

GUIController
commandHandler: CommandHandler = CommandHandler.getInstance() LOGGER: Logger = LogManager.getLogManager().getLogger(getClass().getName()) dataAccess: DataAccess = DataAccess.getInstance() scheduleModelHolder: ScheduleModelHolder = ScheduleModelHolder.getInstance() serialVersionUID: long = 8156760488963380338L
run(): void wedo(): void +getAmountOfTeachers(): String +getAmountOfPedagogicAssistants(): String +getAmountOfEmployees(): String +getAmountOfLocations(): String +getAmountOfRooms(): String +refresh(entity: Entity): void

MeetingTypeController
LOGGER: Logger = LogManager.getLogManager().getLogger(MeetingTypeController.class.getName()) meetingType: MeetingType dataAccess: DataAccess = DataAccess.getInstance() commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() meetingTypeMemento: IMemento rooms: List<Room> hideDeletionDialog: boolean hide: boolean +init(): void +getMeetingTypes(): List<MeetingType> +getAIRooms(): List<Room> +deleteMeetingType(): void +doBeforeEdit(): void +editMeetingType(): void +addMeetingType(): void +refresh(): void +getTypicalDuration(): int

RoomController
room: Room location: Location roomMemento: IMemento locationMemento: IMemento commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() dataAccess: DataAccess = DataAccess.getInstance() hideDeletionDialog: boolean hide: boolean +init(): void +getRoom(): List<Room> +generateRoomMemento(): void +addRoom(): void +editRoom(): void +deleteRoom(): void +doBeforeAdd(): void

ClassTeamController
LOGGER: Logger = LogManager.getLogManager().getLogger(ClassTeamController.class.getName()) dataAccess: DataAccess = DataAccess.getInstance() selectedClassTeam: ClassTeam classTeamMemento: IMemento selectedEmployees: List<Employee> = new ArrayList<Employee>() selectedSchoolClasses: List<SchoolClass> = new ArrayList<SchoolClass>() hideDeletionDialog: boolean hide: boolean +init(): void +addClassTeam(): void +editClassTeam(): void +deleteClassTeam(): void +doBeforeEdit(): void +doBeforeAdd(): void +getClassTeams(): List<ClassTeam> +getTeachers(): List<Teacher> +getPedagogicAssistants(): List<PedagogicAssistant> +getValidSchoolClasses(): List<SchoolClass> +getDisabledTeacherSelectItem(): SelectItem +getDisabledPedagogicAssistantSelectItem(): SelectItem

ConfigController
MAX SPINNER_VALUE: int = 1440 MIN SPINNER_VALUE: int = 1 LOWER_CASE: String = ConfigIdentifier.LOWER_CASE UPPER_CASE: String = ConfigIdentifier.UPPER_CASE BOTH_CASES: String = ConfigIdentifier.BOTH_CASES MAX DAY_VALUE: int = 30 +DISABLED_BACKUPS: String = "disabled" +MINUTE_BACKUPS: String = "minutes" +DAILY_BACKUPS: String = "daily" <<Property>> weekdays: List<Weekday> = Arrays.asList(Weekday.values()) <<Property>> selectedWeekdays: List<Weekday> weekdayStartTime: Date weekdayEndTime: Date +d: SimpleDateFormat = new SimpleDateFormat("HH:mm") timetableGrid: int teacherSettlement: int paSettlement: int typicalActivityDuration: int identifierCase: String backupIntervalValue: int backupInterval: int backupStartTime: Date intervalType: String selectedDayValue: int = 1 lessonStressingColor: ColorEnum lessonRelaxingColor: ColorEnum meetingColor: ColorEnum compoundLessonColor: ColorEnum pauseColor: ColorEnum +init(): void +updateProperties(): void +deleteAllActivities(): void +resetDialogs(): void selectBackupSettings(): void +updateBackupInterval(): boolean +validateBackupSettings(): boolean +getMaxSpinnerValue(): int +getMinSpinnerValue(): int +getLowerCase(): String +getUpperCase(): String +getBothCases(): String +getMaxDayValue(): int +getDisableBackup(): String +getMinuteBackups(): String +getDailyBackups(): String +getAllColors(): List<ColorEnum>

