# Sevice Layer

**ServiceFactory**
- + bs: BoardService
- + us: UserService
- + ts: TaskService
- - uc: UserController
- - bc: BoardController
---
- + ServiceFactory(): void

**UserService**
- - uc: UserController
---
- + Register(email: string, password: string): string
- + Login(email: string, password: string): string
- + Logout(email: string): string
- + ChangePassword(email: string, newPassword: string): string

**BoardService**
- - bc: BoardController
- - uc: UserController
---
- + AddBoard(email: string, boardName: string): string
- + RemoveBoard(email: string, boardName: string): string
- + LimitColumn(email: string, boardName: string, limit: int, columnName: string): string
- + GetColumnLimit(email: string, boardName: string, columnName: string): string
- + GetColumnTasks(email: string, boardName: string, columnName: string): string
- + AddTask(email: string, boardName: string, title: string, description: string, dueDate: DateTime): string
- + MoveTask(email: string, boardName: string, currentColumn: string, nextColumn: string, taskId: int): string
- + GetTasksInProgress(email: string): string
- + JoinBoard(email: string, boardID: int): string
- + LeaveBoard(email: string, boardID: int): string
- + TransferOwnership(currentOwnerEmail: string, newOwnerEmail: string, boardID: int): string
- + LoadData( ): string
- + DeleteAllData( ): string
- + GetUserBoards(email: string): string
- + GetBoardName( boardID:int): string

**GradingService**
- - ordinalToName: Dictionary<int, string>
- - nameToOrdinal: Dictionary<string, int>
---
- + GetColumnOrdinal(columnName: string): int
- + GetColumnName(columnOrdinal: int): string

**Response**
- + ErrorMessage: string
- + ReturnValue: Object

**TaskToSend**
- + Id: int
- + CreationTime: DateTime
- + Title: string
- + Description: string
- + DueDate: DateTime

**TaskService**
- - bc: BoardController
- - uc: UserController
---
- + UpdateTaskDueDate(email: string, boardName: string, currentColumn: string, taskId: int, dueDate: DateTime): string
- + UpdateTaskTitle(email: string, boardName: string, currentColumn: string , taskId: int, title: string): string
- + UpdateTaskDescription(email: string, boardName: string, currentColumn: string, taskId: int, description: string): string
- + AssignTask(email: string, boardName: string, currentColumn: string, taskID: int, emailAssignee: string): string

# Business Layer

**UserController**
- - users: Dictionary<string, User>
- - UserMapper: UserMapper
---
- + GetUser(email: string): User
- + Register(email: string, password: string): void
- + UserExists(email: string): bool
- + IsLoggedIn(email: string): bool
- + loadData(): void
- + DeleteAllData(): void
- + IsLoggedIn(email: string): bool
- + DeleteUser(email: string): void

**BoardController**
- - boards: Dictionary<string,Dictionary<string,Board>>
- - boardsById: Dictionary<int, Board>
- - boardMapper: BoardMapper
- - LoadMapper: LoadMapper
- - idSerializer: int
---
- + GetBoard(email: string, boardName: string): Board
- + IsMemberOfBoard(email: string, boardName: string): bool
- + AddBoard(email: string, boardName: string): void
- + RemoveBoard(email: string, boardName: string): void
- + GetTasksInProgress(email: string): List<Task>
- + TransferOwnership(currentOwnerEmail: string, newOwnerEmail: string, boardID: int): void
- + JoinBoard(email: string, boardID: int): void
- + LeaveBoard(email, boardID: int): void
- + loadData(): void
- + DeleteAllData(): void
- + GetBoardById(id: int): Board
- + GetBoard(email:string, boardName:string): Board
- + IsMemberOfBoard(email:string, boardName:string): bool
- + GetUserBoards(email:string,): List<int>
- + GetBoardName(boardID:int): string

**User**
- + EMAIL: string
- - password: string
- - loggedIn: bool
- - UserDTO : UserDTO
---
- + ValidatePassword(password: string): bool
- + ValidateEmail(email: string): bool
- +BuildUser(email:string, password:string): User
- + Login(password: string): void
- + Logout(): void

**Board**
- + NAME: string
- + ID: int
- - columns: Dictionary<string, Column>
- - idSerializer: int
- - owner: string
- - boardDTO : BoardDTO
---
- + BuildStandardBoard(name: string): Board
- + ColumnExists(column: string): bool
- + GetColumn( columnName: string): Column
- + GetColumnTasks( columnName: string): List<Task>
- + AddColumn(column: string): void
- + RetriveColumn(column: Column): void
- + AddTask(title: string, dueDate:DateTime, description: string, columnName: string): void
- + MoveTask(currentColumn: string, nextColumn: string, taskId: int): void
- + TransferOwnership(currentOwnerEmail : string, newOwnerEmail: string)

