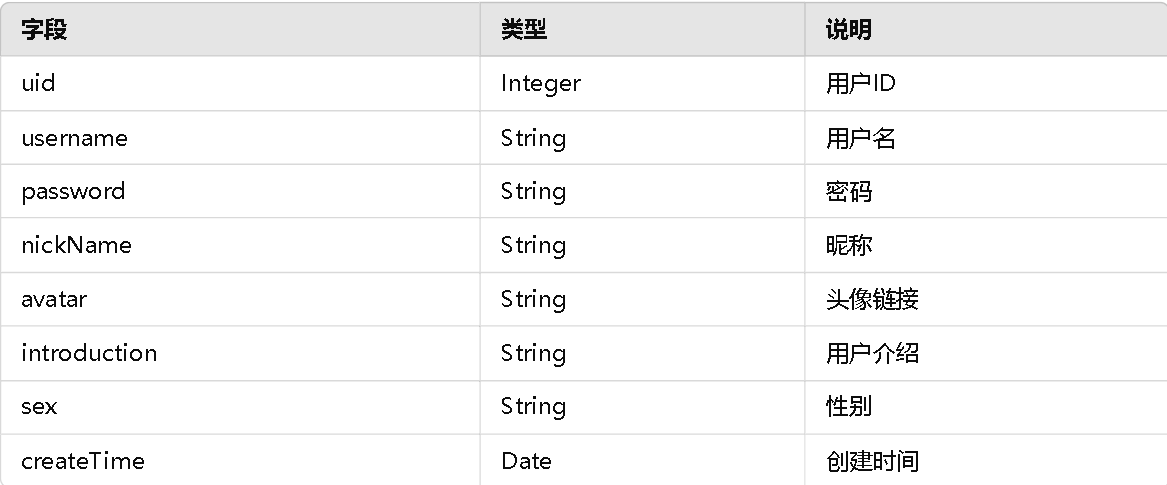
# iDouShu接口说明文档

## 数据库设计

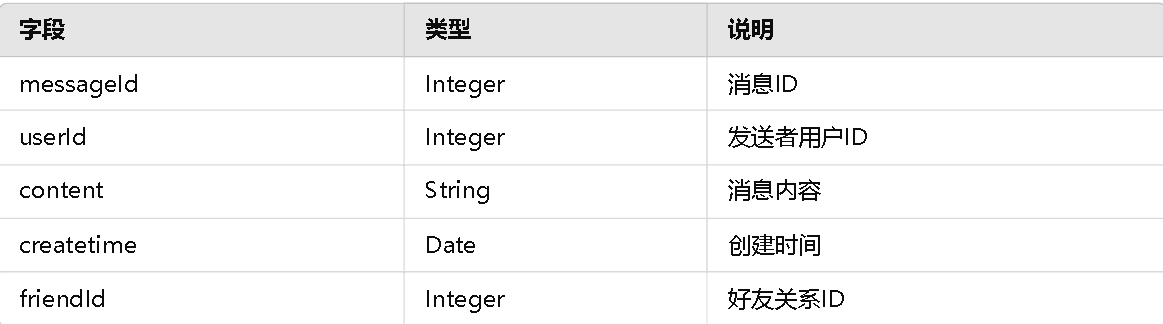
**User表**

****

**note表**

****

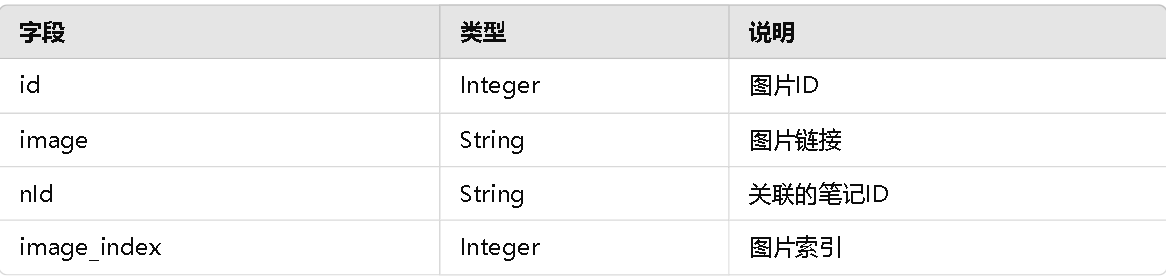
**message表**

****

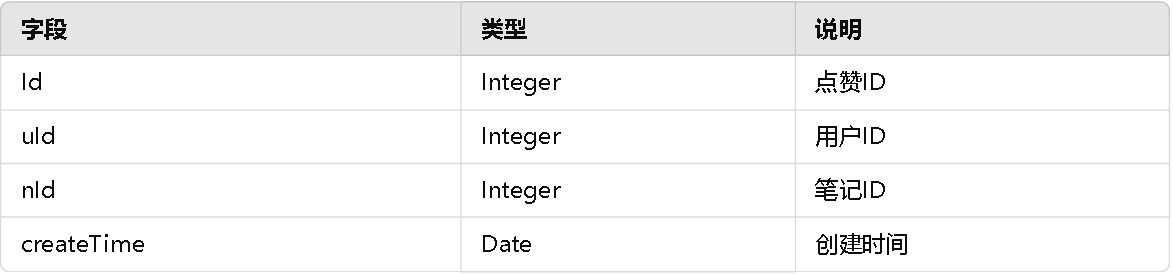
**collection收藏表**



**笔记image表**

****

**like点赞表**



**Comment评论表**



**接口使用说明：**

**用户：**

**1. 用户注册**

* **请求方法: POST**
* **请求 URL: /user/register**
* **请求体:**
  + **User 对象（包含用户的所有信息）**
* **响应:**
  + **返回注册成功后的用户信息（User 对象）**

