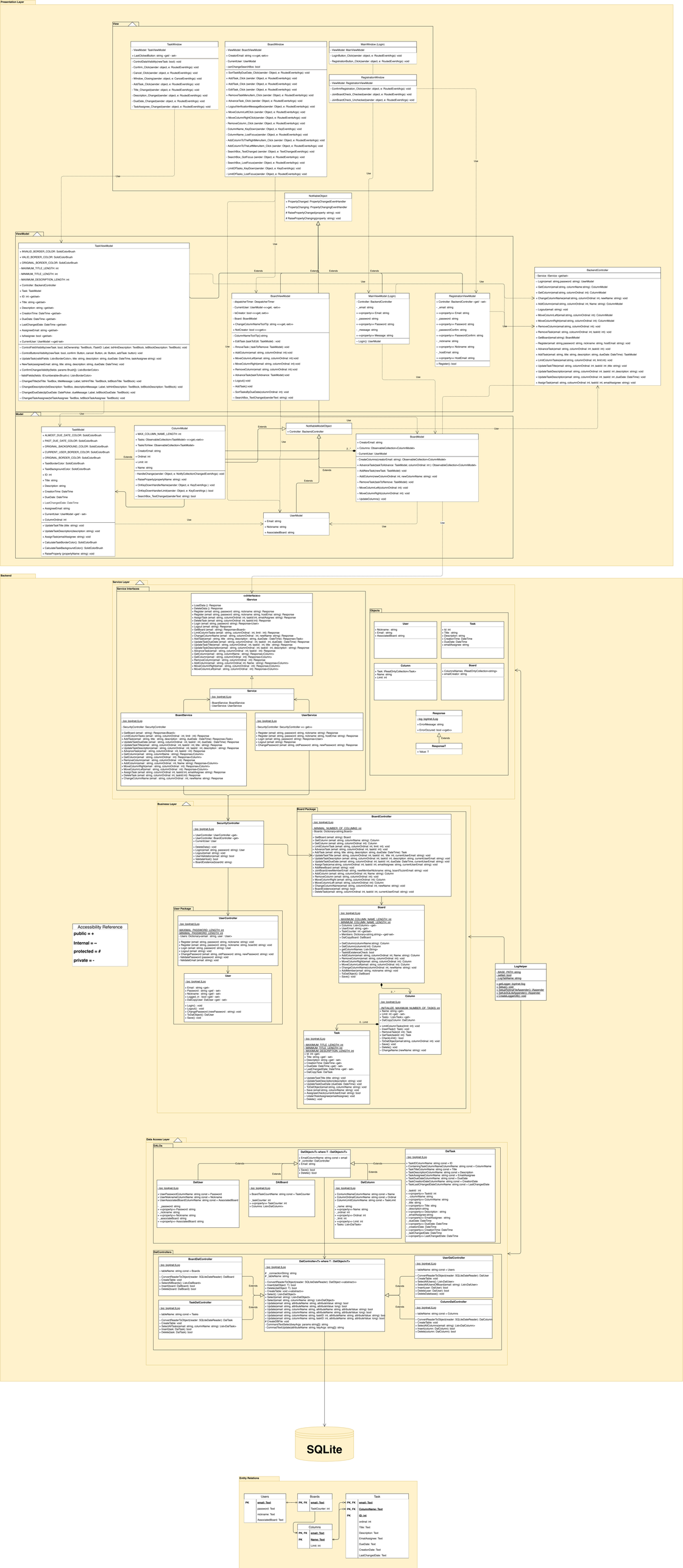
Presentation Layer
View
TaskWindow
- ViewModel: TaskViewModel
+ LastClickedButton: string <get; set>
- ControlDateVisibilityOnewTask: bool: void
- Confirm_Click(sender: object, e: RoutedEventArgs): void
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Window_Closing(sender: object, e: CancelEventArgs): void
- AddTask_Click(sender: object, e: RoutedEventArgs): void
- Title_Changed(sender: object, e: RoutedEventArgs): void
- Description_Changed(sender: object, e: RoutedEventArgs): void
- DueDate_Changed(sender: object, e: RoutedEventArgs): void
- TaskAssignee_Changed(sender: object, e: RoutedEventArgs): void
BoardWindow
- ViewModel: BoardViewModel
+ CreatorEmail: string <<get, set>>
- CurrentUser: UserModel
- canChangeSearchBox: bool
+ SortTaskByDueDate_Click(sender: Object, e: RoutedEventArgs): void
+ AddTask_Click (sender: Object, e: RoutedEventArgs): void
+ AddTask_Click (sender: Object, e: RoutedEventArgs): void
+ EditTask_Click (sender: Object, e: RoutedEventArgs): void
+ RemoveTaskMenuItem_Click (sender: Object, e: RoutedEventArgs): void
+ AdvanceTask_Click (sender: Object, e: RoutedEventArgs): void
+ LogoutNotificationMessageBox(sender: Object, e: RoutedEventArgs): void
+ MoveColumnLeftClick (sender: Object, e: RoutedEventArgs): void
+ MoveColumnRightClick(sender: Object, e: RoutedEventArgs): void
- RemoveColumn_Click (sender: Object, e: RoutedEventArgs): void
- ColumnName_KeyDown(sender: Object, e: KeyEventArgs): void
- ColumnName_LostFocus(sender: Object, e: RoutedEventArgs): void
- AddColumnToTheRightMenuItem_Click (sender: Object, e: RoutedEventArgs): void
- AddColumnToTheLeftMenuItem_Click (sender: Object, e: RoutedEventArgs): void
- SearchBox_TextChanged (sender: Object, e: TextChangedEventArgs): void
- SearchBox_GotFocus (sender: Object, e: RoutedEventArgs): void
- SearchBox_LostFocus(sender: Object, e: RoutedEventArgs): void
- LimitOfTasks_KeyDown(sender: Object, e: RoutedEventArgs): void
- LimitOfTasks_LostFocus(sender: Object, e: RoutedEventArgs): void
MainWindow (Login)
- ViewModel: MainViewModel
- LogoBtn_Click(sender: object, e: RoutedEventArgs): void
- RegistrationBtn_Click(sender: object, e: RoutedEventArgs): void
RegistrationWindow
- ViewModel: RegistrationViewModel
- ConfirmRegistration_Click(sender: object, e: RoutedEventArgs): void
- JoinBoardCheck_Checked(sender: object, e: RoutedEventArgs): void
- JoinBoardCheck_Unchecked(sender: object, e: RoutedEventArgs): void
Use
NotifiableObject
+ PropertyChanged: PropertyChangedEventHandler
+ PropertyChanging: PropertyChangingEventHandler
# RaisePropertyChanges(propertyc: string): void
# RaisePropertyChanging(propertyc: string): void
ViewModel
TaskViewModel
+ INVALID_BORDER_COLOR: SolidColorBrush
+ VALID_BORDER_COLOR: SolidColorBrush
+ ORIGINAL_BORDER_COLOR: SolidColorBrush
- MAXIMUM_TITLE_LENGTH: int
- MINIMUM_TITLE_LENGTH: int
- MAXIMUM_DESCRIPTION_LENGTH: int
+ Controller: BackendController
+ Task: TaskModel
+ ID: int <get;set>
+ Title: string <get;set>
+ Description: string <get;set>
+ CreationTime: DateTime <get;set>
+ DueDate: DateTime <get;set>
