

Routes

```
<?php

use CodeIgniter\Router\RouteCollection;

/**
 * @var RouteCollection $routes
 */
$routes->get('/', 'Home::index');
$routes->get('/products', 'ProductController::api_index');
$routes->get('/products/create_view', 'ProductController::create_view');
$routes->post('/products/create', 'ProductController::api_create');
$routes->get('/products/edit/(:num)', 'ProductController::api_edit/$1');
$routes->post('/products/update/(:num)', 'ProductController::api_update/$1');
$routes->get('/products/delete/(:num)', 'ProductController::api_delete/$1');

// Login routes
$routes->get('/login', 'LoginController::index');
$routes->post('/login', 'LoginController::authenticate');
$routes->get('/logout', 'LoginController::logout');
$routes->get('/signup', 'LoginController::signup');
$routes->post('/signup/register', 'LoginController::register');

// API Routes
$routes->group('api', function($routes) {
    $routes->get('products', 'api\APIProductController::index');
    $routes->post('products/create', 'api\APIProductController::create');
    $routes->get('products/getproduct/(:num)', 'api\APIProductController::getproduct/$1');
    $routes->put('products/update/(:num)', 'api\APIProductController::update/$1');
    $routes->delete('products/delete/(:num)', 'api\APIProductController::delete/$1');
});
```

ProductController

```
<?php

namespace App\Controllers;
class ProductController extends BaseController
{
    public function api_index()
    {
        $client = \Config\Services::curlrequest();
        $response = $client->request('GET', 'http://localhost/bsit3d-ci-4/public/api/products');
        $data = json_decode($response->getBody(), true);
        return view('product_view', $data);
    }
    // create view
    public function create_view()
    {
        return view('create_product');
    }
    public function api_create()
    {
        $data = [
            'product_name' => $this->request->getPost('product_name'),
            'description' => $this->request->getPost('description'),
            'price' => $this->request->getPost('price')
        ];
        /* use the api create method in apicontroller
         * http://localhost/bsit3d-ci-4/public/api/products/create
         * with POST method
         */
        $client = \Config\Services::curlrequest();
        $response = $client->request('POST', 'http://localhost/bsit3d-ci-4/public/api/products/create', [
            'form_params' => $data
        ]);
        if ($response->getStatusCode() == 201) {
            // Successfully created via API
            return redirect()->to('/products');
        } else {
            // Handle error
            throw new \Exception('Failed to create product via API');
        }
    }
    public function api_edit($id)
    {
        /* use the api index method in apicontroller
         * http://localhost/bsit3d-ci-4/public/api/products/getproduct
         * with GET method
         */
        $client = \Config\Services::curlrequest();
        $response = $client->request('GET', "http://localhost/bsit3d-ci-4/public/api/products/getproduct/{$id}");

        $data['product'] = json_decode($response->getBody(), true);
        return view('edit_product', $data);
    }
}
```

```

}

public function api_update($id)
{
    $data = [
        'product_name' => $this->request->getPost('product_name'),
        'description' => $this->request->getPost('description'),
        'price' => $this->request->getPost('price')
    ];
    /* use the api update method in apicontroller
     * http://localhost/bsit3d-ci-4/public/api/products/update/11
     * with PUT method
     */
    $client = \Config\Services::curlrequest();
    $response = $client->request('PUT', "http://localhost/bsit3d-ci-4/public/api/products/update/{$id}", [
        'form_params' => $data
    ]);
    if ($response->getStatusCode() == 200) {
        // Successfully updated via API
        return redirect()->to('/products');
    } else {
        // Handle error
        throw new \Exception('Failed to update product via API');
    }
}
public function api_delete($id)
{
    //dd('test api delete');
    /* use the api delete method in apicontroller
     * http://localhost/bsit3d-ci-4/public/api/products/delete/11
     * with DELETE method
     */
    $client = \Config\Services::curlrequest();
    $response = $client->request('DELETE', "http://localhost/bsit3d-ci-4/public/api/products/delete/{$id}");
    // dd($response->getBody());
    if ($response->getStatusCode() == 200) {
        // Successfully deleted via API
        return redirect()->to('/products');
    } else {
        // Handle error
        throw new \Exception('Failed to delete product via API');
    }
}
}

