

宠爱网 API 文档

1. 用户登录

用户根据注册的账号进行登录

提供用户端登录的后端接口

接口调用: <http://localhost:8080/user/login>

请求类型: POST

请求参数:

字段	类型	说明
username	varchar	用户登录账号, 可使用英文和数字, 不可使用特殊字符, 字符长度最大不能超过 20 个字符
password	varchar	用户登录密码, 可使用英文、数字、下划线 '_', 字符长度最大不能超过 20 个字符, 最小为 11 个字符

返回的 JSON 数据包:

成功返回:

<pre>{ "state": "0", "msg": "登录成功", "data": { "user": [{ "id": "1", "usercode": "986710", "username": "tttyyyqq", "password": "null", "userSex": "null", "userPhone": "null", "userEmail": "null", "userIdCard": "null", "userImage1": "null", "userImage2": "null", "userTime": "null", "state": "null",</pre>
--

```
        "adminCode": "null",
        "adressId": "null",
        "addressList": "null",
        "isDelete": "null"
    ]
}
```

失败返回：

```
{
    "state": "1",
    "msg": "用户名不存在",
    "data": "null"
}
```

```
{
    "state": "2",
    "msg": "用户名存在，但密码错误",
    "data": "null"
}
```

2. 用户注册

用户进行自己账号以及密码的注册

接口调用：注册：<http://localhost:8080/user/register>

验证手机号码：<http://localhost:8080/user/checkPhoneUnique>

验证用户名：<http://localhost:8080/user/checkUsernameUnique>

请求类型: Post

请求参数:

字段	参数类型	说明
username	varchar	用户登录账号，可使用英文和数字，不可使用特殊字符，字符长度最大不能超过 20 个字符。且用户名唯一。
password	varchar	用户登录密码，可使用英文、数字、下划线'_'，字符长度最大不能超过 20 个字符，最小为 11

		个字符
phone	varchar	电话号码必须满足 ‘/^1[3456789]d{9}\$/’，即第一位必须是 1，第二位可以是 3,4,5,6,7,8,9 中的任意一位，后九位是【0123456789】组成的任意九位数，总共 11 位。且电话号码必须唯一

注册返回的 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "注册成功!",
  "data": {
    "user": [
      {
        "id": "1",
        "usercode": "986710",
        "username": "tttyyyqq",
        "password": "null",
        "userSex": "null",
        "userPhone": "null",
        "userEmail": "null",
        "userIdCard": "null",
        "userImage1": "null",
        "userImage2": "null",
        "userTime": "null",
        "state": "null",
        "adminCode": "null",
        "adressId": "null",
        "addressList": "null",
        "isDelete": "null"
      }
    ]
  }
}
```

失败：

```
{
  "state": "1",
```

```
{
  "msg": "注册失败, 请检查",
  "data": "null"
}
```

验证手机号唯一返回的 JSON 数据包:

成功:

```
{
  "state": "0",
  "msg": "该手机号可以使用",
  "data": "null"
}
```

失败:

```
{
  "state": "1",
  "msg": "该手机号已被注册过了",
  "data": "null"
}
```

验证用户账号唯一,返回的 JSON 数据包:

成功:

```
{
  "state": "0",
  "msg": "该用户名可以使用",
  "data": "null"
}
```

失败:

```
{
  "state": "1",
  "msg": "该用户名已被占用",
  "data": "null"
}
```

3. 用户完善个人详细信息

用户如果想要领养宠物,就必须进行个人信息的完善

接口调用：<http://localhost:8080/user/completeUserInfo>

请求类型：Post

请求参数：

字段	类型	说明
user	User	一个 User 对象，其中包含其用户的详细信息，包括个人邮箱、家庭地址、性别、年龄、身份证照片等。

返回 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "完善成功",
  "data": "null"
}
```

失败：

```
{
  "state": "1",
  "msg": "完善个人信息失败",
  "data": {
    "user": [
      {
        "id": "1",
        "usercode": "986710",
        "username": "tttyyyqq",
        "password": "xxx",
        "userSex": "xxx",
        "userPhone": "xxx",
        "userEmail": "xxx",
        "userIdCard": "xxx",
        "userImage1": "xxx",
        "userImage2": "xxx",
        "userTime": "xxx",
        "state": "xxx",
        "adminCode": "xxx",
        "adressId": "xxx",
        "addressList": "xxx",
      }
    ]
  }
}
```

```
        "isDelete": "xxx"
    ]
}
}
```