+ LastChangedDate: DateTime <get;set>
+ AssigneeEmail: string <get;set>
+ IsAssignee: bool <get;set>
+ CurrentUser: UserModel <<get>set>
- ControlFieldVisibilityOnewTask: bool, bsOwnership: TextBlock, rTaskID: Label, txtHintDescription: TextBlock, txtBlockDescription: TextBlock): void
- ControlButtonsVisibilityonewTask: bool, confirm: Button, cancel: Button, ok: Button, addTask: button): void
+ UpdateTaskcolorFields: LinrBorderColor=, title: string, description: string, dueDate: DateTime, taskAssignee string): void
+ NewTaskcolorgneEmail: string, title: string, description: string, dueDate: DateTime): void
+ ConfirmChangesValidityfields: params Brush[]: LinrBorderColor=
- ValidFieldchecks: IEnumerable<Brush>: LinrBorderColor=
- ChangedTitleOnTitle: TextBox, titleMessage: Label, txtHintTitle: TextBlock, txtBlockTitle: TextBlock): void
- ChangedDescriptionOnDescription: TextBox, descriptionMessage: Label, txtHintDescription: TextBlock, txtBlockDescription: TextBlock): void
- ChangedDueDateOnDueDate: DatePicker, dueMessage: Label, txtBlockDueDate: TextBlock): void
- ChangedTaskAssigneeOnTaskAssignee: TextBox, txtBlockTaskAssignee: TextBlock): void
Extends
BoardViewModel
- dispatcherTimer: DispatcherTimer
- CurrentUser: UserModel <<get; set>>
- IsCreator: bool <<get; set>>
- Board: BoardModel
+ ChangeColumnNameToolTip: string <<get; set>>
+ NoCreator: bool <<get>>
- ColumnNameToolTip1: string
+ EditTask (task:TaskEdit: TaskModel): void
- RenameTask (taskToRename: TaskModel): void
+ AddColumn(email: string, columnOrdinal: int):void
+ MoveColumnLeft(email: string, columnOrdinal: int):void
+ MoveColumnRight(email: string, columnOrdinal: int):void
+ RemoveColumn(email: string, columnOrdinal: int):void
+ AdvanceTask(task:TaskAdvance: TaskModel):void
+ LogoUt():void
- AddTask():void
- SortTaskByDueDate(columnOrdinal: int): void
- SearchBox_TextChanged(sender:Text: string): void
MainViewModel (Login)
- Controller: BackendController
- _email: string
+ <<property>> Email: string
- _password: string
+ <<property>> Password: string
- _message: string
+ <<property>> Message: string
- Login(): UserModel
RegistrationViewModel
+ Controller: BackendController <get> set>
- _email: string
+ <<property>> Email: string
- _password: string
+ <<property>> Password: string
- _passwordConfirm: string
+ <<property>> PasswordConfirm: string
- _nickname: string
+ <<property>> Nickname: string
- _hostEmail: string
+ <<property>> HostEmail: string
+ Register(): bool
BackendController
- Service: IService <getset>
- LoginUser(email: string,password: string): UserModel
+ GetColumn(email: string, columnName string): ColumnModel
+ GetColumn(email:string, columnOrdinal int): ColumnModel
+ ChangeColumnName(email:string, columnOrdinal: int, newName: string): void
+ AddColumn(email:string, columnOrdinal: int, Name: string): ColumnModel
- LogoutEmail: string: void
+ MoveColumnLeft(email:string, columnOrdinal int): ColumnModel
+ MoveColumnRight(email:string, columnOrdinal: int): ColumnModel
+ RemoveColumn(email:string, columnOrdinal: int): void
+ RemoveTask(email: string, columnOrdinal: int, taskid: int): void
