



江汉大学授权认证中心

源码文档

2024-8-20

```

1  import React, { useState } from 'react';
2  import { Form, Input, Button, Alert, Typography } from 'antd';
3  import { useNavigate } from 'react-router-dom';
4  import api from '../api/api';
5  import 'antd/dist/reset.css'; // 导入 Ant Design 样式
6
7  const { Title, Link } = Typography;
8
9  const Login: React.FC = () => {
10     const [message, setMessage] = useState<string>("");
11     const navigate = useNavigate();
12
13     const handleSubmit = async (values: { email: string; password: string }) => {
14         try {
15             const response = await api.post('/user/login', {
16                 email: values.email,
17                 password: values.password,
18             });
19             const token = response.data.token;
20             localStorage.setItem('token', token);
21
22             setMessage('登录成功');
23             navigate('/home'); // 跳转到主页或其他你想跳转的页面
24
25         } catch (error) {
26             setMessage('登录失败');
27         }
28     };
29
30     return (
31         <div style={{ maxWidth: 400, margin: 'auto', padding: '20px' }}>
32             <Title level={2} style={{ textAlign: 'center' }}>登录</Title>
33             <Form
34                 name="login"
35                 initialValues={{ remember: true }}
36                 onFinish={handleSubmit}
37                 style={{ padding: '20px', background: '#fff', borderRadius: '8px', boxShadow: '0 4px 8px
38 rgba(0,0,0,0.1)' }}
39             >
40                 <Form.Item
41                     name="email"
42                     rules={[{ required: true, message: '请输入您的邮箱!' }] }
43                 >
44                     <Input placeholder="邮箱" />
45                 </Form.Item>
46
47                 <Form.Item
48                     name="password"
49                     rules={[{ required: true, message: '请输入您的密码!' }] }
50                 >

```

江汉大学授权认证中心

```
1      <Input.Password placeholder="密码" />
2    </Form.Item>
3
4    <Form.Item>
5      <Button type="primary" htmlType="submit" block>
6        登录
7      </Button>
8    </Form.Item>
9
10   {message && (
11     <Form.Item>
12       <Alert message={message} type={message === '登录成功' ? 'success' : 'error'}
13 showicon />
14     </Form.Item>
15   )}
16
17   <Form.Item>
18     <Link
19 href="http://localhost:8080/oauth/authorize?client_id=9cb0c3b1-b5f3-475a-8a0b-17934bee2d99&r
20 edirect_uri=http://localhost:3000/home&response_type=code&scope=*">
21       第三方登录
22     </Link>
23   </Form.Item>
24 </Form>
25 </div>
26 );
27 };
28
29 export default Login;
30 import React, { useEffect, useState } from 'react';
31 import { useNavigate } from 'react-router-dom';
32 import { Button, Table, Spin, Typography, Alert } from 'antd';
33 import api from '../api/api'; // 确保 api 配置正确
34
35 const { Title } = Typography;
36
37 interface User {
38   id: number;
39   name: string;
40   email: string;
41   email_verified_at: string | null;
42   created_at: string;
43   updated_at: string;
44 }
45
46 const User: React.FC = () => {
47   const [users, setUsers] = useState<User[]>([]);
48   const [loading, setLoading] = useState(true);
49   const [error, setError] = useState<string | null>(null);
50   const navigate = useNavigate();
```

```

1
2    useEffect(() => {
3      const fetchUsers = async () => {
4        try {
5          const response = await api.post('/user/list'); // 确保 API 路径正确
6          setUsers(response.data.data);
7        } catch (err) {
8          setError('Failed to fetch users');
9        } finally {
10         setLoading(false);
11       }
12     };
13
14     fetchUsers();
15   }, []);
16
17   const columns = [
18     {
19       title: 'ID',
20       dataIndex: 'id',
21       key: 'id',
22     },
23     {
24       title: '名称',
25       dataIndex: 'name',
26       key: 'name',
27     },
28     {
29       title: '电子邮件',
30       dataIndex: 'email',
31       key: 'email',
32     },
33     {
34       title: '创建时间',
35       dataIndex: 'created_at',
36       key: 'created_at',
37       render: (text: string) => new Date(text).toLocaleString(),
38     },
39     {
40       title: '更新时间',
41       dataIndex: 'updated_at',
42       key: 'updated_at',
43       render: (text: string) => new Date(text).toLocaleString(),
44     },
45   ];
46
47   if (loading) return <Spin size="large" />;
48   if (error) return <Alert message={error} type="error" showIcon />;
49
50   return (

```

```

1      <div style={{ padding: 20 }}>
2          <Title level={1}>用户列表</Title>
3          <div style={{ marginBottom: 20 }}>
4              <Button type="primary" onClick={() => navigate('/home')}>返回首页</Button>
5          </div>
6          <Table
7              columns={columns}
8              dataSource={users}
9              rowKey="id"
10             pagination={{ pageSize: 10 }}
11          />
12      </div>
13  );
14  };
15
16  export default User;
17
18  // src/pages/Register.tsx
19  import React, { useState } from 'react';
20  import axios from 'axios';
21  import api from '../api/api';
22
23  const Register: React.FC = () => {
24      const [username, setName] = useState("");
25      const [email, setEmail] = useState("");
26      const [password, setPassword] = useState("");
27      // const [passwordConfirmation, setPasswordConfirmation] = useState("");
28      const [message, setMessage] = useState("");
29
30      const handleSubmit = async (e: React.FormEvent) => {
31          e.preventDefault();
32          try {
33              const response = await api.post('/signup', {
34                  username,
35                  email,
36                  password,
37                  // password_confirmation: passwordConfirmation,
38              });
39              setMessage(response.data.message);
40          } catch (error) {
41              console.error('Error registering:', error);
42              setMessage('Registration failed!');
43          }
44      };
45
46      return (
47          <div>
48              <h2>注册</h2>
49              <form onSubmit={handleSubmit}>
50                  <div>

