function (Blueprint $table) {
            $table->string('email')->index();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });
    }
}

```

```

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('password_resets');
}
}

```

```

<?php

```

```

use Illuminate\Http\Request;

```

```

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

```

```

//Route::middleware('auth:api')->get('/user', function (Request $request)
{
//    return $request->user();
//});

```

```

Route::group(['middleware' => ['json.response']], function () {

```

```

Route::middleware('auth:api')->get('/user', function (Request $request)
{
    return $request->user();
});

// public routes
Route::post('/login', 'api\authcontroller@login')->name('login.api');
Route::post('/register',
'api\authcontroller@register')->name('register.api');

// private routes
Route::middleware('auth:api')->group(function () {
    Route::get('/logout', 'Api\AuthController@logout')->name('logout');
});

// json store routes list
Route::middleware('auth:api')->group(function () {

    Route::get('/user', 'Api\UserController@index')->name('user');
    Route::get('/ticket', 'Api\TicketController@index')->name('user');
    Route::get('/organization',
'Api\OrganizationController@index')->name('user');

    Route::get('/user/{id}',
'api\usercontroller@searchbyorganizationid')->name('user.search');
    Route::get('/ticket/{id}',
'api\ticketcontroller@searchbyorganizationid')->name('ticket.search');
    Route::get('/organization/{id}',
'api\organizationcontroller@searchbyorganizationid')->name('organization.search');

    Route::get('/organizations/{id}',
'api\organizationcontroller@search')->name('organization.search.ticket');

    Route::get('/users/{name}',
'api\usercontroller@search')->name('user.search.ticket');

```



```
Route::get('/tickets/{subject}',  
'api\ticketcontroller@search')->name('ticket.search.user');  
});
```