+ GetBoard(email:string): BoardModel
- Register(email: string,password: string, nickname: string, hostEmail: string): void
+ AdvanceTask(email: string, columnOrdinal: int taskid: int): void
+ AddTask(email: string, title: string, description: string, dueDate: DateTime): TaskModel
+ LimitColumnTasks(email: string, columnOrdinal: int, limit:int): void
+ UpdateTaskTitle(email: string, columnOrdinal: int, taskid: int, title: string): void
+ UpdateTaskDescription(email: string, columnOrdinal: int, taskid: int, description: string): Response
+ UpdateTaskDueDate(email: string, columnOrdinal: int, taskid: int, dueDate: DateTime): Response
+ AssignTaskUserEmail(email: string, columnOrdinal: int, taskid: int, emailAssigner: string, currentUserEmail: string): void
+ AddNewBoard (email: string): void
+ JoinBoardUsingMemberEmail(email: string, newMemberNickname: string, boardTitleInEmail: string): void
+ RemoveBoard(email: string, boardTitleInEmail: string): void
+ RemoveColumn (email: string, columnOrdinal: int): void
+ MoveColumnRight (email: string, columnOrdinal: int): Column
+ MoveColumnLeft(email: string, columnOrdinal: int): Column
+ ChangeColumnName(email: string, columnOrdinal: int, newName: string): void
+ BoardExistence(email: string): bool
- DeleteTask(email: string, columnOrdinal: int, taskid: int, currentUserEmail: string): void
Model
TaskModel
+ ALMOST_DUE_DATE_COLOR: SolidColorBrush
+ PAST_DUE_DATE_COLOR: SolidColorBrush
+ ORIGINAL_BACKGROUND_COLOR: SolidColorBrush
+ CURRENT_USER_BORDER_COLOR: SolidColorBrush
+ ORIGINAL_BORDER_COLOR: SolidColorBrush
+ TaskBorderColor: SolidColorBrush
+ TaskBackgroundColor: SolidColorBrush
+ ID: int
+ Title: string
+ Description: string
+ CreationTime: DateTime
+ DueDate: DateTime
+ LastChangedDate: DateTime
+ AssigneeEmail: string
+ CurrentUser: UserModel <get> set>
+ ColumnOrdinal: int
+ UpdateTaskTitle (title: string): void
+ UpdateTaskDescription(description: string): void
+ AssignTaskUserEmail(assignee: string): void
+ CalculateTaskBorderColor(): SolidColorBrush
+ CalculateTaskBackgroundColor(): SolidColorBrush
+ RaiseProperty (propertyName: string): void
ColumnModel
+ MAX_COLUMN_NAME_LENGTH: int
+ Tasks: ObservableCollection<TaskModel> <<get,<set>>
+ TasksToView: ObservableCollection<TaskModel>
+ CreatedEmail: string
+ Ordinal: int
+ Limit: int
+ Name: string
- HandleChangeLocaler: Object, e: NotifyCollectionChangedEventArgs): void
+ RaiseProperty(propertyName: string): void
+ OnKeyDownHandlerNameLocaler: Object, e: KeyEventArgs): void
+ OnKeyDownHandlerLimitLocaler: Object, e: KeyEventArgs): bool
- SearchBox_TextChanged(sender:Text: string): bool
NotifiableModelObject
+ Controller: BackendController
BoardModel
+ CreatedEmail: string
+ Columns: ObservableCollection<ColumnModel>
- CurrentUser: UserModel
- CreateColumns(createdEmail: string): ObservableCollection<ColumnModel>
+ AdvanceTask(taskToAdvance: TaskModel, columnOrdinal: int): ObservableCollection<ColumnModel>
+ AddNewTask(newTask: TaskModel): void
+ AddColumn(newColumnName: string): void
+ RemoveTask(taskToRemove: TaskModel): void