```

LoginController

```
<?php
namespace App\Controllers;

use CodeIgniter\Controller;
use App\Models\UserModel;

class LoginController extends Controller
{
    public function index()
    {
        return view('login');
    }

    public function authenticate()
    {
        $session = session();
        $model = new UserModel();

        $username = $this->request->getPost('username');
        $password = $this->request->getPost('password');

        $user = $model->where('username', $username)->first();

        if ($user && password_verify($password, $user['password'])) {
            $session->set([
                'user_id' => $user['id'],
                'username' => $user['username'],
                'logged_in' => true
            ]);
            return redirect()->to('/products');
        } else {
            $session->setflashdata('error', 'Invalid username or password');
            return redirect()->to('/login');
        }
    }

    public function logout()
    {
        session()->destroy();
        return redirect()->to('/login');
    }

    public function signup()
    {
        return view('signup');
    }

    public function register()
    {
        $session = session();
        $model = new UserModel();
```

```
$username = $this->request->getPost('username');
$password = $this->request->getPost('password');
$email = $this->request->getPost('email');
$confirmPassword = $this->request->getPost('confirm_password');

if ($password !== $confirmPassword) {
    $session->setFlashdata('error', 'Passwords do not match');
    return redirect()->to('/signup');
}

if ($model->where('username', $username)->first()) {
    $session->setFlashdata('error', 'Username already exists');
    return redirect()->to('/signup');
}

$data = [
    'username' => $username,
    'email'=> $email,
    'password' => password_hash($password, PASSWORD_DEFAULT)
];

$model->insert($data);

$session->setFlashdata('success', 'Registration successful. Please login.');
return redirect()->to('/login');
}
```

APIProductController

```
<?php

namespace App\Controllers\Api;

use App\Controllers\BaseController;
use App\Models\ProductModel;
use CodeIgniter\API\ResponseTrait;
use CodeIgniter\HTTP\IncomingRequest;

class APIProductController extends BaseController
{
    use ResponseTrait;
    protected $request;

    public function __construct()
    {
        $this->request = \Config\Services::request();
    }

    public function index()
    {
        $model = new ProductModel();
        $data['products'] = $model->findAll();
        return $this->respond($data);
    }

    public function getproduct($id = null)
    {
        $model = new ProductModel();
        $data = $model->find($id);
        if ($data) {
            return $this->respond($data);
        } else {
            return $this->failNotFound('Product not found');
        }
    }

    public function create()
    {
        $model = new ProductModel();
        // Get the input data
        $data = [
            'product_name' => $this->request->getPost('product_name'),
            'description' => $this->request->getPost('description'),
            'price' => $this->request->getPost('price')
        ];
        $id = $model->createProduct($data);
        if ($id) {
            $response = [
                'status' => 201,
                'error' => null,
                'messages' => [
                    'success' => 'Product created successfully'
                ]
            ];
            return $this->respond($response);
        } else {
            return $this->failServerError('Failed to create product');
        }
    }
}
```

```

        ]
    ];
    return $this->respondCreated($response);
} else {
    return $this->fail($model->errors());
}
}
public function update($id = null)
{
    $model = new ProductModel();
    $data = $this->request->getRawInput();
    $updated = $model->update($id, $data);
    if ($updated) {
        $response = [
            'status' => 200,
            'error' => null,
            'messages' => [
                'success' => 'Product updated successfully'
            ]
        ];
        return $this->respond($response);
    } else {
        return $this->fail($model->errors());
    }
}
public function delete($id = null)
{
    $model = new ProductModel();
    $data = $model->find($id);
    if ($data) {
        $model->delete($id);
        $response = [
            'status' => 200,
            'error' => null,
            'messages' => [
                'success' => 'Product deleted successfully'
            ]
        ];
        return $this->respondDeleted($response);
    } else {
        return $this->failNotFound('Product not found');
    }
}
}

```

ProductModel

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class ProductModel extends Model
{
    protected $table = 'products';
    protected $primaryKey = 'id';
    protected $allowedFields = ['product_name', 'description', 'price'];
    // Create
    public function createProduct($data)
    {
        return $this->insert($data);
    }
    // Read
    public function getProduct($id)
    {
        return $this->find($id);
    }
    // Update
    public function updateProduct($id, $data)
    {
        return $this->update($id, $data);
    }
    // Delete
    public function deleteProduct($id)
    {
        return $this->delete($id);
    }
}
```

UserModel

```
<?php

namespace App\Models;
use CodeIgniter\Model;

class UserModel extends Model
{
    protected $table    = 'users';
    protected $primaryKey = 'id';

    protected $allowedFields = [
        'username',
        'email',
        'password',
        'created_at',
        'updated_at'
    ];

    protected $useTimestamps = true;
    protected $createdField = 'created_at';
    protected $updatedField = 'updated_at';

    protected $returnType  = 'array';
    protected $useSoftDeletes = false;

    // Add custom methods if needed
}
```