**接口文档**

# **管理端接口**

## **1.1 获取所有注册请求**

URL: /api/admin/registrations

方法: GET

描述: 获取所有注册请求

响应示例:

[  
 {  
 "userId": 1,  
 "username": "newuser",  
 "name": "New User",  
 "contactInfo": "contact@example.com",  
 "role": "teacher",  
 "status": "pending"  
 }  
]

## **1.2 审核注册请求**

URL: /api/admin/registrations/{userId}

方法: PUT

描述: 审核注册请求

请求体示例:

{  
 "status": "approved"  
}

响应示例:

{  
 "code": 200,  
 "message": "User registration approved"  
}

## **1.3 获取统计数据**

URL: /api/admin/statistics

方法: GET

描述: 获取所有统计数据（提交、用户、比赛、评审）

响应示例:

{  
 "totalSubmissions": 100,  
 "totalUsers": 50,  
 "totalCompetitions": 5,  
 "totalReviews": 20  
}

## **1.4 获取所有用户**

URL: /api/admin/users

方法: GET

描述: 获取所有用户信息

响应示例:

[  
 {  
 "userId": 1,  
 "username": "admin",  
 "role": "admin",  
 "name": "Admin User",  
 "contactInfo": "admin@example.com"  
 }  
]

## **1.5 创建用户**

URL: /api/admin/users

方法: POST

描述: 创建新用户

请求体示例:

{  
 "username": "newuser",  
 "password": "password123",  
 "role": "teacher",  
 "name": "New User",  
 "contactInfo": "newuser@example.com"  
}

响应示例:

{  
 "message": "User created successfully"  
}

## **1.6 更新用户信息**

URL: /api/admin/users/{id}

方法: PUT

描述: 更新用户信息

请求体示例:

{  
 "username": "updateduser",  
 "password": "newpassword123",  
 "role": "teacher",  
 "name": "Updated User",  
 "contactInfo": "updateduser@example.com"  
}

响应示例:

{  
 "message": "User updated successfully"  
}

## **1.7 删除用户**

URL: /api/admin/users/{id}

方法: DELETE

描述: 删除用户

响应示例:

{  
 "message": "User deleted successfully"  
}

## **1.8 获取所有比赛**

URL: /api/admin/competitions

方法: GET

描述: 获取所有比赛信息

响应示例:

[  
 {  
 "competitionId": 1,  
 "competitionName": "Competition 1",  
 "description": "Description for Competition 1",  
 "competitionDate": "2024-07-01"  
 }  
]

## **1.9 创建比赛**

URL: /api/admin/competitions

方法: POST

描述: 创建新比赛

请求体示例:

{  
 "competitionName": "New Competition",  
 "description": "Competition Description",  
 "competitionDate": "2024-07-01"  
}

响应示例:

{  
 "message": "Competition created successfully"  
}

## **1.10 更新比赛信息**

URL: /api/admin/competitions/{id}

方法: PUT

描述: 更新比赛信息

请求体示例:

{  
 "competitionName": "Updated Competition",  
 "description": "Updated Description",  
 "competitionDate": "2024-07-02"  
}

响应示例:

{  
 "message": "Competition updated successfully"  
}

## **1.11 删除比赛**

URL: /api/admin/competitions/{id}

方法: DELETE

描述: 删除比赛

响应示例:

{  
 "message": "Competition deleted successfully"  
}

## **1.12 获取所有提交**

URL: /api/admin/submissions

方法: GET

描述: 获取所有提交的信息

响应示例:

[  
 {  
 "submissionId": 1,  
 "submissionName": "Submission 1",  
 "category": "思政课程组",  
 "school": "School 1",  
 "team": "Team 1",  
 "materialPath": "/path/to/material",  
 "submissionDate": "2024-07-01T12:00:00Z",  
 "userId": 1,  
 "competitionId": 1  
 }  
]

## **1.13 删除提交**

URL: /api/admin/submissions/{id}

方法: DELETE

描述: 删除提交

响应示例:

{  
 "message": "Submission deleted successfully"  
}

## **1.14 获取所有评审**

URL: /api/admin/reviews

方法: GET

描述: 获取所有评审的信息

响应示例:

[  
 {  
 "reviewId": 1,  
 "submissionId": 1,  
 "expertId": 2,  
 "score": 95,  
 "comment": "Great submission",  
 "reviewDate": "2024-07-01T12:00:00Z"  
 }  
]

