

PROJECT NAME

Enter a project name in the textbox below to retrieve the client request for the project. If it does not exist, it will be created.

todo

Load or Create

DESCRIPTION

Describe the Project with its intended use cases and briefly explain its requirements

A microservice backend for a todo service. Main features include 1. adding, updating, and deleting todo items, 2. marking todo items as done, 3. organizing todo items into groups, 4. tagging todo items with priorities and locations

ADTS

Describe ADTs with a description, name, structure, and some example. The JSON structure should be of the form **ADT name: TypeName**, where each TypeName is either defined separately as an ADT or is one of the three primitives String, Number, Boolean. To indicate an n-dimensional array, add n sets of brackets after the type name (e.g., 2 dimensional array - **TypeName[][]**). The description should describe any rules about the ADT and include an example of a value of the ADT in JSON format.

description	A todo object contains field like id, status, due date.... It is used as data transfer object (DTO) for working in methods.	delete ADT																																						
ADT name	Todo																																							
JSON structure:	<table border="1"> <tr><td>title</td><td>String</td><td>delete Structure</td></tr> <tr><td>description</td><td>String</td><td>delete Structure</td></tr> <tr><td>dueDate</td><td>String</td><td>delete Structure</td></tr> <tr><td>dataStoreId</td><td>String</td><td>delete Structure</td></tr> <tr><td>userId</td><td>String</td><td>delete Structure</td></tr> <tr><td>id</td><td>Number</td><td>delete Structure</td></tr> <tr><td>status</td><td>Number</td><td>delete Structure</td></tr> <tr><td>groupId</td><td>String</td><td>delete Structure</td></tr> <tr><td>createdTime</td><td>String</td><td>delete Structure</td></tr> <tr><td>createdDate</td><td>String</td><td>delete Structure</td></tr> <tr><td>priority</td><td>Number</td><td>delete Structure</td></tr> <tr><td>address</td><td>String</td><td>delete Structure</td></tr> <tr><td>repeat</td><td>String</td><td>delete Structure</td></tr> </table>	title	String	delete Structure	description	String	delete Structure	dueDate	String	delete Structure	dataStoreId	String	delete Structure	userId	String	delete Structure	id	Number	delete Structure	status	Number	delete Structure	groupId	String	delete Structure	createdTime	String	delete Structure	createdDate	String	delete Structure	priority	Number	delete Structure	address	String	delete Structure	repeat	String	delete Structure
title	String	delete Structure																																						
description	String	delete Structure																																						
dueDate	String	delete Structure																																						
dataStoreId	String	delete Structure																																						
userId	String	delete Structure																																						
id	Number	delete Structure																																						
status	Number	delete Structure																																						
groupId	String	delete Structure																																						
createdTime	String	delete Structure																																						
createdDate	String	delete Structure																																						
priority	Number	delete Structure																																						
address	String	delete Structure																																						
repeat	String	delete Structure																																						
	Add Field																																							
Examples:	Todo Example1 <pre>{ "title": "coding", "description": "work on the crowd cod", "dueDate": "03/14/2018 15:29", "dataStoreId": "todo3", "userId": "emad.aghay", "id": 1, "status": 1, "groupId": "schoolworks", "createdTime": "13:51", "createdDate": "05/21/2018", "priority": 1, "address": "Fairfax,VA,US 22032", "repeat": "2" }</pre>																																							
	Add Example	delete Example																																						

description	It is an array of todos like [todo1,todo2,...]	delete ADT		
ADT name	TodoArray			
JSON structure:	<table border="1"> <tr><td>todoArray</td><td>Todo[]</td><td>delete Structure</td></tr> </table>	todoArray	Todo[]	delete Structure
todoArray	Todo[]	delete Structure		
Examples:	example Array of todos. <pre>["todoArray": [{ "title": "coding", "description": "work on the crowd cod", "dueDate": "03/14/2020 15:29", "dataStoreId": "todo3", "userId": "emad.aghay", "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.aghay", "id": 2, "status": 3, "groupId": "schoolworks", "createdTime": "13:51", "createdDate": "05/21/2018", "priority": 2, "address": "Fairfax,VA,US 22032", "repeat": "1" }] }</pre>			
	Add Example	delete Example		

Add new ADT

FUNCTIONS

Describe End points of your micro service with a name, list of parameters, description.