4. 查询全部的用户信息

管理员可以在后台查询全部的用户信息

调用接口: <http://localhost:8080/user/selectUserAll>

请求类型: Get

请求参数: 无

返回 JSON 数据包:

成功:

```
{
  "state": "0",
  "msg": "ok",
  "data": {
    "userList": [
      "user1": [
        "id": "1",
        "usercode": "986710",
        "username": "tttyyyqq",
        "password": "null",
        "userSex": "xxx",
        "userPhone": "xxx",
        "userEmail": "xxx",
        "userIdCard": "xxx",
        "userImage1": "xxx",
        "userImage2": "xxx",
        "userTime": "xxx",
        "state": "xxx",
        "adminCode": "xxx",
        "adressId": "xxx",
        "addressList": "xxx",
        "isDelete": "xxx"
      ]
    ]
  }
}
```

```
    ],
    "user2":[
      "id": "2",
      "usercode":"986799",
      "username":"qqppphhh",
      "password":"null",
      "userSex": "xxx",
      "userPhone": "xxx",
      "userEmail": "xxx",
      "userIdCard": "xxx",
      "userImage1": "xxx",
      "userImage2": "xxx",
      "userTime": "xxx",
      "state": "xxx",
      "adminCode": "xxx",
      "adressId": "xxx",
      "addressList": "xxx",
      "isDelete": "xxx"
    ]
  }
}
```

失败:

```
{
  "state": "1",
  "msg": "查询失败",
  "data": "null"
}
```

5. 根据用户编号对该用户详细信息进行查询

管理员可以在后台根据用户编号查询该用户信息

调用接口: <http://localhost:8080/user/selectUserByUserCode>

请求类型: Get

请求参数:

字段	类型	说明
usercode	varchar	用户编号, 随机生成的 6 位数字字符串, 唯一

返回 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "ok",
  "data": {
    "user": {
      "id": "1",
      "usercode": "986710",
      "username": "ttxyyqq",
      "password": "null",
      "userSex": "xxx",
      "userPhone": "xxx",
      "userEmail": "xxx",
      "userIdCard": "xxx",
      "userImage1": "xxx",
      "userImage2": "xxx",
      "userTime": "xxx",
      "state": "xxx",
      "adminCode": "xxx",
      "adressId": "xxx",
      "addressList": "xxx",
      "isDelete": "xxx"
    }
  }
}
```

失败：

```
{
  "state": "1",
  "msg": "查询失败",
  "data": "null"
}
```

6. 根据用户编号对该用户信息进行删除

管理员可以在后台根据用户编号删除该用户信息，注意，此处仅仅是逻辑删除，isDelete=1 表示该用户被删除，isDelete=0 表示该用户没有被删除

调用接口：<http://localhost:8080/user/updateUserIsDeleteByUCode>

请求类型：Post

请求参数：

字段	类型	说明
usercode	varchar	用户编号，随机生成的 6 位数字字符串，唯一

返回 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "删除成功!",
  "data": "null"
}
```

失败：

```
{
  "state": "1",
  "msg": "删除失败，请检查用户编码是否正确！ ",
  "data": "null"
}
```

7. 用户提交领养宠物的申请

用户在宠物详情页面可以进行宠物的领养申请，申请之后等待管理员的审批

调用接口：<http://localhost:8080/adopt/>

请求类型：Post

请求参数：

字段	类型	说明
usercode	varchar	用户编号，随机生成的 6 位数字字符串，唯一

返回 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "申请提交成功，等待管理员审批",
  "data": "null"
}
```

失败：