**Task**
- + id: int
- - CREATIONTIME: DateTime
- - title: string
- - description: string
- - dueDate: DateTime
- - assigneeEmail: string
- - taskDTO: TaskDTO
---
- +BuildTask(dueDate:DateTime,title:string,description:string,id: int, boardID:int,columnName:string)
- + ValidateTask(title: string, dueDate: DateTime, description: string): bool
- + UpdateTaskDueDate(email:string, dueDate: DateTime): void
- + UpdateTaskTitle(email:string, title: string): void
- + UpdateTaskDescription(email:string, description: string): void
- + AssignTask(email:string, emailAssignee: string): void
- + UnassignTask(): void

**Column**
- + NAME: string
- - maxTasks: int
- - NO_LIMIT: int
- - tasksList: Dictionary<int, Task>
- - columnDTO: ColumnDTO
---
- + BuildColumn(name: string, boardID:int, limit:int): Column
- + GetTasks(): Dictionary<int, Task>
- + GetTask( taskId: int): Task
- + GetColumnLimit(): int
- + SetLimit(limit: int): void
- + IsFull(): bool
- + ContainsTask(taskId: int): bool
- + AddTask(task: Task): void
- + AddTask(title: string, dueDate: DateTime, description: string, int idSerializer, int:boardID) :void
- + DeleteTask(taskId: int): Task
- + RetriveTask(task: Task): void

*Use* (labels on associations between Task/Column/Board)

**Changes**
-JoinBoard:
added to BoardService in the ServiceLayer
added to BoardController in BusinessLayer

-LeaveBoard:
added to BoardService in the ServiceLayer
added to BoardController in BusinessLayer

-TransferOwnership:
added to BoardService in the ServiceLayer
added to BoardController in BusinessLayer

-AssignTask:
added to TaskService in the ServiceLayer
added to Task in BusinessLayer

# Data-Access Layer

**UserDTO**
- - _EMAIL: string (PK)
- - password: string
---
- + Persist(): void

**BoardDTO**
- - id: int (PK)
- + name: string
- - ownerEmail: string
---
- + Persist(): void

**ColumnDTO**
- - boardID: int (FK of boards, PK)
- - columnName: string (PK)
- - maxTasks: int
---
- + Persist(): void

**TaskDTO**
- - taskId: int (PK)
- - boardID: int (PK)
- - columnName: string
- - creationTime: DateTime
- - assigneeEmail: string
- - title: string
- - description: string
- - dueDate: DateTime
---
- + Persist(): void

**DTO**
- # Mapper: mapper

**UserToBoardDTO**
- + email: string (FK of users, PK)
- - boardID: int (FK od boards, PK)

**LoadMapper**
- + taskMapper : TaskMapper
- + columnMapper : ColumnMapper
- + userToBoardMapper : UserToBoardMapper
---
- + SelectAllUserToBoard() : List<UserToBoardDTO>
- + SelectAllColumns() : List<ColumnDTO>
- + DeleteAllData() : void
- + SelectAllTasks() : List<TaskDTO>

**BoardMapper**
- - MessageTableName: string
- - userToBoard: UserToBoardMapper
---
- + SelectAllBoards():List<BoardDTO>
- + Update( id:long,attributeName: string ,attributeValue: string) :bool
- + Delete(board: BoardDTO): bool
- + JoinBoard(email: string, boardID: int): bool
- + LeaveBoard(email: string, boardID: int): bool
- # ConvertReaderToObject(reader:SQLiteDataReader ): BoardDTO

**Mapper**
- # _CONNECTIONSTRING: string
- + _TABLENAME: string
---
- + Select( ): List<DTO>
- + DeleteAllData( ): bool
- + ConvertReaderToObject(reader: SqliteDataReader) : DTO

**ColumnMapper**
- - TableName : string
---
- + SelectAllColumns() : List< ColumnDTO>
- + Update(boardid:long, columnName: string, attributeName: string, attributeValue: long) : boolean
- + Select( ): List<DTO>
- + Delete( DTOObj: ColumnDTO): boolean
- + Insert( column: ColumnDTO): boolean
- + ConvertReaderToObject(reader:SqliteDataReader) : ColumnDTO

**UserMapper**
- - TableName: string
---
- + SelectAllUser() : List< UserDTO>
- + Insert( user: UserDTO): bool
- + ConvertReaderToObject(reader:SqliteDataReader) : UserDTO

**TaskMapper**
- + SelectAllTasks() : List< TaskDTO>
- + Update(taskid:long, boardid:long, attributeName: string, attributeValue: string) : boolean
- + Update(taskid:long, boardid:long, attributeName: string, attributeValue: DateTime) : boolean
- + Select( ): List<DTO>
- + Delete( DTOObj: TaskDTO): bool
- + Insert( task: TaskDTO): bool
- # ConvertReaderToObject(reader:SqliteDataReader) : TaskDTO

**UserToBoardMapper**
- - TableName: string
---
- + Insert( email:string, boardID:int): bool
- + Delete( email:string, boardID: int): bool
- # ConvertReaderToObject(reader:SqliteDataReader) : UserToBoardDTO