```

```

1      <label>用户名:</label>
2      <input type="text" value={username} onChange={(e) => setName(e.target.value)}
3  required />
4      </div>
5      <div>
6          <label>邮箱:</label>
7          <input type="email" value={email} onChange={(e) => setEmail(e.target.value)}
8  required />
9      </div>
10     <div>
11         <label>密码:</label>
12         <input type="password" value={password} onChange={(e) =>
13 setPassword(e.target.value)} required />
14     </div>
15     /* <div>
16         <label>确认 密码:</label>
17         <input type="password" value={passwordConfirmation} onChange={(e) =>
18 setPasswordConfirmation(e.target.value)} required />
19     </div> */
20     <button type="submit">注册</button>
21 </form>
22 {message && <p>{message}</p>}
23 </div>
24 );
25 };
26
27 export default Register;
28
29 import React, { useEffect, useState } from 'react';
30 import { Button, Form, Input, Modal, Table, Typography, Spin, Alert } from 'antd';
31 import api from '../api/api';
32
33 const { Title } = Typography;
34
35 interface Client {
36     id: string;
37     user_id: string | null;
38     name: string;
39     provider: string | null;
40     redirect: string;
41     personal_access_client: boolean;
42     password_client: boolean;
43     revoked: boolean;
44     created_at: string;
45     updated_at: string;
46 }
47
48 const Home: React.FC = () => {
49     const [clients, setClients] = useState<Client[]>([]);
50     const [loading, setLoading] = useState(true);

```

```

1    const [error, setError] = useState<string | null>(null);
2
3    const [showForm, setShowForm] = useState(false);
4    const [newClient, setNewClient] = useState({ name: "", redirect: "" });
5
6    useEffect(() => {
7        const fetchClients = async () => {
8            try {
9                const response = await api.post('/client/list');
10               setClients(response.data.data);
11            } catch (err) {
12                setError('Failed to fetch clients');
13            } finally {
14                setLoading(false);
15            }
16        };
17
18        fetchClients();
19    }, []);
20
21    const handleAddClient = async () => {
22        try {
23            const response = await api.post('/client/create', {
24                name: newClient.name,
25                redirect: newClient.redirect,
26            });
27            setClients([...clients, response.data]);
28            setShowForm(false);
29            setNewClient({ name: "", redirect: "" });
30        } catch (err) {
31            setError('添加客户端失败');
32        }
33    };
34
35    const handleDeleteClient = async (id: string) => {
36        try {
37            await api.post('/client/delete', { id: [id] });
38            setClients(clients.filter(client => client.id !== id));
39        } catch (err) {
40            setError('Failed to delete client');
41        }
42    };
43
44    const handleInputChange = (e: React.ChangeEvent<HTMLInputElement>) => {
45        const { name, value } = e.target;
46        setNewClient(prevState => ({
47            ...prevState,
48            [name]: value,
49        }));
50    };

```

```

1
2     const columns = [
3         { title: 'ID', dataIndex: 'id', key: 'id' },
4         { title: '客户端', dataIndex: 'name', key: 'name' },
5         { title: '重定向地址', dataIndex: 'redirect', key: 'redirect' },
6         { title: 'Personal Access Client', dataIndex: 'personal_access_client', key:
7 'personal_access_client', render: (text: boolean) => (text ? 'Yes' : 'No') },
8         { title: 'Password Client', dataIndex: 'password_client', key: 'password_client', render: (text:
9 boolean) => (text ? 'Yes' : 'No') },
10        { title: 'Revoked', dataIndex: 'revoked', key: 'revoked', render: (text: boolean) => (text ? 'Yes' :
11 'No') },
12        { title: '创建时间', dataIndex: 'created_at', key: 'created_at', render: (text: string) => new
13 Date(text).toLocaleString() },
14        { title: '更新时间', dataIndex: 'updated_at', key: 'updated_at', render: (text: string) => new
15 Date(text).toLocaleString() },
16        {
17            title: '操作',
18            key: 'action',
19            render: (_, record: Client) => (
20                <Button type="link" danger onClick={() => handleDeleteClient(record.id)}>
21                    删除
22                </Button>
23            ),
24        },
25    ];
26
27    if (loading) return <Spin size="large" />;
28    if (error) return <Alert message={error} type="error" showIcon />;
29
30    return (
31        <div style={{ padding: 20 }}>
32            <div style={{ marginBottom: 20 }}>
33                <Button type="primary" onClick={() => window.location.href = '/apply'}
34 style={{ marginRight: 10 }}>申请列表</Button>
35                <Button type="primary" onClick={() => window.location.href = '/user'}>用户管理
36            </Button>
37            </div>
38
39            <Title level={1}>客户端列表</Title>
40            <Button type="primary" onClick={() => setShowForm(true)} style={{ marginBottom: 20 }}>
41 添加客户端</Button>
42
43            <Table columns={columns} dataSource={clients} rowKey="id" />
44
45            <Modal
46                title="添加客户端"
47                visible={showForm}
48                onOk={handleAddClient}
49                onCancel={() => setShowForm(false)}
50                okText="提交"

```



```

1      cancelText="取消"
2      >
3      <Form layout="vertical" onFinish={handleAddClient}>
4          <Form.Item
5              label="名称"
6              name="name"
7              rules={[{ required: true, message: '请输入客户端名称!' }]}
8          >
9              <Input
10                 value={newClient.name}
11                 onChange={handleInputChange}
12                 name="name"
13             />
14         </Form.Item>
15         <Form.Item
16             label="重定向 URL"
17             name="redirect"
18             rules={[{ required: true, message: '请输入重定向 URL!' }]}
19         >
20             <Input
21                 type="url"
22                 value={newClient.redirect}
23                 onChange={handleInputChange}
24                 name="redirect"
25             />
26         </Form.Item>
27     </Form>
28 </Modal>
29 </div>
30 );
31 };
32
33 export default Home;
34
35 import React, { useEffect, useState } from 'react';
36 import axios from 'axios';
37 import { useNavigate } from 'react-router-dom';
38 import { Button, Table, Select, Typography, Spin, Alert } from 'antd';
39 import { ColumnsType } from 'antd/es/table';
40 import './Apply.css';
41
42 const { Title } = Typography;
43 const { Option } = Select;
44
45 interface ClientApplication {
46     id: number;
47     applicant: string;
48     phone_number: string;
49     client_name: string;
50     callback_url: string;