AcademicYearAndClassController
LOGGER: Logger = LogManager.getLogManager().getLogger(AcademicYearAndClassController.class.getName()) INITIAL_GENERATOR_SIZE: int = 1 MAX_GENERATOR_SIZE: int = 20 dataAccess: DataAccess = DataAccess.getInstance() commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() academicYear: AcademicYear schoolClasses: SchoolClass location: Location academicYearMemento: IMemento schoolClassMemento: IMemento generatorSize: int hideDeletionDialog: boolean hide: boolean +init(): void +getAcademicYears(): List<AcademicYear> +doBeforeAdd(): void +getExistAcademicYears(): Boolean +getSchoolClasses(): List<SchoolClass> +generateAcademicYears(): void +addAcademicYear(): void +getMaxYear(): int +getValidIdentifiers(): List<Character> +addSchoolClass(): void +doBeforeEdit(): void +getSchoolClasses(): void +deleteSchoolClass(): void +getMaxGeneratorSize(): int

BackupController
backup: File hasName: Boolean = true backupName: String +getAllBackups(): List<File> +doBackup(): void +doRestore(): void +openBackupLocation(): void +getFileExists(): boolean

LessonTypeController
LOGGER: Logger = LogManager.getLogManager().getLogger(LessonTypeController.class.getName()) lessonType: LessonType dataAccess: DataAccess = DataAccess.getInstance() commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() lessonTypeMemento: IMemento rooms: List<Room> hideDeletionDialog: boolean hide: boolean +isRelaxing: Boolean +init(): void +getLessonTypes(): List<LessonType> +getAIRoom(): List<Room> +deleteLessonType(): void +doBeforeEdit(): void +addLessonType(): void +refresh(): void +getTypicalDuration(): int

TeacherController
LOGGER: Logger = LogManager.getLogManager().getLogger(TeacherController.class.getName()) dataAccess: DataAccess = DataAccess.getInstance() commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() teacherMemento: IMemento activityTypes: DualListModel<ActivityType> hideDeletionDialog: boolean hide: boolean hourlySettlement: int +init(): void +onTransfer(event: TransferEvent): void +checkHoursPerWeek(event: ValueChangeEvent): void +refresh(): void +getTeachers(): List<Teacher> +addTeacherDialog(): void +generateTeacherMemento(): void +editTeacher(): void +deleteTeacher(): void +addTeacher(): void

LocationController
dataAccess: DataAccess = DataAccess.getInstance() LOGGER: Logger = LogManager.getLogManager().getLogger(LocationController.class.getName()) commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() locationMemento: IMemento location: Location hideDeletionDialog: boolean hide: boolean firstLocation: Location secondLocation: Location travelTimeValue: String selectedTravelTimeEdge: Edge +init(): void +getLocations(): List<Location> +generateLocationMemento(): void +editLocation(): void +deleteLocation(): void +addLocation(): void +addTravelTime(): void +deleteTravelTime(): void +getTravelTimeEdges(): List<Edge>

PedagogicAssistantController
LOGGER: Logger = LogManager.getLogManager().getLogger(PedagogicAssistantController.class.getName()) dataAccess: DataAccess = DataAccess.getInstance() commandHandler: CommandHandler = CommandHandler.getInstance() commandCreator: CommandCreator = CommandCreator.getInstance() pedagogicAssistantMemento: IMemento pedagogicAssistantMemento: IMemento activityTypes: DualListModel<ActivityType> hideDeletionDialog: boolean hide: boolean hourlySettlement: int +init(): void +onTransfer(event: TransferEvent): void +checkHoursPerWeek(event: ValueChangeEvent): void +refresh(): void +getPedagogicAssistants(): List<PedagogicAssistant> +addPedagogicAssistantDialog(): void +generatePedagogicAssistantMemento(): void +editPedagogicAssistant(): void +deletePedagogicAssistant(): void +addPedagogicAssistant(): void

manage