deleteTodo

Tuesday, August 7, 2018

3:12 PM

It deletes a todo. It should call the 3rd party persistent libraries to delete the todo in the database. It should check the input argument values in the todo object to be valid, id should not be null or empty. If one of them is empty or null throws TypeError ('Illegal Argument Exception'). If it can not find the todo to update it return null.

**Parameters**

todo Todo

**Return:** Todo

fetchTodo

Tuesday, August 7, 2018

3:15 PM

Get details of a specific todo. It should call the 3rd party persistent libraries to fetch data in the database. It should check the input argument values to be valid, id should not be null or empty. It throws TypeError ('Illegal Argument Exception') If id is empty or null. It returns null if it can not find the todo.

**Parameters**

id Number

**Return:** Todo

fetchAllTodos

Tuesday, August 7, 2018

3:22 PM

Get an array of todos. It should call the 3rd party persistent libraries for interacting with the database. It should check the input argument values to be valid, userId should not be null or empty. It throws TypeError ('Illegal Argument Exception') If userId is empty or null. It returns an empty array if it can not find the array.

**Parameters**

userId String

**Return:** Todo[]

getAllArchived

Tuesday, August 7, 2018

3:23 PM

Get an array of archived todos. Each todo has a status which shows this todo is completed, in-progress or archived. A todo is archived if the value of status property in object todo is "3". It should call the 3rd party persistent libraries for interacting with the database. It should check the input argument values to be valid, userId should not be null or empty. It throws TypeError ('Illegal Argument Exception') if userId is empty or null. It returns an empty array if it can not find any archived todo.

Parameters

userId String

Return: Todo[]

getAllDone

Tuesday, August 7, 2018

8:23 PM

Get an array of completed todos. Each todo has a status which shows this todo is completed, in-progress or archived. A todo is completed or done if the value of status be "1". It returns an empty array if it can not find the array. It should call the 3rd party persisting APIs for interacting with the database. It should check the argument values to be valid, userId should not be null or empty. It throws TypeError ('Illegal Argument Exception') if the input argument is empty or null.

**Parameters**

userId String

**Return:** Todo[]

getAllTodoOfaGroup

Tuesday, August 7, 2018

8:13 PM

Get an array of todos of a specific group. It returns an empty array if it can not find the todos items for the groupId. It should call the 3rd party persisting APIs for interacting with the database. It should check the input argument values to be valid, groupId and userId should not be null or empty. It throws TypeError ('IllegalArgumentException') if their values are empty or null.

**Parameters**

groupId String

userId String

**Return:** Todo[]

markTodoAsDone

Tuesday, August 7, 2018

8:28 PM

It updates the value of status of a todo. Each todo has a status which shows this todo is completed, in-progress or archived. An item is completed if the status equals "1". It returns false if it can not find the todo. It should call the 3rd party persistent libraries for interacting with the database. It should check the input argument value to be valid, id should not be null or empty. It throws TypeError('Illegal Argument Exception') if it is null or empty.

**Parameters**

id Number

**Return:** Boolean

markTodoAsArchived

Tuesday, August 7, 2018

8:34 PM

It updates the value of status of a todo. Each todo has a status which shows this todo is completed, in-progress or completed. An item

is completed if the status equals "3". It returns false if it can not find the todo. It should change the status of the function to complete and update it. It should call the 3rd party persistent libraries for interacting with the database

. It should check the input argument value to be valid, id should not be null or empty. It throws

TypeError('Illegal Argument Exception') if it is null or empty.

**Parameters**

id Number

**Return:** Boolean

todo

Saturday, September 8, 2018

4:22 PM

{  
 title: String  
 description: String  
 dueDate: String  
 dataStoreId: String  
 userId: String  
 id: Number  
 status: Number  
 groupId: String  
 createdTime: String  
 createdDate: String  
 priority: Number  
 address: String  
 repeat: String  
}

{  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2018 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 1,  
 "status": 1,  
 "groupId": "schoolworks",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 1,  
 "address": "Fairfax,VA,US 22032",  
 "repeat": "2"  
}

todoArray

Saturday, September 8, 2018

4:22 PM

{  
 todoArray: Todo[]  
}

{  
 "todoArray": [  
 {  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2020 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 1,  
 "status": 1,  
 "groupId": "schoolworks",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 3,  
 "address": "Fairfax,VA,US 22032",  
 "repeat": "1"  
 },  
 {  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2018 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 2,  
 "status": 3,  
 "groupId": "schoolworks",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 2,  
 "address": "Fairfax,VA,US 22032",  
 "repeat": "2"  
 },  
 {  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2019 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 3,  
 "status": 2,  
 "groupId": "Homework",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 1,  
 "address": "Fairfax,VA,US 22032",  
 "repeat": "3"  
 }  
 ]  
}

Objects

Tuesday, August 7, 2018

8:42 PM

Todo= {  
 title: String  
 description: String  
 dueDate: String  
 dataStoreId: String  
 userId: String  
 id: Number  
 status: Number  
 groupId: String  
 createdTime: String  
 createdDate: String  
 priority: Number  
 address: String

repeat:String}

Eample

{  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2018 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 1,  
 "status": 1,  
 "groupId": "schoolworks",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 1,  
 "address": "Fairfax,VA,US 22032",

"repeat": "2"