```

```

1      status: number;
2      created_at: string;
3  }
4
5  const Apply: React.FC = () => {
6      const [applications, setApplications] = useState<ClientApplication[]>([]);
7      const [loading, setLoading] = useState(true);
8      const [error, setError] = useState<string | null>(null);
9      const navigate = useNavigate();
10
11     useEffect(() => {
12         axios.post('http://localhost:8080/api/applylist')
13             .then(response => {
14                 setApplications(response.data);
15             })
16             .catch(error => {
17                 setError('Error fetching data');
18             })
19             .finally(() => {
20                 setLoading(false);
21             });
22     }, []);
23
24     const handleStatusChange = (id: number, newStatus: number) => {
25         axios.post('http://localhost:8080/api/applyupdate', {
26             id: id,
27             status: newStatus
28         })
29         .then(() => {
30             setApplications(prevApplications =>
31                 prevApplications.map(app =>
32                     app.id === id ? { ...app, status: newStatus } : app
33                 )
34             );
35         })
36         .catch(error => {
37             setError('Error updating status');
38         });
39     };
40
41     const columns: ColumnsType<ClientApplication> = [
42         { title: 'ID', dataIndex: 'id', key: 'id' },
43         { title: '申请人', dataIndex: 'applicant', key: 'applicant' },
44         { title: '电话号码', dataIndex: 'phone_number', key: 'phone_number' },
45         { title: '客户端名称', dataIndex: 'client_name', key: 'client_name' },
46         {
47             title: '回调地址',
48             dataIndex: 'callback_url',
49             key: 'callback_url',
50             render: (text: string) => (

```

```

1      <a href={text} target="_blank" rel="noopener noreferrer">{text}</a>
2    )
3  },
4  {
5    title: '申请时间',
6    dataIndex: 'created_at',
7    key: 'created_at',
8    render: (text: string) => new Date(text).toLocaleString()
9  },
10 {
11   title: '状态',
12   dataIndex: 'status',
13   key: 'status',
14   render: (status: number, record: ClientApplication) => (
15     <Select
16       value={status}
17       onChange={(value) => handleStatusChange(record.id, value)}
18       style={{ width: 120 }}
19     >
20       <Option value={1}>待审批</Option>
21       <Option value={2}>已通过</Option>
22       <Option value={3}>未通过</Option>
23     </Select>
24   )
25 },
26 ];
27
28 if (loading) return <Spin size="large" />;
29 if (error) return <Alert message={error} type="error" showIcon />;
30
31 return (
32   <div style={{ padding: 20 }}>
33     <Title level={1}>申请列表</Title>
34     <Button type="primary" onClick={() => navigate('/home')} style={{ marginBottom: 20 }}>
35       返回首页
36     </Button>
37     <Table
38       columns={columns}
39       dataSource={applications}
40       rowKey="id"
41       pagination={{ pageSize: 10 }}
42     />
43   </div>
44 );
45 };
46
47 export default Apply;
48
49 <?php
50

```

```

1 namespace App\Http\Controllers\Auth;
2
3 use Illuminate\Http\Request;
4 use App\Http\Controllers\Controller;
5 use App\Models\Client;
6 use Laravel\Passport\ClientRepository;
7 use Illuminate\Support\Facades\Auth;
8
9 class ClientController extends Controller
10 {
11     public function list(Request $request)
12     {
13         $request->validate([
14             'current' => 'integer|nullable',
15             'pageSize' => 'integer|nullable',
16         ]);
17
18         // 获取当前页码，默认为 1
19         $current = $request->input('current', 1);
20
21         // 获取每页显示的记录数，默认为 10
22         $pageSize = $request->input('pageSize', 10);
23
24         // 排除特定的 id
25         $excludedIds = [
26             '9cab4d79-1dff-4bd2-ab5b-3ec6bc05b2fa',
27             '9cab4d79-188a-4959-80e3-e532be230b26'
28         ];
29
30         // 使用分页查询并排除特定的 id,
31         $clients = Client::whereNotIn('id', $excludedIds)
32             ->where('revoked', '!=', 1)
33             ->paginate($pageSize, ['*'], 'page', $current);
34
35         // 返回用户列表和分页信息
36         return response()->json([
37             'data' => $clients->items(),
38             'total' => $clients->total(),
39             'current' => $clients->currentPage(),
40             'pageSize' => $clients->perPage(),
41         ]);
42     }
43     // 创建一个新的客户端
44     public function create(Request $request)
45     {
46         // 验证字段
47         $request->validate([
48             'name' => 'required',
49             'redirect' => 'required|url',
50         ]);

```

```

1      $clientRepository = app(ClientRepository::class);
2      $client = $clientRepository->create(
3          $request->name, $request->redirect
4      );
5      // exit;
6      return response()->json($client);
7  }
8
9  // 更新客户端
10 public function update(Request $request)
11 {
12     // 验证字段
13     $request->validate([
14         'id' => 'required|uuid', // 确保 id 是 UUID 格式
15         'name' => 'required',
16         'redirect' => 'required|url',
17     ]);
18
19     // 获取 Client 实例
20     $clientRepository = app(ClientRepository::class);
21     $client = $clientRepository->find($request->id);
22
23     if (!$client) {
24         return response()->json(['error' => 'Client not found'], 404);
25     }
26
27     // 更新 Client
28     $updatedClient = $clientRepository->update(
29         $client, $request->name, $request->redirect
30     );
31
32     return response()->json($updatedClient);
33 }
34
35 // 删除客户端
36 public function delete(Request $request)
37 {
38     $request->validate([
39         'id' => 'required|array',
40         'id.*' => 'uuid',
41     ]);
42
43     $clientRepository = app(ClientRepository::class);
44     foreach ($request->id as $id) {
45         $client = $clientRepository->find($id);
46         if ($client) {
47             $clientRepository->delete($client);
48         }
49     }
50

```

```

1      return response()->json(['msg' => '删除成功']);
2  }
3
4
5  }
6
7  <?php
8
9  namespace App\Http\Controllers\Auth;
10
11  use App\Http\Controllers\Controller;
12  use Illuminate\Foundation\Auth\ConfirmsPasswords;
13
14  class ConfirmPasswordController extends Controller
15  {
16      /*
17       |-----
18       | Confirm Password Controller
19       |-----
20       |
21       | This controller is responsible for handling password confirmations and
22       | uses a simple trait to include the behavior. You're free to explore
23       | this trait and override any functions that require customization.
24       |
25       */
26
27      use ConfirmsPasswords;
28
29      /**
30       * Where to redirect users when the intended url fails.
31       *
32       * @var string
33       */
34      protected $redirectTo = '/home';
35
36      /**
37       * Create a new controller instance.
38       *
39       * @return void
40       */
41      public function __construct()
42      {
43          $this->middleware('auth');
44      }
45  }
46
47  <?php
48
49  namespace App\Http\Controllers\Auth;
50

