LoadSave

---

fromString(String data): LoadSave
toString(): String

Project:LoadSave

name: String
description: String
listOfTasks: List<Task>
listOfTaskLogs: List<TaskLog>
leader: Employee

EmpType:LoadSave

name: String
isManager: boolean

Request:LoadSave

employee: Employee
reason: String
approval: enum(pending, rejected, approved)

TaskLog:LoadSave

fromTime: DateTime
toTime: DateTime
assignedEmployee: Employee
task: Task

calculateTime(): double

Task:LoadSave

code: String
title: String
description: String
assignedEmployee: Employee
taskPhase: TaskPhase
project: Project
priority: enum(high, normal, low)
creator: Employee
startDate: Date
endDate: Date
EST: double

LeaveRequest:Request

duration: int
leaveType: enum(sickleave, vacation, urgent)

User:LoadSave

-uname: String
-password: String
-utype: enum(admin, employee)

canlogin(String uname, String password): boolean

TaskPhase:LoadSave

phase: String

TimeCard:LoadSave

employee: Employee
attendance: DateTime
departure: DateTime

Employee: User

empType: EmpType
missionRequests: List<Request>
permissionRequests: List<Request>
leaveRequests: List<LeaveRequest>

Application

+userDataHandler: DataHandler<User>
+employeeDataHandler: DataHandler<Employee>
+taskDataHandler: DataHandler<Task>
+leaveRequestDataHandler: DataHandler<LeaveRequest>
+empTypeDataHandler: DataHandler<EmpType>
+taskPhaseDataHandler: DataHandler<TaskPhase>
+requestDataHandler: DataHandler<Request>
+timeCardDataHandler: DataHandler<TimeCard>
+taskLogDataHandler: DataHandler<TaskLog>
+projectDataHandler: DataHandler<Project>
+user: User
+currentmodule: Module

+main(): void
+initializeData(): void
+login(): void
+startModule(): void

DataHandler<T extends LoadSave>

filePath: String
file: File
fileWriter: FileWriter
fileReader: Scanner
list: List<T>

save(): void
update(int index, T object)
add(T object): int
delete (int index): void
get(int index): T
getLength(): int

Module

currentUser: User

startModule(): void

EmployeeModule: Module

currentEmployee: Employee

viewTimeCards(): void
createTimeCard(TimeCard timeCard): void
viewRequests(): void
makeRequest(Request request): void
makeLeaveRequest(LeaveRequest leaveRequest): void
manageRequests(): void
manageRequest(Request request): void

TaskModule: Module

currentEmployee: Employee
project: Project

viewTasks(): void
viewTask(Task task): void
viewTasklogs(Task task): void
viewTasklog(TaskLog taskLog): void
manageTasks(): void
createTaskLog(Task task, DateTime fromTime, DateTime toTime):TaskLog

AdminModule: Module

manageUsers(): void
manageEmployees(): void
mangeProjects(): void
mangeEmpType(): void
manageTaskPhases(): void