**自动贩卖机项目 API 文档**

**用户端控制器 (UserController)**

**1. 获取所有商品**

* **URL**: /user/list
* **方法**: GET
* **描述**: 获取所有可售商品的列表。
* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": [
* {
* "id": 1,
* "name": "Coca-Cola",
* "price": 3.5,
* "stock": 10
* },
* {
* "id": 2,
* "name": "Pepsi",
* "price": 3.0,
* "stock": 5
* }
* ]

}

**2. 根据 ID 获取商品详情**

* **URL**: /user/product/{id}
* **方法**: GET
* **描述**: 根据商品 ID 获取商品的详细信息。
* **参数**:
  + id (路径参数): 商品的唯一标识符。
* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": {
* "id": 1,
* "name": "Coca-Cola",
* "price": 3.5,
* "stock": 10
* }

}

**3. 购买商品**

* **URL**: /user/sell
* **方法**: POST
* **描述**: 根据提供的商品信息完成购买。
* **请求体**:
* {
* "product\_id": 1,（购买商品的id）
* "quantity": 2,（购买商品的数量）
* "total\_price": 7.0（购买商品的总价）

}

* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": null

}

**管理端控制器 (AdminController)**

**1. 管理员登录**

* **URL**: /admin/login
* **方法**: POST
* **描述**: 管理员登录，生成并返回 JWT 令牌。
* **请求体**:
* {
* "username": "admin",
* "password": "password123"

}

* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": {
* "id": 1,
* "username": "admin",
* "token": "eyJhbGciOiJIUzI1NiIsInR5cCI..."
* }

}

**2. 创建商品**

* **URL**: /admin
* **方法**: POST
* **描述**: 添加一个新的商品到系统中。
* **请求体**:
* {
* "name": "Sprite",
* "price": 3.2,
* "stock": 15

}

* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": {
* "id": 3,
* "name": "Sprite",
* "price": 3.2,
* "stock": 15
* }

}

**3. 删除商品**

* **URL**: /admin/{id}
* **方法**: DELETE
* **描述**: 删除指定 ID 的商品。
* **参数**:
  + id (路径参数): 商品的唯一标识符。
* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": null

}

**4. 获取所有商品**

* **URL**: /admin/list
* **方法**: GET
* **描述**: 获取所有商品的列表（与用户端类似）。
* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": [
* {
* "id": 1,
* "name": "Coca-Cola",
* "price": 3.5,
* "stock": 10
* },
* {
* "id": 2,
* "name": "Pepsi",
* "price": 3.0,
* "stock": 5
* }
* ]

}

**5. 根据 ID 获取商品详情**

* **URL**: /admin/{id}
* **方法**: GET
* **描述**: 根据商品 ID 获取商品的详细信息（与用户端类似）。
* **参数**:
  + id (路径参数): 商品的唯一标识符。
* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": {
* "id": 1,
* "name": "Coca-Cola",
* "price": 3.5,
* "stock": 10
* }

}

**6. 更新商品信息**

* **URL**: /admin/update
* **方法**: PUT
* **描述**: 更新现有商品的信息。
* **请求体**:
* {
* "id": 1,
* "name": "Coca-Cola",
* "price": 4.0,
* "stock": 20

}

* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": null

}

**错误响应示例**

* **错误响应**:
* {
* "code": 0,
* "msg": "具体的错误信息",
* "data": null

}

* **描述**: 当操作失败时，返回的 code 为 0，并提供相应的错误信息。

**7. 文件上传**

* **URL**: /admin/upload
* **方法**: POST
* **描述**: 上传文件并返回文件的请求路径。
* **请求参数**:
  + file (请求体): 要上传的文件，类型为 MultipartFile。
* **响应示例**:
* {
* "code": 1,
* "msg": null,
* "data": "https://your-oss-url.com/path/to/file.png"

}

* **错误响应示例**:
* {
* "code": 0,
* "msg": "文件上传失败",
* "data": null

}

**错误响应示例**

* **错误响应**:
* {
* "code": 0,
* "msg": "具体的错误信息",
* "data": null

}

* **描述**: 当操作失败时，返回的 code 为 0，并提供相应的错误信息。