## **1.15 获取所有日志**

URL: /api/admin/logs

方法: GET

描述: 获取所有操作日志

响应示例:

[  
 {  
 "logId": 1,  
 "userId": 1,  
 "operation": "User login",  
 "operationDate": "2024-07-01T12:00:00Z"  
 }  
]

# **用户接口**

## **2.1 用户注册**

URL: /api/auth/register

方法: POST

描述: 用户注册

请求体示例:

{  
 "username": "newuser",  
 "password": "password123",  
 "role": "teacher",  
 "name": "New User",  
 "contactInfo": "contact@example.com"  
}

响应示例:

{  
 "code": 200,  
 "message": "Registration successful, waiting for approval"  
}

## **2.2 登录**

URL: /api/auth/login

方法: POST

描述: 用户登录

请求体示例:

{  
 "username": "user1",  
 "password": "password123"  
}

响应示例:

{  
 "code": 200,  
 "message": "Login successful",  
 "data": {  
 "token": "jwt-token-string"  
 }  
}

## **2.3 修改密码**

URL: /api/auth/change-password

方法: POST

描述: 修改用户密码

请求体示例:

{  
 "username": "user1",  
 "oldPassword": "oldpassword123",  
 "newPassword": "newpassword123"  
}

响应示例:

{  
 "code": 200,  
 "message": "Password changed successfully"  
}

## **2.4 提交作品（参赛教师）**

URL: /api/submissions

方法: POST

描述: 提交新作品

请求体示例:

{  
 "submissionName": "New Submission",  
 "category": "思政课程组",  
 "school": "School 1",  
 "team": "Team 1",  
 "materialPath": "/path/to/material",  
 "userId": 1,  
 "competitionId": 1  
}

响应示例:

{  
 "message": "Submission created successfully"  
}

## **2.5 获取所有提交（参赛教师）**

URL: /api/submissions

方法: GET

描述: 获取所有提交的信息

响应示例:

[  
 {  
 "submissionId": 1,  
 "submissionName": "Submission 1",  
 "category": "思政课程组",  
 "school": "School 1",  
 "team": "Team 1",  
 "materialPath": "/path/to/material",  
 "submissionDate": "2024-07-01T12:00:00Z",  
 "userId": 1,  
 "competitionId": 1  
 }  
]

## **2.6 更新提交（参赛教师）**

URL: /api/submissions/{id}

方法: PUT

描述: 更新提交的信息

请求体示例:

{  
 "submissionName": "Updated Submission",  
 "category": "思政课程组",  
 "school": "Updated School",  
 "team": "Updated Team",  
 "materialPath": "/path/to/new/material",  
 "userId": 1,  
 "competitionId": 1  
}

响应示例:

{  
 "message": "Submission updated successfully"  
}

## **2.7 删除提交（参赛教师）**

URL: /api/submissions/{id}

方法: DELETE

描述: 删除提交

响应示例:

{  
 "message": "Submission deleted successfully"  
}

## **2.8 获取所有提交（评审专家）**

URL: /api/expert/submissions

方法: GET

描述: 获取评审专家可以评审的所有提交

响应示例:

[  
 {  
 "submissionId": 1,  
 "submissionName": "Submission 1",  
 "category": "思政课程组",  
 "school": "School 1",  
 "team": "Team 1",  
 "materialPath": "/path/to/material",  
 "submissionDate": "2024-07-01T12:00:00Z",  
 "userId": 1,  
 "competitionId": 1  
 }  
]

## **2.9 评分提交（评审专家）**

URL: /api/expert/reviews

方法: POST

描述: 评审专家对提交进行评分

请求体示例:

{  
 "submissionId": 1,  
 "expertId": 2,  
 "score": 95,  
 "comment": "Great submission"  
}

响应示例:

{  
 "message": "Review submitted successfully"  
}

## **2.10 更新评分（评审专家）**

URL: /api/expert/reviews/{id}

方法: PUT

描述: 更新评分的信息

请求体示例:

{  
 "score": 98,  
 "comment": "Updated comment"  
}

响应示例:

{  
 "message": "Review updated successfully"  
}

## **2.11 删除评分（评审专家）**

URL: /api/expert/reviews/{id}

方法: DELETE

描述: 删除评分

响应示例:

{  
 "message": "Review deleted successfully"  
}