## **UserService**

- userM UserManagement;
- log llog;
- + createNewUser(String email, String password): json
- + login(String email, String password): json
- + logout(String email) : json
- + isLoggedIn(String email) : json

# Response<T>

- + ErrorMessage string?;
- + ReturnValue T?;
- + Response(string? errorMessage, T? returnValue): Response

## **JsonController**

- + js JsonSerializerOptions;
- + Serialize<T>(T o): string
- + DeSerialize<T> (string os) : T

# ResponseInt

+ ReturnValue int;

## **BoardService**

- bm BoardManagement;
- log ILog;
- + addBoard(string toAddTitle, string userEmail): json
- + removeBoard(string userEmail, string toRemove) : json
- $+\ add Task (string\ user Email,\ string\ board Name,\ string\ title,\ string\ description,\ Date Time\ due Date): json$
- $+\ move Task (string\ user Email,\ string\ board Name,\ int\ column Ordinal,\ int\ task Id):\ js on$
- + editTask(string userEmail, string boardName, int columnOrdinal, int taskId, string? newTitle, string? newDesc, DateTime? newDueDate, int numAct): json
- $+\ set Capacity (string\ user Email, string\ board Name, int\ limit,\ int\ column Index): json$
- + getCapacity(string userEmail, string boardName,int columnIndex): json
- + getColumnName(string userEmail, string boardName, int columnIndex): json
- + ListProgressTasks(string client) : json
- + GetColumn(string email, string boardName, int columnOrdinal) : json

# **Business Layer**

#### Board

- + boardName string;
- backlogList List<Task>;
- inProgressList List<Task>;
- doneList List<Task>:
- email string;
- capacities int∏;
- log ILog;
- + getCapacity(int columnIndex): int
- + getColumn(int columnOrdinal): List<Task>
- + getTask(int taskId): Task
- + getTask(int columnOrdinal, int taskId): Task
- + setCapacity(int limit, int columnIndex): void
- + addTask(string title, string desc, DateTime dueDate, int taskId) : void
- + moveTask(int columnOrdinal, int taskId): void
- + editTask(int columnOrdinal, int taskId, string? newTitle, string? newDesc, DateTime? newDueDate, int numAct): void

# BoardManagement

- boardList List<Board>;
- um UserManagement:
- taskldcounter int:
- log ILog;
- + getBoard(User user, string boardName): Board
- + getColumnName(string userEmail, string boardName, int column): string
- + getCapacity(string userEmail, string boardName, int columnIndex): int
- + GetColumn(string email, string boardName, int columnOrdinal): List<Task>
- + setCapacity(string userEmail, string boardName, int limit, int columnIndex): void
- + addBoard(string toAddTitle, string userEmail): void
- + boardCheck(string boardName): bool
- + removeBoard(string boardNameToRemove, string userEmail): void
- + boardExists(string boardName, string userEmail): bool
- + addTask(string userEmail, string boardName, string title, string description, DateTime dueDate): void
- + moveTask(string userEmail, string boardName, int columnOrdinal, int taskId): void
- + editTask(string userEmail, string boardName, int columnOrdinal, int taskId, string? newTitle, string? newDesc, DateTime? newDueDate): void
- + listProgressTasks(string userEmail) : List<Task>

## UserManagment

- \_users List<User>;
- log ILog;
- emailCheck EmailAddressAttribute;
- + GetUser(string email): User
- + createNewUser(string email, string password):bool
- + logIn (string email, string password): bool
- + logOut (string email): bool
- + exist (string email): bool
- + checks(string email): void
- + isValidEmail (string email): bool
- + passwordCheck (string password); bool
- + isLoggedIn (string email): bool

## User

- + email string;
- password string;
- loggedIn bool;
- log ILog;
- + isLoggedIn(): bool
- + logIn(): void
- + logOut(): void
- + passwordMatch (string pass): bool

#### Task

- -title string;
- -description string;
- -creationTime DateTime;
- -dueDate DateTime;
- -columnIndex int:
- -taskld int;
- log ILog;
- \_ -
- + editTitle(string newTitle) : void
- + editDescription(string newDesc) : void
- + editDueDate(DateTime newDueDate): void