```

```

1  use App\Http\Controllers\Controller;
2  use Illuminate\Foundation\Auth\AuthenticatesUsers;
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Session;
5
6  class LoginController extends Controller
7  {
8      /*
9      |-----
10     | Login Controller
11     |-----
12     |
13     | This controller handles authenticating users for the application and
14     | redirecting them to your home screen. The controller uses a trait
15     | to conveniently provide its functionality to your applications.
16     |
17     */
18
19     use AuthenticatesUsers;
20
21     /**
22      * Where to redirect users after login.
23      *
24      * @var string
25      */
26     protected $redirectTo = '/home';
27
28     /**
29      * Create a new controller instance.
30      *
31      * @return void
32      */
33     public function __construct()
34     {
35         $this->middleware('guest')->except('logout');
36         $this->middleware('auth')->only('logout');
37     }
38
39     protected function credentials(Request $request)
40     {
41         // 验证码验证
42         $request->validate([
43             'captcha' => 'required|numeric',
44         ]);
45
46         // 验证码检查
47         $captcha = $request->input('captcha');
48         if ($captcha != Session::get('captcha')) {
49             return back()->withErrors(['captcha' => '验证码错误']);
50         }

```

```

1
2     // 清除验证码
3     Session::forget('captcha');
4
5     return $request->only('email', 'password');
6 }
7 }
8
9 <?php
10
11 namespace App\Http\Controllers\Auth;
12
13 use App\Http\Controllers\Controller;
14 use App\Models\User;
15 use Illuminate\Foundation\Auth\RegistersUsers;
16 use Illuminate\Support\Facades\Hash;
17 use Illuminate\Support\Facades\Validator;
18
19 class RegisterController extends Controller
20 {
21     /*
22     |-----
23     | Register Controller
24     |-----
25     |
26     | This controller handles the registration of new users as well as their
27     | validation and creation. By default this controller uses a trait to
28     | provide this functionality without requiring any additional code.
29     |
30     */
31
32     use RegistersUsers;
33
34     /**
35      * Where to redirect users after registration.
36      *
37      * @var string
38      */
39     protected $redirectTo = '/home';
40
41     /**
42      * Create a new controller instance.
43      *
44      * @return void
45      */
46     public function __construct()
47     {
48         $this->middleware('guest');
49     }
50

```



```

1      /**
2      * Get a validator for an incoming registration request.
3      *
4      * @param array $data
5      * @return \Illuminate\Contracts\Validation\Validator
6      */
7      protected function validator(array $data)
8      {
9          return Validator::make($data, [
10             'name' => ['required', 'string', 'max:255'],
11             'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
12             'password' => ['required', 'string', 'min:8', 'confirmed'],
13         ]);
14     }
15
16     /**
17     * Create a new user instance after a valid registration.
18     *
19     * @param array $data
20     * @return \App\Models\User
21     */
22     protected function create(array $data)
23     {
24         return User::create([
25             'name' => $data['name'],
26             'email' => $data['email'],
27             'password' => Hash::make($data['password']),
28         ]);
29     }
30 }
31
32 <?php
33
34 namespace App\Http\Controllers\User;
35 use Illuminate\Http\Request;
36 use App\Models\User;
37 use App\Http\Controllers\Controller;
38 use Illuminate\Support\Facades\Hash;
39 use Illuminate\Support\Str;
40
41 class UserController extends Controller
42 {
43     public function signup(Request $request){
44         $request->validate([
45             'name'=>'required',
46             'email'=>'required',
47             'password'=>'required'
48         ]);
49         $name=$request->input('name');
50         $email=$request->input('email');

```

```

1      $password=$request->input('password');
2      $check=User::where('email',$email)->first();
3      if($check){
4          return response()->json(['msg'=>'此邮箱已被占用!']);
5      }
6      $user = User::create(['name' => $name, 'email' => $email, 'password' => $password]);
7      if($user){
8          return response()->json(['msg'=>'注册成功']);
9      }else{
10         return response()->json(['msg'=>'注册失败']);
11     }
12
13 }
14
15 public function login(Request $request) {
16     // 验证请求数据
17     $request->validate([
18         'email' => 'required',
19         'password' => 'required',
20     ]);
21
22     // 查找用户
23     $user = User::where('email', $request->email)->first();
24
25     if ($user && Hash::check($request->password, $user->password)) {
26         $token = $user->createToken('personal token');
27         return response()->json(['msg' => '登录成功', 'data' => $token]);
28     } else {
29         // 登录失败
30         return response()->json(['msg' => '登录失败', 401]);
31     }
32 }
33
34 public function info(Request $request){
35     $userinfo=$request->user('api');
36     return response()->json(['data'=>$userinfo]);
37 }
38
39 public function logout(Request $request){
40     $user = $request->user('api');
41     if ($user) {
42         $user->token()->revoke();
43         return response()->json(['msg' => '成功退出登录']);
44     } else {
45         return response()->json(['msg' => '用户未认证', 401]);
46     }
47 }
48
49 public function list(Request $request)
50 {

```

```

1      $request->validate([
2          'current' => 'integer|nullable',
3          'pageSize' => 'integer|nullable',
4      ]);
5
6      $current = $request->input('current', 1);
7      $pageSize = $request->input('pageSize', 10);
8
9      // 查询用户列表
10     $users = User::paginate($pageSize, ['*'], 'page', $current);
11
12     // 返回用户列表和分页信息
13     return response()->json([
14         'data' => $users->items(),
15         'current' => $users->currentPage(),
16         'pageSize' => $users->perPage(),
17         'total' => $users->total(),
18     ]);
19
20 }
21
22 public function update(Request $request)
23 {
24     $request->validate([
25         'id' => 'required',
26         'name' => 'string',
27         'email' => 'email',
28         'password' => 'string',
29     ]);
30     $user = User::find($request->id);
31     $user->name = $request->name;
32     $user->email = $request->email;
33     $user->password = Hash::make($request->password);
34     $user->save();
35     return response()->json(['msg' => '更新成功']);
36 }
37
38 public function delete(Request $request)
39 {
40     // 数据为 id 列表
41     $request->validate([
42         'id' => 'required|array',
43     ]);
44     $ids = $request->input('id');
45     User::destroy($ids);
46     return response()->json(['msg' => '删除成功']);
47 }
48
49
50 }
```

```

1
2  <?php
3
4  namespace App\Http\Controllers;
5
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Session;
8
9  class CaptchaController extends Controller
10 {
11     public function show()
12     {
13         $width = 120;
14         $height = 40;
15
16         // 创建图像资源
17         $image = imagecreatetruecolor($width, $height);
18
19         // 定义颜色
20         $background = imagecolorallocate($image, 255, 255, 255); // 背景色
21         $textColor = imagecolorallocate($image, 0, 0, 0); // 文字颜色
22         $lineColor = imagecolorallocate($image, 150, 150, 150); // 干扰线颜色
23         $noiseColor = imagecolorallocate($image, 200, 200, 200); // 噪点颜色
24
25         // 填充背景
26         imagefilledrectangle($image, 0, 0, $width, $height, $background);
27
28         // 添加噪点
29         for ($i = 0; $i < 50; $i++) {
30             imagefilledrectangle($image, rand(0, $width), rand(0, $height), rand(0, $width), rand(0,
31 $height), $noiseColor);
32         }
33
34         // 添加干扰线
35         for ($i = 0; $i < 5; $i++) {
36             imageline($image, rand(0, $width), rand(0, $height), rand(0, $width), rand(0, $height),
37 $lineColor);
38         }
39
40         // 生成验证码文本
41         $captchaText = rand(1000, 9999);
42         Session::put('captcha', $captchaText);
43
44         // 使用内置字体绘制文本
45         imagestring($image, 5, 20, 10, $captchaText, $textColor);
46
47         // 设置响应头并输出图像
48         header('Content-Type: image/png');
49         imagepng($image);
50         imagedestroy($image);