+ MoveColumnLeft(columnOrdinal: int): void
+ MoveColumnRight(columnOrdinal: int): void
+ UpdateColumns(): void
UserModel
+ Email: string
+ Nickname: string
+ AssociatedBoard: string
Backend
Service Layer
Service Interfaces
«interface»
IService
+ LoadData () : Response
+ DeleteData () : Response
+ Register (email: string, password: string, nickname: string): Response
+ Register (email: string, password: string, nickname: string, hostEmail: string): Response
+ LoginUser (email: string, password: string): Response<User>
+ DeleteTask (email: string, columnOrdinal: int, taskid:int): Response
+ Login (email: string, password: string): Response<User>
+ LogoutUser (email: string): void
+ GetBoard (email : string): Response<Board>
+ LimitColumnTasks (email : string, columnOrdinal : int, limit : int): Response
+ AddTask(email: string, columnOrdinal: int, title: string, description: string): Response
+ AddTask(email: string, title: string, description: string, dueDate: DateTime): Response<Task>
+ UpdateTaskDueDate (email : string, columnOrdinal : int, taskid: int, dueDate: DateTime): Response
+ UpdateTaskTitle(email : string, columnOrdinal : int, taskid: int, title : string): Response
+ UpdateTaskDescription(email : string, columnOrdinal : int, taskid: int, description : string): Response
+ AdvanceTask(email : string, columnOrdinal : int, taskid: int): Response
+ GetColumn(email : string, columnName : string): Response<Column>
+ GetColumn(email : string, columnOrdinal : int): Response<Column>
+ AddColumn(email : string, Name : string): Response<Column>
+ AddColumn(email : string, columnOrdinal int, Name : string): Response<Column>
+ MoveColumnRight(email : string, columnOrdinal : int): Response<Column>
+ MoveColumnLeft(email : string, columnOrdinal : int): Response<Column>
Service
- bpc.topkrest.Ejog
- BoardService: BoardService
- UserService: UserService
BoardService
- bpc.topkrest.Ejog
- SecurityController: SecurityController
+ GetBoard (email : string): Response<Board>
+ LimitColumnTasks (email : string, columnOrdinal : int, limit : int): Response
+ AddTask(email: string, columnOrdinal: int, title: string, description: string): Response<Task>
+ UpdateTaskDueDate (email : string, columnOrdinal : int, taskid: int, dueDate : DateTime): Response
+ UpdateTaskTitle(email : string, columnOrdinal : int, taskid: int, title : string): Response
+ UpdateTaskDescription(email : string, columnOrdinal : int, taskid: int, description : string): Response
+ AdvanceTask(email : string, columnOrdinal : int, taskid: int): Response
+ GetColumn(email : string, columnName : string): Response<Column>
+ GetColumn(email : string, columnOrdinal : int): Response<Column>
+ AddColumn(email : string, Name : string): Response<Column>
+ AddColumn(email : string, columnOrdinal int, Name : string): Response<Column>
+ MoveColumnRight(email : string, columnOrdinal : int): Response<Column>
+ MoveColumnLeft(email : string, columnOrdinal : int): Response<Column>
UserService
- bpc.topkrest.Ejog
- SecurityController: SecurityController <<- getor>
+ Register (email: string, password: string, nickname: string): Response