```
});
```

-----BEGIN RSA PRIVATE KEY-----

MIIJKAIBAAKCAgEApPIjDkByFFCuBB1Ka6ShpN/GM3N91M/zA78H2ej
SR17ZcwhG
svc+5b2pxahMkGlXjich1AVn0B3JhipUKirJFXT33c54UC731pTfiDM7YGg
cCis+
uOoWyA5/XIJbVfpHZkoBLcmgMY0joy9N1OXbHwjkkPEGZMHXyY8/aQx
h3igiGnpH
T7VBGo4V2DxcRz77J11IELunD4QZgrHHYWznHTXvCZNN9agzERmU
p9+u2VUajccx
lhOsC1ldgCVTqTKa5J7yEelxr3LoR9G0RrVbXz5JJVdl3tIL8CC6SYI3Jid
OT1IN
ls/3R9fHy9cosi4sGwtrEbEHyzZxGdlzBauYMLWYT4CssLsXCH8Nt7HX2
9jQ1UFV
vuLe7Le3dNHSb78u9ZE1z+rrekbnmxl3QBASc7LJt1iVp2mQVELBCc88
n6vg0Zs2
B8JN4Ykl35LI3oaSCmE1Cs6suYRW/eh1VMbe3uunkshm8IVBUGbmY7
3g7fek4ZXu
o8Rce/nVijz7qJvcac6RLekr/FPN1HZ6nRjB2TI9QtcJSJ377ZRazOEGZJz
gMzp8
IZDVmKIOL8edYeZ1UwhS5WvVv35Mc11qQfu8koJpfLULVW0wWazpyn
sxuFVvAl1+
0k1WzToqtAXBaE8k4DfB7WB3jG51VgosqqV85h8jLNz6Nlged1Vx3KAX
xVMCAwEA
AQKCAgA4BO8b8cpLF7Jfd0rEDp1NDSeSZQ8/anWMuhuCZUqyLYUN
SKwpLUfQX3z8
v09wYj3LFYKpKkS8KME7q7f0sZ6iiJt6WBgGskw6M0dMSolJDVEoKtMb
m3Gs8S9S

6ghVw6znEQ4hbCvXZ6PIT9HQD1U+zpfnqdCP+jq6GMhhdXDULRaXgx
of+N5olALa
8HQrtRTengFFHQ1FRNXwslw4fCQnZo2zOBZUda8YbcmJGNvB6oClqJ
0+xNHDhUbEs
SICGHcb3BvoQFIzmlNBHQvBCambiipXv9iODbDPdkF9xgqQTcplm3F4
6bdf8L0uK
7j7FcmRLsq7izQhmetAhBPPH4yLEK/RkOOWi3LhMgEwaxg2kDePp/EI
adGIAHTfB
/7CN9R/acN2M+QMz7k3g4yTSb+6wk/h8gCsal1Wi+nAaRPjVAvWSX4s
CGr2GOtmX
rzv/a9PIU2DXTa+zcTPNM3xIGZp1+GPpQVbNUV0grAeuJ1DecXYkOLX
gzNJmpJtX
4MPGN6WC5wUqq733li4d92NoASef09QqvS73XQmSIILvwpKvm918sn
Kik4uDlt9f
9ob5mrF145smSabcbH8aPNOHIV0iXJS6y3oux8WdOMaiM057ei6YhxD
LEVlovW/q
iqTQrr1UyOGwdD3ZW8JOFf+ax0seED318Z9Wos3eFzyMStL9sQKCAQ
EAzZIngveu
rr8C6vcXbHYVM+pboUAoMmr+1x5mvQv0wdAKk4hqLPNOCADSzWB5
4dMgT/bbQqCx
TyVy66ympra7p0g+5EsprO82gKEAcxVR/0lszwkeHwZOpZHAiy0Keqzcj
862E3SJ
2NqhNb+a2fEydVotk22T/CBt5oA8CH34Z21AdXSSfp4mpVltOeoHpElq6
ES8Gw5L
NWCzxAFci/4zn01eGTLLdNWJpA9xsN95mL+SwXOxvpJYjv9PM/49laa
BX4eiA/GC
Y5LjR5ShMrEbMMpwoKx0jUPa31oy2gfQ/LVrXzHr+hMvlyjCOt5LL5gHm
+jqjZEX
FQXUztDHvZrKnwKCAQEAzXHCJZR6jijX20PwDs3oeFAaJNWTHgQb0
S9BHgSOwatm
Bk4amCqvgcMnUo32JEubC6kYrc7r/38fdV4YDP5s0hDQ3evuDaTIVu5X
iK9F6/KU
JXvudsPUn5oHGU1nDs5TCO5PD4xXUhvJSi3ead89OBw1nzo0leGY44Z
OFrIJ1HOk
PvOuVzKf3b4vcza/gWeZ8JC2NJZIEQ23/GtHIIg4T9t7hKOJEWp7T3utw
UJ5C4eA

B1w/Jewcvogfff+t1Cua0GOk/ahCl612yE+VEnIPwzrd03lOlMrLn8Ja/t5X
WCn6
/3ZKiNd5ZavXmnzEFGhkvy2gDeJ8cKlpGSf1PH8zQKCAQB8leNGDHY
pAyKp66bz
UaKgBMZr6qfbOgzErZDwWwXi+tNIHzmuBwV5g7XHWph+qtR6NOBVr
V8klcLxWQIQ
7YsHCnMfu9tgl9pbdd0loRwwgY/4vNK5fT+EbyzLXdlUFonhNLtxqA+NK
QfrwtWp
ASKtBIU7DxlrTxHN/GVD431aRviDdG75klpXwlZYgF847zRIKQxFci4r/0b
N9szO
aCOX8EoAAVImLMO+k6zdVzS/7f6aXlhYcVqUjsqu5LjZVwgMPYSxyOY
pZotjr/Ze
+98laTj1bO28V9sUwAyw58NktdAuXaDXEzqFipkunwjbpnwBw3XPImvD
wA+vCxdy
AMSVaolBAQCSv6h7qYSIWR1fHTczi/d7QVR7bSeKMEsicAumgoDqbC
ETbeTnPs5g
9WJ4QLIKkcrWFstg/nlZgu1jYHGfIKC9soLS0Se8E6+cb+K/hAHCx6xa8
bQYe/W
FXLuqvBrjguX/InyxD93gj4ntX4CyBh1pQjUdQIs8ZmBJGVRLAdFv1jaMr
wcwyeY
gApq1PeUWsmGlghUpehSbic3lfYM2Ky8Nj4Tr7DwBMySKPkja/UEvER
Cfa0Jzl93
tcow+VpUjNUn95VaLz8XNwf664XDTW4hOUJMkFNEC1vKAU14cp+Xd
V0572gtSIFw
xuL7oNijInvAf81o5sMdMvmY5Nuag0QdAoIBAH5q0trabzrf13z+Chlqe00
wdzra
jFu1qSJvtLemYfkgHEjcbgQVH7JRg8Qkbb+uk0g1wdFfGFSjpgiMq7jsZi6
+KJfv
dVtCtHaaaCPudVsdbYH0qizmePUdQ1YxgPr14wY12jxpvs5GdKvnlopxZ
M19tZwa
fjECBlwJYMeeh+mvnbRvajF+To/Mxv69WrnYR+Uz7ynnuR8ve71yijdqIjB
Z0Mhx
tM6Fnisr7l1rSoiB/8gZFffZoekQkjhJW9Pyhz0PrePbFf4vD+BIK3AU7uizQ
ySj
1bzmWwz7lzeTXs0dRuA8JjAKctT320ledzLBIPyeO51Sh7W+YGQfzWK
WI7I=
-----END RSA PRIVATE KEY-----

-----BEGIN PUBLIC KEY-----

MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICGgKCAgEApPljDkByFFC
uBB1Ka6Sh
pN/GM3N91M/zA78H2ejSR17ZcwhGsvc+5b2pxahMkGlXjich1AVn0B3Jh
ipUKirJ
FXT33c54UC731pTfiDM7YGgcCis+uOoWyA5/XIJbVfpHZkoBLcmgMY0j
oy9N1OXb
HwjjkPEGZMHXyY8/aQxh3igiGnpHT7VBGo4V2DxcRz77J11IELunD4Q
ZgrHHYWzn
HTXvCZNN9agzERmUp9+u2VUajccxhOsC1ldgCVTqTKa5J7yEelxr3Lo
R9G0RrVb
Xz5JJVdl3tIL8CC6SYI3JidOT1INIs/3R9fHy9cosi4sGwtrEbEHyzZxGdlzB
auY
MLWYT4CsslXCH8Nt7HX29jQ1UFVvuLe7Le3dNHSb78u9ZE1z+rrekb
nmxl3QBAS
c7LJt1iVp2mQVELBCc88n6vg0Zs2B8JN4Ykl35LI3oaSCmE1Cs6suYR
W/eh1VMbe
3uunkshm8IVBUGbmY73g7fek4ZXuo8Rce/nVijz7qJvcac6RLekr/FPN1H
Z6nRjB
2TI9QtcJSJ377ZRazOEGZJzgMzp8IZDVmKIOL8edYeZ1UwhS5WvVv3
5Mc11qQfu8
koJpfLULVW0wWazpynsxuFVvAl1+0k1WzToqtAXBaE8k4DfB7WB3jG5
1VgosqqV8
5h8jLNz6NIged1Vx3KAXxVMCAwEAAQ==
-----END PUBLIC KEY-----