```

```

1      }
2  }
3
4  <?php
5
6  namespace App\Http\Controllers;
7
8  use App\Models\ClientApplication;
9  use Illuminate\Http\Request;
10 use App\Models\Client;
11 use Laravel\Passport\ClientRepository;
12
13 class ClientApplicationController extends Controller
14 {
15     public function index()
16     {
17         $applications = ClientApplication::all();
18         return response()->json($applications);
19     }
20
21     public function store(Request $request)
22     {
23
24         $request->validate([
25             'applicant' => 'required|string|max:255',
26             'phone_number' => 'required|string|max:20',
27             'client_name' => 'required|string|max:255',
28             'callback_url' => 'required|string|max:2048',
29             // 'status' => 'required|integer'
30         ]);
31
32         $application = ClientApplication::create([
33             'applicant' => $request->applicant,
34             'phone_number' => $request->phone_number,
35             'client_name' => $request->client_name,
36             'callback_url' => $request->callback_url,
37             'status' => $request->status,
38         ]);
39
40         return response()->json($application, 201);
41     }
42
43     public function update(Request $request)
44     {
45         $request->validate([
46             'id' => 'required|integer|exists:client_applications,id',
47             'status' => 'required|integer|in:1,2,3'
48         ]);
49
50         $application = ClientApplication::findOrFail($request->id);

```

```

1      $application->status = $request->status;
2      $application->save();
3      if ($request->status === 2) {
4          $clientRepository = app(ClientRepository::class);
5          $clientRepository->create(
6              $application->client_name,
7              $application->callback_url
8          );
9      }
10
11
12      return response()->json($application);
13  }
14
15 }
16
17 <?php
18
19 namespace App\Models;
20
21 use Laravel\Passport\Client as BaseClient;
22 use Illuminate\Database\Eloquent\Factories\HasFactory;
23
24 class Client extends BaseClient
25 {
26
27     protected $table = 'oauth_clients';
28     use HasFactory;
29
30     protected $fillable = [
31         'id',
32         'user_id',
33         'name',
34         'secret',
35     ];
36
37     protected function casts(): array
38     {
39         return [
40             'id' => 'string',
41         ];
42     }
43
44     /**
45      * 确定客户端是否应该跳过授权提示。
46      */
47     public function skipsAuthorization(): bool
48     {
49         return $this->firstParty();
50     }

```

```

1  }
2  <?php
3
4  namespace App\Models;
5
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Database\Eloquent\Model;
8
9  class ClientApplication extends Model
10 {
11     use HasFactory;
12
13     protected $fillable = [
14         'applicant',
15         'phone_number',
16         'client_name',
17         'callback_url',
18         'status',
19     ];
20
21     public $timestamps = false; // 禁用时间戳
22 }
23
24 <?php
25
26 namespace App\Models;
27
28 // use Illuminate\Contracts\Auth\MustVerifyEmail;
29 use Illuminate\Database\Eloquent\Factories\HasFactory;
30 use Illuminate\Foundation\Auth\User as Authenticatable;
31 use Illuminate\Notifications\Notifiable;
32 use Laravel\Passport\HasApiTokens;
33
34 class User extends Authenticatable
35 {
36     use HasFactory, Notifiable, HasApiTokens;
37
38     /**
39      * The attributes that are mass assignable.
40      *
41      * @var array<int, string>
42      */
43     protected $fillable = [
44         'name',
45         'email',
46         'password',
47     ];
48
49     /**
50      * The attributes that should be hidden for serialization.

```

```

1      *
2      * @var array<int, string>
3      */
4      protected $hidden = [
5          'password',
6          'remember_token',
7      ];
8
9      /**
10     * Get the attributes that should be cast.
11     *
12     * @return array<string, string>
13     */
14     protected function casts(): array
15     {
16         return [
17             'email_verified_at' => 'datetime',
18             'password' => 'hashed',
19         ];
20     }
21 }
22
23 <?php
24
25 namespace App\Providers;
26
27 use Illuminate\Support\ServiceProvider;
28 use Laravel\Passport\Passport;
29
30 class AppServiceProvider extends ServiceProvider
31 {
32     /**
33      * Register any application services.
34      */
35     public function register(): void
36     {
37         //
38     }
39
40     /**
41      * Bootstrap any application services.
42      */
43     public function boot(): void
44     {
45         //
46         // $this->registerPolicies();
47         // Passport::routes();
48         Passport::tokensExpireIn(now()->addDays(15));
49         Passport::refreshTokensExpireIn(now()->addDays(30));
50         Passport::personalAccessTokensExpireIn(now()->addMonths(6));

