# 高效任务管理系统task-mgr学习笔记

## 准备阶段

### 1.技术准备：

数据库：MySQL

MVC：Spring、Spring MVC和MyBatis。

前端框架：BootStrap

其他技术：JS JQuery Ajax Velocity（前端页面渲染引擎）

数据传输：json

构建工具：Maven

开发工具：Eclipse

JDK8

### 2.目标功能

功能：

任务的添加、删除、修改和查询。

数据统计（图表）

1）今天已完成任务

2）逾期任务

3）任务实际耗时、实际时间比较、拖延比例

4）待办任务

5）紧急重要任务、紧急不重要任务…列表

6）任务完成比例

7）时间管理方法计算自我评估能力

8）算法计算执行效率（日 周 月）

9）能不能自动给出建议

10）专注度、连续程度

11）当前任务

### 3.数据库表设计

表名：t\_task

字段名：

id int

content varchar(100)

desc varchar(2000)

priority byte

is\_habit byte

is\_complete byte

create\_time

update\_time

delete\_time

is\_delete（逻辑删除标志 1|0）

t\_time\_actual

id

task\_id

begin\_time\_actual

end\_time\_actual

t\_time\_expected

id

task\_id

begin\_time\_expected

end\_time\_expected

### 4.任务属性

名称、描述、计划时间（用时、开始时间和结束时间）、实际完成时间、是否是待办任务、是否是当前任务、重要紧急四象限、是否是常规任务，即习惯、

Class Task {

id Int  
name String

content String

startTime Date

endTime Date

priority Byte

isHabbit Byte

createTime Date

updateTime Date

deleteTime Date

isDelete Byte

}

## 二、实现阶段

### Tasks（任务）

#### Task：

定义任务主体Task的属性，包括：id，content（内容），desc（描述），priority（优先级），isHabit（日常任务）, isComplete（已完成）, createTime（创建时间）, updateTime（更新时间）, deleteTime（删除时间）,isDelete（已删除）, timeExpecteds（预期时间）, timeActuals（实际时间）。

public class Task {

private int id;

private String content;

private String desc;

private byte priority;

private byte isHabit;

private byte isComplete;

private Date createTime;

private Date updateTime;

private Date deleteTime;

private byte isDelete;

private List<TimeExpected> timeExpecteds;

private List<TimeActual> timeActuals;

}

#### TaskDao：

public interface TaskDao {

获取所有任务

public List<Task> getAllTasks();

根据id获取任务

public Task getTaskById(Integer id);

新增任务

public Integer insertTask(Task task);

更新任务

public void updateTask(Task task);

删除任务

public void deleteTask(Integer id);

}

#### TaskService

public interface TaskService {

查找所有任务

public abstract List<Task> getAllTasks();

根据id查找任务

public abstract Task getTaskById(Integer id);

根据id的集合查找任务

public List<Task> getTaskByIds(List<Integer> ids);

插入一条任务

public void addTask(Task task);

更新一条任务

public void updateTask(Task task);

根据id删除一条任务

public void deleteTask(Integer id);

查找已完成任务

public List<Task> getCompletedTasks();

查找逾期任务

public List<Task> getOverdueTasks();

查找待办任务

public List<Task> getTodos();

查找当前任务

public List<Task> getPresentTasks();

根据优先级查找任务

public List<Task> getTasksByPriority(Integer priority);

计算已完成任务所占的比例

public double getPercentOfCompletedTasks();

}

#### TimeController

public class TaskController {

@Autowired

private TaskService taskService;

获取所有未完成未删除任务

@RequestMapping("/getAllTasks")

public String getAllTasks(Model model) {

}

根据id查询任务

@RequestMapping("/getTaskById")

public String getTaskById(Model model, Integer id) {

}

新增任务

@RequestMapping(value = "/addTask", method = RequestMethod.GET)

public String addTask() {

}

@RequestMapping(value = "/addTask", method = RequestMethod.POST)

public String addTask(){

}

更新任务

@RequestMapping("/updateTask")

public String updateTask(Task task) {

}

删除任务（逻辑删除）

@RequestMapping("/deleteTask")

public String deleteTask(Integer id) {

}

查询已完成任务

@RequestMapping("/getCompletedTasks")

public String getCompletedTasks(Model model) {

}

逾期任务

@RequestMapping("/getOverdueTasks")

public String getOverdueTasks(Model model) {

}

待办任务

@RequestMapping("/getTodos")

public String getTodos(Model model) {

}

当前任务

@RequestMapping("/getPresentTasks")

public String getPresentTasks(Model model) {

}

根据优先级查询任务

@RequestMapping("/getTasksByPriority")

public String getTasksByPriority(Model model, Integer priority) {

}

任务完成度

@RequestMapping("/getPercentOfCompletedTasks")

public Double getPercentOfCompletedTasks() {

}

}

### TimeExpected（计划时间）

### TimeActual（实际时间）