

DataBaseServletListener
ontextDestroyed(arg0 : ServletContextEvent) : void
ontextInitialized(arg0 : ServletContextEvent) : void

-INSTANCE : DataAccess = new DataAccess() -em : EntityManager = DataBase.getInstance().getEntityManager() -DataAccess() +getInstance() : DataAccess +update(o : Observable, arg : Object) : void +persist(object : Entity) : void Adelete(phiect : Entity) : unid refresh(entity : Entity) : void +getAllAcademicYears() : List<AcademicYear> +getOneAcademicYear(year: int): AcademicYear +getOneAcademicYear(schoolclass: Schoolclass): AcademicYear +getAllSchoolclasses() : List<Schoolclass> +getAllSchoolclassesWithoutClassteam() : List<Schoolclass> +getAllSchoolclasses(teacher : Teacher) : List<Schoolclass> rgetAllSchoolclasses(employee : Employee) : List<Schoolclass> +getOneSchoolclass(room : Room) : Schoolclass ademicYear : int, identifier : char) : Schoolclass +getAllEmployees(activityType : ActivityType) : List<Employee> +getAllPAs(): List<PedagogicAssistant> +getAllTeachers() : List<Teacher> +getOneEmployee(symbol : String) : Employee +searchEmployees(searchSymbol : String) : List<Employee> +searchTeachers(searchSymbol : String) : List<Teacher> +searchPAs(searchSymbol : String) : List<PedagogicAssistant> getAllLocations() : List<Loca +getOneLocation(room : Room) : Location +getOneLocation(name : String) : Location +getAllRooms(purpose : String) : List<Room> +getOneRoom(locationName : String, roomName : String) : Room +getRoomPurposes() : List<String> etAllActivityTypes() : List<ActivityType> +getAllActivityTypes(room : Room) : List<ActivityType> +getAllMeetingTypes() : List<MeetingType> getAllLessonTypes() : List<LessonType> +getAllProjectTypes() : List<ProjectType> +getOneActivityType(name : String) : ActivityType ontainsSpecialRooms(type : ActivityType) : boolean +getAllActivities(): List<Activity> +getAllActivitiesNotBetween(startTime : Date, endTime : Date) : List<Activity> +getAllActivities(employee: Employee, refreshCache: boolean): List<Activity> +getAllActivities(object: lActivityObject, refreshCache: boolean): List<Activity> +getAllActivities(schoolclass : Schoolclass, refreshCache : boolean) : List<Activity> +getAllActivities(room : Room, refreshCache : boolean) : List<Activity +getAllActivities(weekday : Weekday) : List<Activity> +getAllActivities(employee : Employee, weekday : Weekday) : List<Activity> getAllActivities(employee : Employee, timePeriod : TimePeriod) : List<Activity>
+getAllActivities(schoolclass : Schoolclass, timePeriod : TimePeriod) : List<Activity> +getAllActivities(room : Room, timePeriod : TimePeriod) : List<Activity> +getAllCompoundLessons(lessonType ; LessonType) ; List<CompoundLesson> +getAllLessons(lessonType : LessonType) : List<Lesson aetAllMeetings(meetingType : MeetingType) : List<Meeting> rgetEmployeeTimePeriods(employee: Employee): List<EmployeeTimePeriods>
+sumLessonDuration(employee: Employee, schoolclass: Schoolclass, lessonType: LessonType): Long +getFirstActivityBefore(employee : Employee, timePeriod : TimePeriod, location : Location) : Activity +getFirstActivityAfter(employee : Employee, timePeriod : TimePeriod, location : Location) : Activity +getAllActivitiesBefore(employee : Employee, timePeriod : TimePeriod) : List<Activity> +getAllActivitiesAfter(employee : Employee, timePeriod : TimePeriod) : List<Activity> +getFirstActivityBefore(schoolclass : Schoolclass, timePeriod : TimePeriod, location : Location) : Activity LastFirstActivityAfter(schoolclass : Schoolclass timePeriod : TimePeriod (Incation : Location) : Activity +getAllClassteams(): List<Classteam> rgetAllClassteams(employee : Employee) : List<Classteam> rgetOneClassteam(schoolclass : Schoolclass) : Classteam +getByld(clazz : Class<E>, id : Long) : E +getTravelTimeList() : TravelTimeList +getUsedChars(academicYear : AcademicYear) : List<Character> -getAll(clazz : Class<? extends Serializable>, orderBy : String) : List<? extends Serializable>
-searchEmployees(clazz : Class<? extends Employee>, searchSymbol : String) : List<? extends Serializable ▲ has DataBase +DB\_URL : String = "idbc:h2:-/WOYM/timetable.db/timetable" +DB\_LOCATION\_PATH : String = System + File.separator + "WOYM" + File.separator + "timetable.db" +DB\_BACKUP\_LOCATION : String = System .getProperty("user.home") + File.separator + "WOYM" + File.separator USER : String = "timetable" -PASSWORD : String = "timetable" -LOGGER : Logger = LogManager.getLogger("DataBase") -entityManager : EntityManager -INSTANCE : DataBase = new DataBase() serialVersionUID : long = 6237080407072 +getInstance(): DataBase
-getEntityManager(): EntityManager
+backup(backupName: String): String restore(filePath : String) : void -setUp(): void -backup() : String -checkZip(filePath : String) : boolean

-deleteFolder(folder : File) : void -copyDirectory(src : File, dest : File) : void

DataAccess