```



```

1
2     }
3 }
4
5 @extends('layouts.app')
6
7 @section('content')
8 <div class="container">
9     <div class="row justify-content-center">
10         <div class="col-md-8">
11             <div class="card">
12                 <div class="card-header">{{ __('登录') }}</div>
13
14                 <div class="card-body">
15                     <form method="POST" action="{{ route('login') }}">
16                         @csrf
17
18                         <div class="row mb-3">
19                             <label for="email" class="col-md-4 col-form-label text-md-end">{{ __('邮箱
20 ') }}</label>
21
22                             <div class="col-md-6">
23                                 <input id="email" type="email" class="form-control @error('email')
24 is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email"
25 autofocus>
26
27                                 @error('email')
28                                     <span class="invalid-feedback" role="alert">
29                                         <strong>{{ $message }}</strong>
30                                     </span>
31                                 @enderror
32                             </div>
33                         </div>
34
35                         <div class="row mb-3">
36                             <label for="password" class="col-md-4 col-form-label text-md-end">{{ __('
37 密码') }}</label>
38
39                             <div class="col-md-6">
40                                 <input id="password" type="password" class="form-control
41 @error('password') is-invalid @enderror" name="password" required
42 autocomplete="current-password">
43
44                                 @error('password')
45                                     <span class="invalid-feedback" role="alert">
46                                         <strong>{{ $message }}</strong>
47                                     </span>
48                                 @enderror
49                             </div>
50                         </div>

```

```

1
2         <div class="row mb-3">
3             <label for="captcha" class="col-md-4 col-form-label text-md-end">{{ __('验
4 验证码')}}</label>
5             <div class="col-md-6 d-flex align-items-center">
6                 
8                 <input id="captcha" type="text" class="form-control @error('captcha')
9 is-invalid @enderror" name="captcha" required>
10                @error('captcha')
11                    <span class="invalid-feedback" role="alert">
12                        <strong>{{ $message }}</strong>
13                    </span>
14                @enderror
15            </div>
16        </div>
17
18        <div class="row mb-3">
19            <div class="col-md-6 offset-md-4">
20                <div class="form-check">
21                    <input class="form-check-input" type="checkbox"
22 name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>
23
24                    <label class="form-check-label" for="remember">
25                        {{ __('记住密码')}}
26                    </label>
27                </div>
28            </div>
29        </div>
30
31        <div class="row mb-0">
32            <div class="col-md-8 offset-md-4">
33                <button type="submit" class="btn btn-primary">
34                    {{ __('登录')}}
35                </button>
36
37                @if (Route::has('password.request'))
38                    <a class="btn btn-link" href="{{ route('password.request') }}">
39                        <!-- {{ __('Forgot Your Password?')}} -->
40                    </a>
41                @endif
42            </div>
43        </div>
44    </form>
45</div>
46</div>
47</div>
48</div>
49</div>
50@endsection

```

```

1
2  @extends('layouts.app')
3
4  @section('content')
5  <div class="container">
6      <div class="row justify-content-center">
7          <div class="col-md-8">
8              <div class="card">
9                  <div class="card-header">{{ __('注册') }}</div>
10
11                 <div class="card-body">
12                     <form method="POST" action="{{ route('register') }}">
13                         @csrf
14
15                         <div class="row mb-3">
16                             <label for="name" class="col-md-4 col-form-label text-md-end">{{ __('用户
17 名') }}</label>
18
19                             <div class="col-md-6">
20                                 <input id="name" type="text" class="form-control @error('name')
21  is-invalid @enderror" name="name" value="{{ old('name') }}" required autocomplete="name"
22  autofocus>
23
24                                 @error('name')
25                                     <span class="invalid-feedback" role="alert">
26                                         <strong>{{ $message }}</strong>
27                                     </span>
28                                 @enderror
29                             </div>
30                         </div>
31
32                         <div class="row mb-3">
33                             <label for="email" class="col-md-4 col-form-label text-md-end">{{ __('邮箱
34  ') }}</label>
35
36                             <div class="col-md-6">
37                                 <input id="email" type="email" class="form-control @error('email')
38  is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email">
39
40                                 @error('email')
41                                     <span class="invalid-feedback" role="alert">
42                                         <strong>{{ $message }}</strong>
43                                     </span>
44                                 @enderror
45                             </div>
46                         </div>
47
48                         <div class="row mb-3">
49                             <label for="password" class="col-md-4 col-form-label text-md-end">{{ __('
50  密码') }}</label>

```

```

1
2         <div class="col-md-6">
3             <input id="password" type="password" class="form-control
4 @error('password') is-invalid @enderror" name="password" required
5 autocomplete="new-password">
6
7             @error('password')
8                 <span class="invalid-feedback" role="alert">
9                     <strong>{{ $message }}</strong>
10                </span>
11            @enderror
12        </div>
13    </div>
14
15    <div class="row mb-3">
16        <label for="password-confirm" class="col-md-4 col-form-label
17 text-md-end">{{ __('确认密码 ') }}</label>
18
19        <div class="col-md-6">
20            <input id="password-confirm" type="password" class="form-control"
21 name="password_confirmation" required autocomplete="new-password">
22        </div>
23    </div>
24
25    <div class="row mb-0">
26        <div class="col-md-6 offset-md-4">
27            <button type="submit" class="btn btn-primary">
28                {{ __('注册') }}
29            </button>
30        </div>
31    </div>
32 </form>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 @endsection
39
40 @extends('layouts.app')
41
42 @section('content')
43 <div class="container">
44     <div class="row justify-content-center">
45         <div class="col-md-8">
46             <div class="card">
47                 <div class="card-header">{{ __('Verify Your Email Address') }}</div>
48
49                 <div class="card-body">
50                     @if (session('resent'))