+ Register (email: string, password: string, nickname: string, hostEmail: string): Response
+ LoginUser (email: string, password: string): Response<User>
+ LogoUt (email: string): Response
+ ChangePassword (email: string, oldPassword: string, newPassword: string): Response
Objects
User
+ Nickname : string
+ Email : string
+ AssociatedBoard: string
Task
+ Id: int
+ Title : string
+ Description: string
+ CreationTime: DateTime
+ DueDate: DateTime
+ emailAssignee: string
Colums
+ Task: IReadOnlyCollection<Task>
+ Name: string
+ Limit: int
Board
+ ColumnsNames: IReadOnlyCollection<string>
+ emailCreator: string
Response
- bpc.topkrest.Ejog
+ ErrorMessage: string
+ ErrorOccured: bool <<get>>
Extends
ResponseT
+ Value: T
Business Layer
SecurityController
- bpc.topkrest.Ejog
- SecurityController: SecurityController
+ UserController: UserController <get>
+ User Controller: BoardController <get>
+ CurrentUser: User
+ DeleteDatac(): void
+ LoginUser(email: string, password: string): User
+ LogoUt(email: string): void
+ ChangePassword(email: string): bool
+ ValidateHash(): bool
+ BoardExistence(boardid: string)
Board Package
BoardController
- bpc.topkrest.Ejog
- MINIMAL_NUMBER_OF_COLUMNS: int
- Boards: Dictionary<string,Board>
+ GetBoard (email: string): Board
+ GetColumn (email: string, columnName: string): Column
+ GetColumn (email: string, columnOrdinal: int): Column
+ LimitColumnTask (email: string, columnOrdinal: int, limit: int): void
+ AddTask(email: string, columnOrdinal: int, title: string): Task
+ AddTask (email string, title: string, description: string, dueDate: DateTime): Task
+ UpdateTaskTitle (email: string, columnOrdinal: int, taskid: int, title: int, currentUserEmail: string): void
+ UpdateTaskDescription (email: string, columnOrdinal: int, taskid: int, description: string, currentUserEmail: string): void
+ UpdateTaskDueDate (email: string, columnOrdinal: int, taskid: int, dueDate: DateTime, currentUserEmail: string): void
+ AssignTaskUserEmail (email: string, columnOrdinal: int, taskid: int, emailAssigner: string, currentUserEmail: string): void
+ AddNewBoard (email: string): void
+ JoinBoardUsingMemberEmail(email: string, newMemberNickname: string, boardTitleInEmail: string): void
+ RemoveBoard(email: string, boardTitleInEmail: string): void
+ RemoveColumn (email: string, columnOrdinal: int): void
+ MoveColumnRight (email: string, columnOrdinal: int): Column
+ MoveColumnLeft(email: string, columnOrdinal: int): Column
+ ChangeColumnName(email: string, columnOrdinal: int, newName: string): void
+ BoardExistence(email: string): bool
- DeleteTask(email: string, columnOrdinal: int, taskid: int, currentUserEmail: string): void
Board
- bpc.topkrest.Ejog
- MAXIMUM_COLUMN_NAME_LENGTH: int
- MINIMUM_COLUMN_NAME_LENGTH: int
+ CreatedEmail: string <get>
+ UserEmail: string <get>
+ TaskCounter: int <get;set>
+ Columns: ObservableCollection<string> <get> set>
+ DaCopyBoard: DaBoard
+ GetColumn(columnName string): Column
+ GetColumn(columnOrdinal int): Column
+ getColumnNames: List<String>
+ RemoveColumn(int): void
