

Coffee Bean Detect Project API 文档 (美化版)

统一接口前缀: `/api` 返回统一封装:

```
{
  "code": 200,
  "message": "操作信息",
  "data": {}
}
```

目录

- [1. 用户相关接口](#)
- [2. Coze 配置接口](#)
- [3. 咖啡豆缺陷检测接口](#)

用户相关接口 - UserController

接口前缀: `/api/user`

1. 修改用户昵称

- URL: `/api/user/update/name`
- 方法: POST
- 请求头:

```
Authorization: <JWT Token>
Content-Type: application/x-www-form-urlencoded
```

- 请求参数: | 参数 | 类型 | 必填 | 说明 | |-----|-----|-----|-----| | name | String | 是 | 新昵称 |

<details> <summary>请求示例 (cURL)</summary>

```
curl -X POST "http://localhost:8080/api/user/update/name"
-H "Authorization: Bearer <JWT Token>"
-d "name=Jayden"
```

</details>

<details> <summary>返回示例</summary>

```
{
  "code": 200,
  "message": "昵称修改成功",
  "data": null
}
```

</details>

2. 上传/修改用户头像

- URL: `/api/user/update/avatar`
- 方法: `POST`
- 请求头:

```
Authorization: <JWT Token>
Content-Type: multipart/form-data
```

- 请求参数: | 参数 | 类型 | 必填 | 说明 | |-----|-----|-----|-----| | file | MultipartFile | 是 | 上传头像文件 |

<details> <summary>请求示例 (cURL)</summary>

```
curl -X POST "http://localhost:8080/api/user/update/avatar"
-H "Authorization: Bearer <JWT Token>"
-F "file=@/path/to/avatar.jpg"
```

</details>

<details> <summary>返回示例</summary>

```
{
  "code": 200,
  "message": "头像更新成功",
  "data": "http://localhost:8080/uploads/avatar.jpg"
}
```

</details>

3. 用户注册

- URL: `/api/user/register`
- 方法: `POST`
- 请求头:

```
Content-Type: application/json
```

- 请求体:

```
{
  "username": "testUser",
  "password": "123456",
  "email": "test@example.com"
}
```

<details> <summary>返回示例</summary>

```
{
  "code": 200,
  "message": "注册成功",
  "data": null
}
```

</details>

4. 用户登录

- URL: /api/user/login
- 方法: POST
- 请求体:

```
{
  "username": "testUser",
  "password": "123456"
}
```

<details> <summary>返回示例</summary>

```
{
  "code": 200,
  "message": "登录成功",
  "data": {
    "id": 1,
    "username": "testUser",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5In0.eyJ1IjoiMSIsInVzIjoiYm9keSIsImVhbnQiOiJ1b3RlbnQifQ"
  }
}
```

</details>

5. 密码重置

- URL: /api/user/reset-password
- 方法: POST
- 请求体:

```
{
  "username": "testUser",
  "oldPassword": "123456",
  "newPassword": "abcdef"
}
```

<details> <summary>返回示例</summary>

```
{
  "code": 200,
  "message": "密码重置成功",
  "data": null
}
```

</details>

Coze 配置接口 - CozeController

接口前缀: /api/coze

获取 Coze 配置

- URL: /api/coze/config
- 方法: GET

<details> <summary>返回示例</summary>

```
{
  "botId": "your-bot-id",
  "token": "your-pat-token"
}
```

</details>

⚠ 生产环境请使用临时 token 或服务端代理，不直接返回 PAT。

咖啡豆缺陷检测接口 - CoffeeDefectController

接口前缀: /api/coffee

上传图片并检测缺陷

- URL: /api/coffee/detect
- 方法: POST

- **请求头:** | Header | 值 | |-----|----| | userId | <用户ID> | | Content-Type | multipart/form-data |
- **请求参数:** | 参数 | 类型 | 必填 | 说明 | |-----|-----|-----|-----| | file | MultipartFile | 是 | 上传图片 |

<details> <summary>请求示例 (cURL)</summary>

```
curl -X POST "http://localhost:8080/api/coffee/detect"
-H "userId: 1"
-F "file=@/path/to/coffee.jpg"
```

</details>

<details> <summary>返回示例</summary>

```
{
  "code": 200,
  "message": "检测成功",
  "data": {
    "imagePath": "/uploads/temp.jpg",
    "defects": [
      {"type": "黑豆", "x": 100, "y": 150, "width": 50, "height": 50},
      {"type": "裂豆", "x": 200, "y": 300, "width": 60, "height": 60}
    ]
  }
}
```

</details>