}

todoArray

{  
 todoArray: Todo[]  
}

Example

{  
 "todoArray": [  
 {  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2020 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 1,  
 "status": 1,  
 "groupId": "schoolworks",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 3,  
 "address": "Fairfax,VA,US 22032",

"repeat": "1"

},  
 {  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2018 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 2,  
 "status": 3,  
 "groupId": "schoolworks",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 2,  
 "address": "Fairfax,VA,US 22032",

|  |  |
| --- | --- |
|  | "repeat": "2" |

},  
 {  
 "title": "coding",  
 "description": "work on the crowd cod",  
 "dueDate": "03/14/2019 15:29",  
 "dataStoreId": "todo3",  
 "userId": "emad.aghayi",  
 "id": 3,  
 "status": 2,  
 "groupId": "Homework",  
 "createdTime": "13:51",  
 "createdDate": "05/21/2018",  
 "priority": 1,  
 "address": "Fairfax,VA,US 22032",

"repeat": "3"

}  
 ]  
}

updateTodo

Tuesday, August 7, 2018

2:54 PM

It updates a todo and stores it. It should call the 3rd party persistent libraries to update data in the database. It should check the input arguments values in the todo object like userId, dataStoreId, id, createdTime, createdDate to be valid, It means they should not be null or empty. It throws TypeError ('Illegal Argument Exception') If one of them is empty or null. Also, It throws TypeError ('No Such Element Exception') if it can not find the todo to update it. If the value of dueDate is not empty or null, It should check the value of dueDate properties to be in the format of "MM/DD/YYY,HH:MM", example: "05/02/2018,23:25". If the value of dueDate is not in the desired format, it should throw TypeError ('Illegal Argument Exception').

**Parameters**

todo Todo

**Return:** Todo

addTodo

Tuesday, August 7, 2018

2:54 PM

It adds a todo item. It calls the 3rd party persistent libraries to store todo in the database. It should check the input arguments values in the todo object like id, title, userId, dataStoreId, to be valid, It means they should not be null or empty. It throws TypeError ('Illegal Argument Exception') if one of them is empty or null. If the values of createdTime or createdDate are empty or null it should set the current date (ex: 02/24/2018) and time (ex: 19:25) as their values. If the dueDate value is not empty or null, It should check the value of dueDate properties to be in the format of "MM/DD/YYY,HH:MM", example: "05/02/2018,23:25". If the value of dueDate is not in the desired format, it should throw TypeError ('Illegal Argument Exception').

**Parameters**

todo Todo

**Return:** Todo

remindOnDueDate

Tuesday, August 7, 2018

8:36 PM

It reminds todos based on the dueDate. It looks between all active (not completed nor archived) todos find the todos which their dueDate is same as method input arguments. It also should consider the value of the property of repeat (value 1 means daily repeat, 2 means weekly repeat, 3 means yearly repeat), based on that calculate the value of dueDate of todos in the database and compare it with input arguments; for example a todo in the database which its todo.dueDate property is "03/14/2020 15:29" and todo.repeat value is 1 if the input arguments are "03/15/2020 15:29", it should add this todo to the array of return values. It should call the 3rd party persistent libraries for interacting with the database. It should check the input argument value to be valid, dueDate and userId should not be null or empty. It throws TypeError('Illegal Argument Exception') if dueDate or userId be null or empty. It should check the value of dueDate properties to be in the format of "MM/DD/ YYYY, HH:MM", example: "05/02/2018,23:25". If the value of dueDate is not in the desired format, it should throw TypeError ('Illegal Argument Exception').

**Parameters**

userId String

dueDate String

**Return:** Todo[]

updatePriorityOfaTodo

Tuesday, August 7, 2018

8:15 PM

Update priority of a todo. The value of the property of priority can be valued between 1 to 4 which 1 means the lowest priority and 4 means the highest priority. It returns false if it can not find the todo. It should call the 3rd party persisting API for interacting with the database. It should check the input argument values to be valid, priority and id should not be null or empty. It throws TypeError ('Illegal Argument Exception') if they are empty or null. It throws TypeError ('IllegalArgumentException'), if the value of priority is more than 4 or less than 1.

**Parameters**

id Number

priority Number

**Return:** Boolean

fetchTodosBasedOnStatus

Wednesday, August 8, 2018

10:08 PM

Get an array of todos based on the status. Each todo has a status which shows this todo is completed, in-progress or archived. A todo is completed or done if the value of status be "1", is archived if the value of status property in object todo is "3", and it is in -progress if the status property in object todo is "2". It returns an empty array if it can not find the array. It should call the 3rd party persisting APIs for interacting with the database. It should check the argument values to be valid, userId and status should not be null or empty. It throws TypeError ('Illegal Argument Exception') if the input arguments are empty or null. It throws TypeError ('Illegal Argument Exception') if the value of status is anything except 1, 2 or 3.

**Parameters**

userId String

**Return:** Todo[]

createGroup

Tuesday, August 7, 2018

8:01 PM

It creates a group from an array of todos. It sets a label (groupId) for an array of todos. It should call the 3rd party persisting API for interacting with the database. It should set the groupId for selected todos and update it in the database. It returns false if it can not find the all todos items for creating the group. It should check the input argument values to be valid, groupId should not be null or empty. It throws TypeError ('IllegalArgumentException'), if it is empty or null. It throws TypeError ('DuplicateGroupId ') if the groupId already existed in the other todos.

**Parameters**

todoArray Todo[]

groupId String

**Return:** Boolean