+ AddColumn(email: string, columnOrdinal: int, Name: string): Column
+ RemoveColumn(email: string, columnOrdinal: int): void
+ MoveColumnRight(email: string, columnOrdinal: int): Column
+ MoveColumnLeft(email: string, columnOrdinal: int): Column
+ ChangeColumnName(email: string, columnOrdinal: int, newName: string): void
+ AddMember(email: string, nickname: string): void
+ ToDeObject(): DaBoard
+ Save(): void
2..*
Column
- bpc.topkrest.Ejog
- INITIALIZE_MAXIMUM_NUMBER_OF_TASKS: int
+ Name: string <get>
+ Limit: int <get> set>
+ Tasks : List<Task> <get>
+ DaCopyColumn: DaColumn
+ LimitColumnTasks(limit int): void
+ AddTask(Task): void
+ RemoveTask(id: int): Task
+ GetTaskbaseId(id: int): Task
+ CheckLimit(): bool
+ ToDeObject(email:string, columnOrdinal: int): void
+ Save(): void
+ Delete(): void
+ ChangeName (newName: string): void
0..Limit
Task
- bpc.topkrest.Ejog
- MAXIMUM_TITLE_LENGTH: int
- MINIMUM_TITLE_LENGTH: int
- MAXIMUM_DESCRIPTION_LENGTH: int
+ ID: int <get>
+ Title: string <get> set>
+ Description: string <get> set>
+ CreationTime: DateTime <get>
+ DueDate: DateTime <get> set>
+ LastChangedDate: DateTime <get> set>
+ DaCopyTask: DaTask
+ UpdateTaskTitle (title: string): void
+ UpdateTaskDescription(description: string): void
+ UpdateTaskDueDate (dueDate: DateTime): void
+ AssignTaskUserEmail(assignee: string): void
- Save (email:string, columnName: string): void
+ AssignneeCheckIsUserInUserEmail: bool
+ AssignneeCheckIsAssigneeInNewAssignee(): void
+ Delete(): void
LogHelper
- BASE_PATH: string
- uebbd: bool
- LogTaskName: string
+ getLogger: log4net.ILog
+ StaticLog: void
+ StaticWriteLine(messageArg): Object[]
+ CreateLogger(logFileNameArg): ILogger
+ CreateLogger(ilog): void
Accessibility Reference
public = +
Internal = ~
protected = #
private = -
Data Access Layer
DAL0s
DaObjectsTx where T : DaObjectsTx
+ EmailColumnName: string const = email
# _controller: DaController
+ Email: string
+ Save(): bool
+ Delete(): bool
Extends
DaTask
- bpc.topkrest.Ejog
+ TaskIDColumnName: string const = ID
+ ContainingTaskColumnNameColumnName: string const = ColumnName
+ TaskTitleColumnName: string const = Title
+ TaskDescriptionColumnName: string const = Description
+ TaskAssigneeEmailColumnName: string const = EmailAssignee
+ TaskDueDateColumnName: string const = DueDate
+ TaskCreationTimeColumnName: string const = CreationDate
+ TaskLastChangedDateColumnName: string const = LastChangedDate
_ taskid : int
+ <<property>> Taskid: int
_ columnName: string
+ <<property>> ColumnName: string
_ title: string
+ <<property>> Title: string
_ description:string
+ <<property>> Description : string
_ emailAssignee: string
+ <<property>> EmailAssignee : string
_ dueDate: DateTime
+ <<property>> DueDate: DateTime
_ creationDate: DateTime
+ <<property>> CreationTime: DateTime
_ lastChangedDate: DateTime
+ <<property>> LastChangedDate: DateTime
DaUser
- bpc.topkrest.Ejog
+ UserPasswordColumnName: string const = Password
+ UserNicknameColumnName: string const = Nickname
+ UserAssociatedBoardColumnName: string const = AssociatedBoard
_ password: string
+ <<property>> Password: string
_ nickname: string