**2. 用户登录**

* **请求方法: POST**
* **请求 URL: /user/login**
* **请求体:**
  + **包含用户名和密码的键值对（Map<String, String>）**
* **请求参数:**
  + **username (String, 必填): 用户名**
  + **password (String, 必填): 密码**
* **响应:**
  + **返回登录成功的用户信息（User 对象）**

**3. 更新用户信息**

* **请求方法: PUT**
* **请求 URL: /user/update**
* **请求体:**
  + **User 对象（包含更新后的用户信息）**
* **响应:**
  + **返回更新操作是否成功（boolean）**

**4. 根据用户 ID 查询用户信息**

* **请求方法: GET**
* **请求 URL: /user/search?id=1**
* **请求参数:**
  + **id (Integer, 必填): 用户 ID**
* **响应:**
  + **返回查询到的用户信息（User 对象）**

**笔记：**

**1. 按创建时间查询和分页**

**请求方法: GET**

**请求 URL: /note/list?type=time&page=1&size=10**

**请求参数:**

**type (String, 必填): 排序类型（time 表示按时间排序，good 表示按点赞数排序）**

**page (int, 必填): 当前页码**

**size (int, 必填): 每页大小**

**响应:**

**返回分页结果（Page<Note>）**

**2. 根据用户 ID 查询笔记列表**

**请求方法: GET**

**请求 URL: /note/mylist?userid=1**

**请求参数:**

**userid (String, 必填): 用户 ID**

**响应:**

**返回笔记列表（List<Note>）**

**3. 添加新的笔记**

**请求方法: POST**

**请求 URL: /note/add**

**请求体:**

**Note 对象（包含笔记的所有信息）**

**响应:**

**返回操作结果（Map<String, Object>），包含 success 和 noteId 或 message**

**4. 根据 ID 获取单个笔记信息**

**请求方法: GET**

**请求 URL: /note/get?id=N123456789**

**请求参数:**

**id (String, 必填): 笔记 ID**

**响应:**

**返回笔记对象（Note）**

**5. 更新笔记**

**请求方法: PUT**

**请求 URL: /note/update**

**请求体:**

**Note 对象（包含更新后的笔记信息）**

**响应:**

**返回操作结果（String: "true" 或 "false"）**

**6. 根据关键词搜索笔记**

**请求方法: GET**

**请求 URL: /note/search?keyword=笔记&page=1&size=10**

**请求参数:**

**keyword (String, 必填): 搜索关键词**

**page (int, 必填): 当前页码**

**size (int, 必填): 每页大小**

**响应:**

**返回分页结果（Page<Note>）**

**好友：**

**1. 添加好友**

* **请求方法: GET**
* **请求 URL: /friend/add/{user1Id}/{user2Id}**
* **路径参数:**
  + **user1Id (Integer, 必填): 用户 1 的 ID**
  + **user2Id (Integer, 必填): 用户 2 的 ID**
* **响应:**
  + **返回好友 ID（Integer）**

**2. 更新好友消息**

* **请求方法: PUT**
* **请求 URL: /friend/update/{fid}/{message}**
* **路径参数:**
  + **fid (Integer, 必填): 消息 ID**
  + **message (String, 必填): 新消息内容**
* **响应:**
  + **操作成功返回 "消息已更新"**
  + **失败返回 "更新失败"**

**3. 标记好友消息为已读**

* **请求方法: PUT**
* **请求 URL: /friend/read/{user1Id}/{user2Id}**
* **路径参数:**
  + **user1Id (Integer, 必填): 用户 1 的 ID**
  + **user2Id (Integer, 必填): 用户 2 的 ID**
* **响应:**
  + **操作成功返回 "消息已标记为已读"**
  + **失败返回 "标记失败"**

**4. 查询好友列表**

* **请求方法: GET**
* **请求 URL: /friend/list/{userId}**
* **路径参数:**
  + **userId (Integer, 必填): 用户 ID**
* **响应:**
  + **返回好友列表（List<Friend>）**

**消息：**

**Websocket连接：ws://localhost:8082/websocket**

**在websocket的send方法里面发送请求，格式为：**

**{**

**"action": "send",**

**"userid": "1",**

**"touserid": "2",**

**"content": "Hello!",**

**"name": "User1",**

**"friendid": 123**

**}**

**消息会同步添加到数据库。**

**Message表接口：**

**1. 根据 friendId 分页查询消息**

* **请求方法: GET**
* **请求 URL: /message/list/{friendId}?pageNum=1&pageSize=30**
* **路径参数:**
  + **friendId (Integer, 必填): 朋友 ID**
* **请求参数:**
  + **pageNum (int, 可选, 默认值: 1): 当前页码**
  + **pageSize (int, 可选, 默认值: 30): 每页大小**
* **响应:**
  + **返回分页结果（IPage<Message>）**