```

江汉大学授权认证中心

```
1      <div class="alert alert-success" role="alert">
2          {{ __('A fresh verification link has been sent to your email address.') }}
3      </div>
4      @endif
5
6      {{ __('Before proceeding, please check your email for a verification link.') }}
7      {{ __('If you did not receive the email') }},
8      <form class="d-inline" method="POST" action="{{ route('verification.resend') }}">
9          @csrf
10         <button type="submit" class="btn btn-link p-0 m-0 align-baseline">{{ __('click
11 here to request another') }}</button>.
12     </form>
13 </div>
14 </div>
15 </div>
16 </div>
17 </div>
18 @endsection
19
20 <!doctype html>
21 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
22 <head>
23     <meta charset="utf-8">
24     <meta name="viewport" content="width=device-width, initial-scale=1">
25
26     <!-- CSRF Token -->
27     <meta name="csrf-token" content="{{ csrf_token() }}">
28
29     <title>{{ config('app.name', '江汉大学授权中心') }}</title>
30     <link rel="icon" href="{{ asset('jhun.png') }}" type="image/x-icon">
31     <!-- Fonts -->
32     <link rel="dns-prefetch" href="//fonts.bunny.net">
33     <link href="https://fonts.bunny.net/css?family=Nunito" rel="stylesheet">
34
35     <!-- Scripts -->
36     @viteReactRefresh
37     @vite(['resources/sass/app.scss', 'resources/js/app.js'])
38 </head>
39 <body>
40     <div id="app">
41         <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
42             <div class="container">
43                 <a class="navbar-brand" href="{{ url('/') }}">
44                     {{ config('江汉大学授权中心', '江汉大学授权中心') }}
45                 </a>
46                 <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
47 data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
48 aria-expanded="false" aria-label="{{ __('Toggle navigation') }}">
49                     <span class="navbar-toggler-icon"></span>
50                 </button>
```

```

1
2      <div class="collapse navbar-collapse" id="navbarSupportedContent">
3          <!-- Left Side Of Navbar -->
4          <ul class="navbar-nav me-auto">
5
6          </ul>
7
8          <!-- Right Side Of Navbar -->
9          <ul class="navbar-nav ms-auto">
10             <!-- Authentication Links -->
11             @guest
12                 @if (Route::has('login'))
13                     <li class="nav-item">
14                         <a class="nav-link" href="{{ route('login') }}">{{ __('登录') }}</a>
15                     </li>
16                 @endif
17
18                 @if (Route::has('register'))
19                     <li class="nav-item">
20                         <a class="nav-link" href="{{ route('register') }}">{{ __('注册') }}</a>
21                     </li>
22                 @endif
23             @else
24                 <li class="nav-item dropdown">
25                     <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#"
26 role="button" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>
27                         {{ Auth::user()->name }}
28                     </a>
29
30                     <div class="dropdown-menu dropdown-menu-end"
31 aria-labelledby="navbarDropdown">
32                         <a class="dropdown-item" href="{{ route('logout') }}"
33                             onclick="event.preventDefault();
34                                 document.getElementById('logout-form').submit();">
35                             {{ __('退出') }}
36                         </a>
37
38                         <form id="logout-form" action="{{ route('logout') }}"
39 method="POST" class="d-none">
40                             @csrf
41                         </form>
42                     </div>
43                 </li>
44             @endguest
45         </ul>
46     </div>
47 </div>
48 </nav>
49
50 <main class="py-4">

```

```

1      @yield('content')
2      </main>
3  </div>
4  </body>
5  </html>
6
7  @extends('layouts.app')
8
9  @section('content')
10 <div class="container">
11     <div class="row justify-content-center">
12         <div class="col-md-8">
13             <div class="text-center mb-4">
14                 <!-- Logo 图片 -->
15                 
16             </div>
17             <div class="card">
18                 <div class="card-header">{{ __('欢迎进入江汉大学授权认证中心') }}</div>
19
20                 <div class="card-body">
21                     @if (session('status'))
22                         <div class="alert alert-success" role="alert">
23                             {{ session('status') }}
24                         </div>
25                     @endif
26
27                     {{ __('登陆成功!') }}
28                 </div>
29             </div>
30         </div>
31     </div>
32 </div>
33 @endsection
34
35 <!DOCTYPE html>
36 <html>
37 <head>
38     <title>江汉大学授权中心</title>
39     <!-- Favicon -->
40     <link rel="icon" href="{{ asset('jhun.png') }}" type="image/x-icon">
41     <meta charset="UTF-8">
42     <meta name="viewport" content="width=device-width, initial-scale=1.0">
43 </head>
44 <body style="font-family: Arial, sans-serif; background-color: #f5f5f5; color: #333; margin: 0; padding:
45 0;">
46
47 <header style="display: flex; justify-content: center; gap: 20px; padding: 20px; background-color:
48 #ffffff; box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);">
49     <a href="{{ route('login') }}" style="text-decoration: none; color: #007bff; padding: 10px 20px;
50 border-radius: 4px; transition: background-color 0.3s;"

```

江汉大学授权认证中心

```
1      onmouseover="this.style.backgroundColor='#e2e6ea';"
2  onmouseout="this.style.backgroundColor=";">
3      登录
4  </a>
5      <a href="{{ route('register') }}" style="text-decoration: none; color: #007bff; padding: 10px 20px;
6  border-radius: 4px; transition: background-color 0.3s;"
7      onmouseover="this.style.backgroundColor='#e2e6ea';"
8  onmouseout="this.style.backgroundColor=";">
9      注册
10     </a>
11 </header>
12
13 <div id="app" style="display: flex; flex-direction: column; align-items: center; max-width: 1200px;
14 margin: 20px auto; padding: 20px;">
15     <!-- Center Logo -->
16     
18
19     <h1 style="text-align: center; color: #333; font-size: 2rem; margin-top: 20px;">
20         欢迎进入江汉大学授权认证中心
21     </h1>
22 </div>
23
24 </body>
25 </html>
26
27 /* Home.css */
28
29 /* 页面容器 */
30 .container {
31     display: flex;
32     flex-direction: column;
33     align-items: center;
34     padding: 20px;
35     background-color: #f9f9f9;
36     min-height: 100vh;
37 }
38
39 /* 标题样式 */
40 h1 {
41     font-size: 2rem;
42     margin-bottom: 20px;
43     color: #333;
44 }
45
46 /* 表格样式 */
47 table {
48     width: 100%;
49     border-collapse: collapse;
50     margin-top: 20px;
```



```

1      background-color: #fff;
2      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
3  }
4
5  /* 表头样式 */
6  th {
7      background-color: #007bff;
8      color: #fff;
9      padding: 10px;
10     text-align: left;
11 }
12
13 /* 表格单元格样式 */
14 td {
15     padding: 10px;
16     border-bottom: 1px solid #ddd;
17 }
18
19 /* 表格行的交替背景色 */
20 tr:nth-child(even) {
21     background-color: #f2f2f2;
22 }
23
24 /* 表格行 hover 样式 */
25 tr:hover {
26     background-color: #f1f1f1;
27 }
28
29 /* 错误信息样式 */
30 .error {
31     color: #d9534f;
32     font-size: 1rem;
33     margin-top: 20px;
34 }
35
36 /* 加载状态样式 */
37 .loading {
38     font-size: 1rem;
39     margin-top: 20px;
40 }
41
42 /* 按钮样式 */
43 .buttons {
44     margin-bottom: 20px;
45 }
46
47 .button {
48     background-color: #007bff;
49     color: #fff;
50     border: none;