```
{
  "state": "1",
  "msg": "申请提交失败，请检查您的信息是否完整 huo",
  "data": "null"
}
```

```
}
```

8. 新增宠物

管理员可以新增宠物，返回时隐藏重要字段

接口调用：<http://localhost:8080/pet/insertPet>

请求类型: Post

请求参数:

字段	类型	说明
name	varchar	宠物名称，可用英文或汉字，字符长度不大于 20 即可
birthday	timestamp	宠物的生日
age	int	宠物的年龄
sex	int	宠物的性别,1 表示男性，0 表示女性
describe	varchar	对宠物性格、体型等描述，字符长度不大于 255 即可
categoryCode	varchar	宠物的类别编号
isDelete	varchar	宠物是否被删除，默认 0 表示没有被删除，若要进行删除修改其值为 1 即可
code	varchar	宠物的编号，由系统自动生成一个 6 位数的字符串
state	int	宠物目前是否被领养，0 为默认值表示没有被领养，1 表示已经被领养

返回 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "新增宠物成功! ",
  "data": {
    "pet": {
      "id": "1",
      "petCode": "908882",
      "petBirthday": "2020-10-21",
      "petAge": "2",
      "petSex": "1",
```

```
        "petDescribe": "宠物性格温和， 体型偏小， 非常可爱",
        "petState": "0",
        "categoryCode": "9829",
        "isDelete": "0"
    }
}
```

失败:

```
{
    "state": "1",
    "msg": "新增宠物失败， 请检查您填入信息的内容是否正确！ ",
    "data": "null"
}
```

9. 删除宠物信息

管理员可以根据宠物编号对宠物信息进行删除（被领养后），仅需修改 isDelete 字段值即可

接口调用: <http://localhost:8080/pet/updatePetIsDeleteByPCode>

请求类型: Post

请求参数:

字段	类型	说明
petCode	varchar	宠物的编号，系统随机生成的六位数字符串，且唯一

返回 JSON 数据包:

成功:

```
{
    "state": "0",
    "msg": "删除成功!",
    "data": "null"
}
```

失败:

```
{
    "state": "1",
    "msg": "删除失败， 请检查用户编码是否正确！ ",
    "data": "null"
}
```

10.根据编号查询宠物的详细信息

管理员可以新增宠物, 返回时隐藏重要字段

接口调用: <http://localhost:8080/pet/selectPetByPetCode>

请求类型: Get

请求参数:

字段	类型	说明
petCode	varchar	宠物的编号, 系统随机生成的六位数字符串, 且唯一

返回 JSON 数据包:

成功:

```
{
  "state": "0",
  "msg": "查询宠物信息成功! ",
  "data": {
    "pet": {
      "id": "1",
      "petCode": "908882",
      "petBirthday": "2020-10-21",
      "petAge": "2",
      "petSex": "1",
      "petDescribe": "宠物性格温和, 体型偏小, 非常可爱",
      "petState": "0",
      "categoryCode": "982",
      "isDelete": "0"
    }
  }
}
```

失败:

```
{
  "state": "1",
  "msg": "查询宠物信息失败, 请检查您的宠物编号是否正确! "
  "data": "null"
}
```

11.根据宠物类别编号查询这一类别的全部宠物信息

根据宠物类别编号查询这一类别的全部宠物，数据返回时可隐藏重要字段

接口调用：<http://localhost:8080/pet/selectPetByCategoryCode>

请求类型: Get

请求参数:

字段	类型	说明
categoryCode	varchar	宠物类别的编号，系统随机生成的三位数字字符串，且唯一

返回 JSON 数据包：

成功：

```
{
  "state": "0",
  "msg": "查询成功! ",
  "data": {
    "petList": [
      "pet1": {
        "id": "1",
        "petCode": "908882",
        "petBirthday": "2020-10-21",
        "petAge": "2",
        "petSex": "1",
        "petDescribe": "宠物性格温和，体型偏小，非常可爱",
        "petState": "0",
        "categoryCode": "9829",
        "isDelete": "0"
      },
      "pet2": {
        "id": "2",
        "petCode": "778882",
        "petBirthday": "2020-7-21",
        "petAge": "3",
        "petSex": "0",
        "petDescribe": "宠物性格温和，体型偏小，非常可爱",
        "petState": "0",
        "categoryCode": "982",

```

```
        "isDelete": "0"
      }
    ]
  }
}
```

失败:

```
{
  "state": "1",
  "msg": "查询宠物信息失败, 请检查您的宠物类别编号是否正确! ",
  "data": "null"
}
```