Description	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/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') .		Delete this Function
Return type	Todo		
Endpoint name	addTodo		
Third party API			
parameters	todo	Todo	It contain some field like title, date, desc of a todo
	Add New Param		Delete this Param
Description	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/YYYY:HH:MM", example: "05/02/2018,23:25". If the value of dueDate is		Delete this Function

	not in the desired format, it should throw TypeError ('Illegal Argument Exception').		
Return type	Todo		
Endpoint name	updateTodo		
Third party API	<input type="checkbox"/>		
parameters	todo	Todo	it contain some field like title, date, desc of a todo Delete this Param
	Add new Param		
Description	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		
Return type	Todo		
Endpoint name	deleteTodo		
Third party API	<input type="checkbox"/>		
parameters	todo	Todo	it contain some field like title, desc of a todo Delete this Param
	Add new Param		
Description	It inserts an object in the database. It returns 'Duplicate' if it is already persisted in the database. Delete this Function		
Return type	Todo		
Endpoint name	SaveObject		
Third party API	<input checked="" type="checkbox"/>		
parameters	todo	Todo	it contain some field like title, date, desc of a todo Delete this Param
	Add new Param		
Description	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 Delete this Function		
Return type	Todo		
Endpoint name	fetchTodo		
Third party API	<input type="checkbox"/>		
parameters	id	Number	id of todo Delete this Param
	Add new Param		
Description	Fetch the object from the database. It returns null if the desired object does not exist. Delete this Function		
Return type	Todo		
Endpoint name	FetchObject		
Third party API	<input checked="" type="checkbox"/>		
parameters	objectId	Number	id for todo Delete this Param
	Add new Param		
Description	Delete an object from the database. It returns null if the desired object does not exist. Delete this Function		
Return type	Todo		
Endpoint name	DeleteObject		
Third party API	<input checked="" type="checkbox"/>		
parameters	todo	Todo	an todo item Delete this Param
	Add new Param		
Description	Update an object in the database. It returns null if the desired object does not exist. Delete this Function		
Return type	Todo		
Endpoint name	UpdateObject		
Third party API	<input checked="" type="checkbox"/>		
parameters	todo	Todo	an todo item Delete this Param
	Add new Param		
Description	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 Delete this Function		
Return type	Todo[]		
Endpoint name	fetchAllTodos		
Third party API	<input type="checkbox"/>		
parameters	userId	String	get all todos of each user Delete this Param
	Add new Param		
Description	Get the list of objects in the database. Delete this Function		
Return type	Todo[]		
Endpoint name	FetchAllObjects		
Third party API	<input checked="" type="checkbox"/>		
parameters	userId	String	get all todos of each user from DB Delete this Param
	Add new Param		
Description	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 Delete this Function		
Return type	Boolean		
Endpoint name	createGroup		
Third party API	<input type="checkbox"/>		
parameters	todoArray	Todo[]	an array of todos Delete this Param
	groupId	String	Id of the group Delete this Param
	Add new Param		
Description	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 Delete this Function		
Return type	Todo[]		
Endpoint name	getAllTodosOfaGroup		
Third party API	<input type="checkbox"/>		
parameters	groupId	String	Id of the group Delete this Param
	userId	String	Id for user want to get his todos Delete this Param
	Add new Param		
Description	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 Delete this Function		
Return type	Boolean		
Endpoint name	updatePriorityOfaTodo		
Third party API	<input type="checkbox"/>		

parameters	<input type="text" value="id"/>	<input type="text" value="Number"/>	<input type="text" value="id for a todo"/>	<input type="button" value="Delete this Param"/>
	<input type="text" value="priority"/>	<input type="text" value="Number"/>	<input type="text" value="priority of a todo it can be 1,2,3,4"/>	<input type="button" value="Delete this Param"/>
Add new Param				
Description	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			<input type="button" value="Delete this Function"/>
Return type	<code>Todo[]</code>			
Endpoint name	fetchTodosBasedOnStatus			
Third party API				
parameters	<input type="text" value="userId"/>	<input type="text" value="String"/>	<input type="text" value="an id for todos of a user"/>	<input type="button" value="Delete this Param"/>
Add new Param				
Description	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			<input type="button" value="Delete this Function"/>
Return type	<code>Boolean</code>			
Endpoint name	markTodoAsDone			
Third party API				
parameters	<input type="text" value="id"/>	<input type="text" value="Number"/>	<input type="text" value="id for a todo"/>	<input type="button" value="Delete this Param"/>
Add new Param				
Description	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			<input type="button" value="Delete this Function"/>
Return type	<code>Todo[]</code>			
Endpoint name	remindOnDueDate			
Third party API				
parameters	<input type="text" value="userId"/>	<input type="text" value="String"/>	<input type="text" value="an id for todos of a user"/>	<input type="button" value="Delete this Param"/>
	<input type="text" value="dueDate"/>	<input type="text" value="String"/>	<input type="text" value="the deadline of doing a todo"/>	<input type="button" value="Delete this Param"/>
Add new Param				
Description	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 "3". It returns false if it can not find the todo. It should call the 3rd party persistent libraries			<input type="button" value="Delete this Function"/>
Return type	<code>Boolean</code>			
Endpoint name	markTodoAsArchived			
Third party API				
parameters	<input type="text" value="id"/>	<input type="text" value="Number"/>	<input type="text" value="id of todo"/>	<input type="button" value="Delete this Param"/>
Add new Param				

[Add new function](#)

[Save or update the Project](#)