**管理员接口文档（员工信息管理）——index**

**企业员工考勤系统 - 后台管理接口文档**

**文档信息**

|  |  |
| --- | --- |
| 项目名称 | 企业员工考勤系统 - 管理后台接口 |
| 版本号 | v1.1.0 |
| 最后更新时间 | 2023-11-21 |
| 技术栈 | SpringBoot 2.3.7 + Vue.js 2.x + MyBatis |
| 协议 | HTTP/HTTPS |
| 认证方式 | Session-Based Authentication |

**1. 系统管理模块**

**1.1 退出登录接口**

**接口描述**：管理员退出系统

**请求URL**：/logout

**请求方式**：GET

**请求参数**：无

**响应类型**：重定向到登录页

**状态码说明**：

|  |  |
| --- | --- |
| 状态码 | 说明 |
| 302 | 重定向到登录页面 |

**前端调用示例**：

|  |
| --- |
| HTML <a href="/logout">退出登录</a> |

**2. 人事管理模块**

**2.1 员工信息管理接口**

**接口描述**：获取所有员工信息列表

**请求URL**：/employee/list

**请求方式**：GET

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| code | Int | 响应码 |
| data | Array | 员工列表 |
| └─ id | Long | 员工ID |
| └─ name | String | 员工姓名 |
| └─ dept | String | 部门名称 |
| └─ position | String | 职位 |
| └─ status | String | 在职状态 |

**成功响应示例**：

|  |
| --- |
| JSON {  "code": 200,  "data": [  {  "id": 1001,  "name": "张三",  "dept": "技术部",  "position": "高级工程师",  "status": "在职"  }  ] } |

**2.2 员工人事变动接口**

**接口描述**：更新员工部门或职位

**请求URL**：/employee/update

**请求方式**：POST

**Content-Type**：application/json

**请求参数**：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 描述 |
| id | Long | 是 | 员工ID |
| dept | String | 否 | 新部门 |
| position | String | 否 | 新职位 |

**请求示例**：

|  |
| --- |
| JSON {  "id": 1001,  "dept": "产品部",  "position": "产品经理" } |

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| code | Int | 响应码 |
| msg | String | 操作结果 |

**3. 部门管理模块**

**3.1 部门列表接口**

**接口描述**：获取所有部门信息

**请求URL**：/department/list

**请求方式**：GET

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| id | Long | 部门ID |
| name | String | 部门名称 |
| count | Int | 员工数量 |

**3.2 职务管理接口**

**接口描述**：获取所有职务信息

**请求URL**：/position/list

**请求方式**：GET

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| id | Long | 职务ID |
| name | String | 职务名称 |
| level | Int | 职级 |

**4. 考勤管理模块**

**4.1 每日考勤统计接口**

**接口描述**：获取指定日期的考勤统计

**请求URL**：/attendance/daily

**请求方式**：GET

**请求参数**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 描述 | 示例值 |
| date | String | 否 | 查询日期 | 2023-11-20 |

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| total | Int | 总员工数 |
| signed | Int | 已签到人数 |
| unsigned | Int | 未签到人数 |
| late | Int | 迟到人数 |
| leaveEarly | Int | 早退人数 |

**4.2 已签到员工列表接口**

**接口描述**：获取已签到员工列表

**请求URL**：/attendance/signed

**请求方式**：GET

**请求参数**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 描述 | 示例值 |
| date | String | 否 | 查询日期 | 2023-11-20 |

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| id | Long | 员工ID |
| name | String | 员工姓名 |
| dept | String | 部门 |
| signTime | String | 签到时间 |
| status | String | 签到状态 |

**4.3 未签到员工列表接口**

**接口描述**：获取未签到员工列表

**请求URL**：/attendance/unsigned

**请求方式**：GET

**请求参数**：同4.2

**响应参数**：同4.2（不含signTime字段）

**4.4 考勤统计视图接口**

**接口描述**：获取考勤统计图表数据

**请求URL**：/attendance/stats

**请求方式**：GET

**请求参数**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 描述 | 示例值 |
| type | String | 是 | 统计类型 | daily/weekly/monthly |
| date | String | 否 | 基准日期 | 2023-11-20 |

**响应参数**：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| labels | Array | 日期/部门标签 |
| data | Array | 考勤数据 |

**5. 接口安全规范**

1. **权限控制**：

* 所有管理接口需管理员权限
* 敏感操作需超级管理员权限

1. **数据验证**：

* 部门/职务名称唯一性验证
* 员工状态合法性验证

1. **操作日志**：

* 记录关键操作（人事变动、考勤修改）

**6. 错误码体系**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 4005 | 无效的员工ID | 检查员工是否存在 |
| 4006 | 部门名称已存在 | 修改部门名称 |
| 4007 | 无效的日期格式 | 使用yyyy-MM-dd格式 |
| 5002 | 数据统计异常 | 检查统计参数 |

**7. 后端实现建议**

**7.1 部门管理实现**

|  |
| --- |
| Java @RestController @RequestMapping("/department") public class DepartmentController {   @Autowired  private DepartmentService departmentService;   @GetMapping("/list")  public ResponseEntity<List<Department>> listAll() {  return ResponseEntity.ok(departmentService.findAll());  }    @PostMapping("/add")  public ResponseEntity<?> addDepartment(@Valid @RequestBody Department department) {  departmentService.save(department);  return ResponseEntity.ok().build();  } } |

**7.2 考勤统计实现**

|  |
| --- |
| Java @GetMapping("/attendance/daily") public ResponseEntity<DailyAttendanceStats> getDailyStats(  @RequestParam(required = false) String date) {    LocalDate queryDate = date != null ?   LocalDate.parse(date) : LocalDate.now();    return ResponseEntity.ok(  attendanceService.getDailyStats(queryDate)  ); } |

**8. 前端路由映射表**

|  |  |  |
| --- | --- | --- |
| 前端路由 | 后端接口 | 说明 |
| stuList | /employee/list | 员工列表 |
| updateDutyDept | /employee/update | 人事变动 |
| deptManage | /department/list | 部门管理 |
| dutyManage | /position/list | 职务管理 |
| signList | /attendance/daily | 每日考勤 |
| stuSign | /attendance/signed | 已签到员工 |
| stuNoSign | /attendance/unsigned | 未签到员工 |
| signImg | /attendance/stats | 考勤统计图表 |

**附录**

**A. 数据字典**

**员工状态**：

* 1: 在职
* 2: 休假
* 3: 离职

**考勤状态**：

* NORMAL: 正常
* LATE: 迟到
* EARLY: 早退
* ABSENT: 缺勤

**B. 数据库设计建议**

|  |
| --- |
| SQL CREATE TABLE department (  id BIGINT PRIMARY KEY AUTO\_INCREMENT,  name VARCHAR(50) NOT NULL UNIQUE,  create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP );  CREATE TABLE employee (  id BIGINT PRIMARY KEY AUTO\_INCREMENT,  name VARCHAR(20) NOT NULL,  dept\_id BIGINT REFERENCES department(id),  position\_id BIGINT REFERENCES position(id),  status TINYINT DEFAULT 1 ); |