```

```

1      border-radius: 4px;
2      padding: 10px 20px;
3      margin: 5px;
4      cursor: pointer;
5      font-size: 1rem;
6  }
7
8  .button:hover {
9      background-color: #0056b3;
10 }
11
12 .button.delete {
13     background-color: #dc3545;
14 }
15
16 .button.delete:hover {
17     background-color: #c82333;
18 }
19
20 .container {
21     padding: 20px;
22 }
23
24 .buttons {
25     margin-bottom: 20px;
26 }
27
28 .button {
29     padding: 10px 20px;
30     margin-right: 10px;
31     border: none;
32     border-radius: 5px;
33     cursor: pointer;
34 }
35
36 .button.delete {
37     background-color: red;
38     color: white;
39 }
40
41 .button.cancel {
42     background-color: gray;
43     color: white;
44 }
45
46 .form-container {
47     margin-bottom: 20px;
48     padding: 20px;
49     border: 1px solid #ddd;
50     border-radius: 5px;

```

```

1  }
2
3  .form-group {
4      margin-bottom: 15px;
5  }
6
7  .form-group label {
8      display: block;
9      margin-bottom: 5px;
10 }
11
12 .form-group input {
13     width: 100%;
14     padding: 8px;
15     border: 1px solid #ddd;
16     border-radius: 5px;
17 }
18 .navigation-buttons {
19     position: absolute;
20     top: 10px;
21     left: 10px;
22     display: flex;
23     gap: 10px;
24 }
25
26 .nav-button {
27     padding: 10px 20px;
28     font-size: 16px;
29     cursor: pointer;
30 }
31
32 .button {
33     padding: 10px 20px;
34     font-size: 16px;
35     cursor: pointer;
36 }
37
38 /* 其他样式... */
39
40 /* User.css */
41 /* .container {
42     padding: 20px;
43 }
44
45 table {
46     width: 100%;
47     border-collapse: collapse;
48 }
49
50 table, th, td {

```

```

1      border: 1px solid #ddd;
2  }
3
4  th, td {
5      padding: 8px;
6      text-align: left;
7  }
8
9  th {
10     background-color: #f4f4f4;
11 } */
12 /* 样式文件，例如 auth-form.css */
13
14 .auth-form {
15     max-width: 600px;
16     margin: 0 auto;
17     padding: 20px;
18     border-radius: 8px;
19     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
20     background-color: #f9f9f9;
21 }
22
23 .auth-form h2 {
24     text-align: center;
25     margin-bottom: 20px;
26     color: #333;
27 }
28
29 .form-group {
30     margin-bottom: 15px;
31 }
32
33 .form-group label {
34     display: block;
35     margin-bottom: 5px;
36     font-weight: bold;
37     color: #555;
38 }
39
40 .form-group select {
41     width: 100%;
42     padding: 10px;
43     border: 1px solid #ddd;
44     border-radius: 4px;
45     font-size: 16px;
46     box-sizing: border-box;
47     background-color: #fff;
48     cursor: pointer;
49 }
50

```

```

1  .form-group select:focus {
2      border-color: #007bff;
3      outline: none;
4  }
5
6  .submit-button {
7      display: block;
8      width: 100%;
9      padding: 10px;
10     border: none;
11     border-radius: 4px;
12     background-color: #007bff;
13     color: #fff;
14     font-size: 16px;
15     cursor: pointer;
16     transition: background-color 0.3s ease;
17 }
18
19 .submit-button:hover {
20     background-color: #0056b3;
21 }
22
23 /* src/pages/Login.module.css */
24 .container {
25     display: flex;
26     justify-content: center;
27     align-items: center;
28     height: 100vh; /* 使容器高度为视口高度 */
29     background-color: #f0f2f5; /* 背景色（可选） */
30 }
31
32 .loginForm {
33     width: 100%;
34     max-width: 400px; /* 设置表单最大宽度 */
35     padding: 20px;
36     border: 1px solid #ddd;
37     border-radius: 8px;
38     background-color: #fff;
39     box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1); /* 添加阴影（可选） */
40 }
41
42 .formGroup {
43     margin-bottom: 15px;
44 }
45
46 .formGroup label {
47     display: block;
48     margin-bottom: 5px;
49 }
50

```

```

1  .formGroup input {
2      width: 100%;
3      padding: 8px;
4      border: 1px solid #ddd;
5      border-radius: 4px;
6  }
7
8  .formGroup button {
9      width: 100%;
10     padding: 10px;
11     border: none;
12     border-radius: 4px;
13     background-color: #007bff;
14     color: white;
15     font-size: 16px;
16     cursor: pointer;
17 }
18
19 .formGroup button:hover {
20     background-color: #0056b3;
21 }
22
23 .message {
24     margin-top: 15px;
25     color: #ff0000; /* 错误消息颜色 */
26 }
27
28 .formGroup a {
29     display: block;
30     margin-top: 10px;
31     text-align: center;
32     color: #007bff;
33     text-decoration: none;
34 }
35
36 .formGroup a:hover {
37     text-decoration: underline;
38 }
39
40 <?php
41 namespace App\Http\Controllers;
42
43 use App\Models\User;
44 use Illuminate\Http\Request;
45 use Illuminate\Support\Facades\Validator;
46
47 class AuthController extends Controller
48 {
49     public function register(Request $request)
50     {

```

```

1      // 使用 Validator 进行验证
2
3      $validator = Validator::make($request->all(), [
4          'name' => 'required',
5          'email' => 'required',
6          'password' => 'required',
7      ]);
8
9      // 检查验证是否失败
10     if ($validator->fails()) {
11         return response()->json(['errors' => $validator->errors()], 422);
12     }
13     // dd(1);
14     // 创建用户
15     $user = User::create([
16         'name' => $request->name,
17         'email' => $request->email,
18         'password' => $request->password,
19     ]);
20
21     return response()->json(['message' => 'User successfully registered', 'user' => $user], 201);
22 }
23 public function login(Request $request)
24 {
25     // 验证请求数据
26     $validator = Validator::make($request->all(), [
27         'email' => 'required|email',
28         'password' => 'required',
29     ]);
30
31     if ($validator->fails()) {
32         return response()->json(['errors' => $validator->errors()], 422);
33     }
34
35     $credentials = $request->only('email', 'password');
36
37     $user = User::where('email', $credentials['email'])->first();
38
39     return response()->json($user, 200);
40 }
41
42
43 }
44
45

```