+ <<property>> Nickname: string
_ associatedBoard: string
+ <<property>> AssociatedBoard: string
DaBoard
- bpc.topkrest.Ejog
+ BoardTaskCounterName: string const = TaskCounter
_ taskCounter: int
+ <<property>> TaskCounter: int
+ Columns: List<DaColumn>
DaColumn
- bpc.topkrest.Ejog
+ ColumnNameColumnName: string const = Name
+ ColumnOrdinalColumnName: string const = Ordinal
+ ColumnLimitColumnName: string const = TasLimit
_ name: string
+ <<property>> Name: string
_ ordinal: int
+ <<property>> Ordinal: int
_ limit: int
+ <<property>> Limit: int
+ Tasks: List<DaTask>
DaControllers
BoardDaController
- bpc.topkrest.Ejog
- tableName: string const = Boards
+ ConvertReaderToObject(reader: SQLiteDataReader): DaBoard
+ CreateTable: void
+ SelectAllBoards(): List<DaBoard>
+ Insert(board: DaBoard): bool
+ Delete(board: DaBoard): bool
TaskDaController
- bpc.topkrest.Ejog
- tableName: string const = Tasks
+ ConvertReaderToObject(reader: SQLiteDataReader): DaTask
+ CreateTable: void
+ SelectAllTasks(tableName string, columnName string): List<DaTask>
+ Insert(task: DaTask): bool
+ Delete(task: DaTask): bool
Extends
DaControllers<T> where T : DaObjectsTx
- bpc.topkrest.Ejog
_connectionString: string
_tableName: string
+ ConvertReaderToObject(reader: SQLiteDataReader): DaObject <<abstract>>
+ Insert(T t): bool
+ Delete(int id): bool
+ CreateTable: void <<abstract>>
+ Select(): List<DaObject>
+ SelectColumn(): List<DaObject>
+ Updatecolumn(string, columnName string): List<DaObject>
+ Updatecolumn(string, attributeName string, attributeValue string): bool
+ Updatecolumn(string, attributeName string, attributeValue long): bool
+ Updatecolumn(string, columnOrdinal int, attributeName string, attributeValue string): bool
+ Updatecolumn(string, columnName string, attributeName string, attributeValue long): bool
+ Updatecolumn(string, columnName string, taskID int, attributeName string, attributeValue long): bool
+ Updatecolumn(string, columnOrdinal int, attributeName string, attributeValue long): bool
+ CreateORM( ): void
- CommandTextSelectAllByArgs: params string[]: string
- CommandTextUpdateUnitByUserName: string, keyArgs: string[]: string
UserDaController
- bpc.topkrest.Ejog
- tableName: string const = Users
+ ConvertReaderToObject(reader: SQLiteDataReader): DaUser
+ CreateTable: void
+ SelectAllUsers(): List<DaUser>
+ SelectUserByEmail(email: string): List<DaUser>
+ Insert(user: DaUser): bool
+ Delete(user: DaUser): bool
+ DeleteDatabase(): void
ColumnDaController
- bpc.topkrest.Ejog
- tableName: string const = Columns
+ ConvertReaderToObject(reader: SQLiteDataReader): DaColumn
+ CreateTable: void
+ SelectAllColumns(email: string): List<DaColumn>
+ Insert(column: DaColumn): bool
+ Delete(column: DaColumn): bool
SQLite
Entity Relations
Users
PK email: Text
password: Text
nickname: Text
AssociatedBoard: Text
Boards
PK, FK email: Text
TaskCounter: int
Columns
PK, FK email: Text
PK Name: Text
Limit: int
Task
PK, FK email: Text
PK, FK ColumnName: Text
PK ID: int
ordinal: int
Title: Text
Description: Text
EmailAssignee: Text
DueDate: Text
CreationDate: Text
LastChangedDate: Text