**2. 添加新消息**

* **请求方法: POST**
* **请求 URL: /message/add**
* **请求体:**
  + **Message 对象（包含消息的所有信息）**
* **响应:**
  + **返回操作结果（boolean）**

**3. 删除指定消息**

* **请求方法: DELETE**
* **请求 URL: /message/delete/{messageId}**
* **路径参数:**
  + **messageId (Integer, 必填): 消息 ID**
* **响应:**
  + **返回操作结果（boolean）**

**收藏**

**1. 添加收藏**

* **请求方法: POST**
* **请求 URL: /collection/add?uId=1&nId=101**
* **请求参数:**
  + **uId (Integer, 必填): 用户 ID**
  + **nId (Integer, 必填): 笔记 ID**

**2. 删除收藏**

* **请求方法: DELETE**
* **请求 URL: /collection/delete?uId=1&nId=101**
* **请求参数:**
  + **uId (Integer, 必填): 用户 ID**
  + **nId (Integer, 必填): 笔记 ID**

**3. 根据用户 ID 获取收藏列表**

* **请求方法: GET**
* **请求 URL: /collection/user/1**
* **路径参数:**
  + **uId (Integer, 必填): 用户 ID**

**评论：**

**1. 添加评论**

* **请求方法: POST**
* **请求 URL: /comment/add**
* **请求参数:**
  + **Comment 对象（包含评论的所有信息）**

**2. 删除评论**

* **请求方法: DELETE**
* **请求 URL: /comment/delete/{id}**
* **路径参数:**
  + **id (Integer, 必填): 评论 ID**

**3. 更新评论**

* **请求方法: PUT**
* **请求 URL: /comment/update**
* **请求参数:**
  + **Comment 对象（包含更新后的评论信息）**

**4. 根据 ID 获取评论**

* **请求方法: GET**
* **请求 URL: /comment/get/{id}**
* **路径参数:**
  + **id (Integer, 必填): 评论 ID**

**5. 获取所有评论**

* **请求方法: GET**
* **请求 URL: /comment/list?nid=101&page=1&size=20**
* **请求参数:**
  + **nid (String, 必填): 笔记 ID**
  + **page (int, 可选, 默认值: 1): 当前页码**
  + **size (int, 可选, 默认值: 20): 每页大小**

**6. 根据父评论 ID 获取子评论**

* **请求方法: GET**
* **请求 URL: /comment/children/{parentCommentId}**
* **路径参数:**
  + **parentCommentId (Integer, 必填): 父评论 ID**

**点赞：**

**1. 添加点赞**

* **请求方法: POST**
* **请求 URL: /like/add?uId=1&nId=101**
* **请求参数:**
  + **uId (Integer, 必填): 用户 ID**
  + **nId (Integer, 必填): 笔记 ID**

**2. 删除点赞**

* **请求方法: DELETE**
* **请求 URL: /like/delete?uId=1&nId=101**
* **请求参数:**
  + **uId (Integer, 必填): 用户 ID**
  + **nId (Integer, 必填): 笔记 ID**

**3. 根据用户 ID 获取用户的所有点赞记录**

* **请求方法: GET**
* **请求 URL: /like/user/{uId}**
* **路径参数:**
  + **uId (Integer, 必填): 用户 ID**
* **响应:**
  + **返回点赞列表（List<Like>）**

**图片：**

**图片上传：**

**1. 上传图片**

* **请求方法: POST**
* **请求 URL: /upload**
* **请求参数:**
  + **file (MultipartFile, 必填): 要上传的图片文件**

**Image表接口：**

**1. 添加图片**

* **请求方法: POST**
* **请求 URL: /image/add/{nId}**
* **路径参数:**
  + **nId (String, 必填): 笔记 ID**
* **请求体:**
  + **Image 对象（包含图片的所有信息）**
* **响应:**
  + **返回操作结果（boolean）**

**2. 根据笔记 ID 获取所有图片**

* **请求方法: GET**
* **请求 URL: /image/list/{nId}**
* **路径参数:**
  + **nId (String, 必填): 笔记 ID**
* **响应:**
  + **返回图片列表（List<Image>）**

**3. 更新指定笔记 ID 下的所有图片**

* **请求方法: PUT**
* **请求 URL: /image/update/{nId}**
* **路径参数:**
  + **nId (String, 必填): 笔记 ID**
* **请求体:**
  + **图片列表（List<Image>）**
* **响应:**
  + **返回操作结果（boolean）**

**4. 删除指定笔记 ID 下的所有图片**

* **请求方法: DELETE**
* **请求 URL: /image/delete/{nId}**
* **路径参数:**
  + **nId (String, 必填): 笔记 ID**
* **响应:**
  + **返回操作结果（boolean）**