<table>
<tr>
<td>GET
/api/:role/inStock</td>
<td>
<h3>Path variables</h3>

role : Role in the Inventory Management

<h3>Description</h3>
This API allows the retrieval of the in stock levels for all resources in the inventory, based on the role. For example, if the role of INVENTORY_MANAGER is assumed, then the current inventory for the incoming components is returned.
<h3>Example</h3>
/api/INVENTORY_MANAGER/inStock
<h3>Request body</h3>
Empty
<h3>Response</h3>
<pre>
[
    {
        "id": "5e19ad6e0274390007e95477",
        "name": "REV",
        "inStock": 5
    },
    {
        "id": "5e240b010274390007c41802",
        "name": "switch00",
        "inStock": 2
    },
    {
        "id": "5e2939f60274390007c41842",
        "name": "REV1",
        "inStock": 0
    }
]
</pre>
</td>
</tr>
<tr>
<td>GET
api/:role/:resource/inStock</td>
<td>
<h3>Path variables</h3>

role : Role in the Inventory Management
resource : Resource maintained in the Inventory

</td>
</tr>
</table>

	Description
	This API allows the retrieval of the in stock levels for a particular resource in the inventory, based on the role. For example, if the role of INVENTORY_MANAGER is assumed, then the current inventory for a specific incoming component is returned.
	
	Example
	/api/INVENTORY_MANAGER/REV/inStock
	
	Request body
	Empty
	
	Response
	<pre>{
	"id": "5e19ad6e0274390007e95477",
	"name": "REV",
	"inStock": 5
	}</pre>

GET **api/:role/:resource/daysOfStock**	Path variables - role : Role in the Inventory Management - resource : Resource maintained in the Inventory Description This API allows the calculation of the days of stock available for a particular resource in the inventory, based on the role and current inventory. For example, if the role of INVENTORY_MANAGER is assumed, then the days of stock available for a specific incoming component is returned. Example /api/INVENTORY_MANAGER/REV/inStock Request body *{* *"demand":3* *}* demand - daily demand of the resource

<table>
<tr><td></td><td>

Response

<pre>
{
    "id": "5e19ad6e0274390007e95477",
    "name": "REV",
    "inStock": 5,
    "daysOfStock": 1
}
</pre>

</td></tr>
<tr><td>GET
api/:role/:resource/safetyStock</td><td>

Path variables

- role : Role in the Inventory Management
- resource : Resource maintained in the Inventory

Description

This API allows the calculation of the safety stock of a particular resource to be maintained, based on the role, lead time, and average demand. For example, if the role of INVENTORY_MANAGER is assumed, then the safety stock to be maintained for a specific incoming component is returned.

Example

/api/INVENTORY_MANAGER/REV/safetyStock

Request body

{
"leadTime":2,
"demandAvg":5
}
leadTime - Lead time for the resource
demandAvg - average demand of the resource in terms of days

Response

<pre>
{
    "id": "5e19ad6e0274390007e95477",
    "name": "REV",
    "safetyStock": 10
}
</pre>

</td></tr>
<tr><td>GET
api/:role/:model/components</td><td>

Path variables

- role : Role in the Inventory Management

</td></tr>
</table>

- model : Model name of bike

Description

This API allows the retrieval of all the components required to assemble a bike model, and each of the possible choices for the same.

Example

/api/ASSEMBLY_MANAGER/Battre_RS/components

Request body

Empty

Response


```
{
    "List": [
        "CO1",
        "CO2",
        "BA1",
        "BA2",
        "SP1",
        "TY1",
        "TY2",
        "MO1",
        "MO2",
        "CH1",
        "DC1",
        "LI1",
        "LI2",
        "SW1",
        "OT1"
    ],
    "Components": {
        "CONTROLLER": [
            "CO1",
            "CO2"
        ],
        "BATTERY": [
            "BA1",
            "BA2"
```

<table>
<tr>
<td></td>
<td>

```
        ],
        "SPEEDOMETER": [
            "SP1"
        ],
        "TYRE": [
            "TY1",
            "TY2"
        ],
        "MOTOR": [
            "MO1",
            "MO2"
        ],
        "CHASSIS": [
            "CH1"
        ],
        "DC-DC CONVERTER": [
            "DC1"
        ],
        "LIGHTS": [
            "LI1",
            "LI2"
        ],
        "SWITCHES": [
            "SW1"
        ],
        "OTHERS": [
            "OT1"
        ]
    }
}
```

</td>
</tr>
<tr>
<td>GET
api/:role/:model/makes</td>
<td>
<h3>Path variables</h3>

role : Role in the Inventory Management
model : Model name of bike

<h3>Description</h3>
This API allows the retrieval of the different makes of a bike model based on the different choices of components that can be used.
</td>
</tr>
</table>

Example

/api/ASSEMBLY_MANAGER/Battre_RS/makes

Request body

Empty

Response


```
{
    "Combinations": [
        [
            "CO1",
            "BA1",
            "SP1",
            "TY1",
            "MO1",
            "CH1",
            "DC1",
            "LI1",
            "SW1",
            "OT1"
        ],
        [
            "CO1",
            "BA1",
            "SP1",
            "TY1",
            "MO1",
            "CH1",
            "DC1",
            "LI2",
            "SW1",
            "OT1"
        ],
        [
            "CO1",
            "BA1",
            "SP1",
            .
```

	. . .

GET
api/:role/:model/counts
	Path variables
	- role : Role in the Inventory Management
	- model : Model name of bike
	
	Description
	This API allows the retrieval of the number of different makes of a bike model that can be assembled based on the current inventory.
	
	Example
	/api/ASSEMBLY_MANAGER/Battre_RS/counts
	
	Request body
	{
	"available": [9,8,8,9,8,7,8,7,9,8,9,7,9,8]
	}
	available - list of in stock values for the list of components returned by the **api/:role/:model/components** api
	
	Response
	[
	{
	"Configuration": {
	"CO1": 1,
	"CO2": 0,
	"BA1": 1,
	"BA2": 0,
	"SP1": 1,
	"TY1": 1,
	"TY2": 0,
	"MO1": 1,
	"MO2": 0,
	"CH1": 1,
	"DC1": 1,
	"LI1": 1,
	"LI2": 0,
	"SW1": 1,

```
            "OT1": 1
        },
        "VehicleCount": 7,
        "PostInventory": [
            2,
            8,
            1,
            9,
            1,
            0,
            8,
            0,
            9,
            1,
            2,
            0,
            9,
            1
        ]
    },
    {
        "Configuration": {
            "CO1": 1,
            "CO2": 0,
            "BA1": 1,
            "BA2": 0,
            "SP1": 1,
            "TY1": 1,
            "TY2": 0,
            "MO1": 1,
            "MO2": 0,
            "CH1": 1,
            "DC1": 1,
            "LI1": 0,
            "LI2": 1,
            "SW1": 1,
            "OT1": 1
        },
```

	"VehicleCount": 7, "PostInventory": [2, 8, 1, 9, 1, 0, 8, 0, 9, 1, 2, 7, 2, 1] }, ...

<table>
<tr>
<td>GET
api/lm/getData</td>
<td>
<h3>Path variables</h3>
none
<h3>Description</h3>
This API extracts details of all the rental subscriptions.
<h3>Request body</h3>
Empty
<h3>Response</h3>
<pre>
[
    {
        "id": "5e4e29d524aa9a0007d3b274",
        "vin": "office_test_miracle",
        "startTime": "2020-02-21T00:00:00.000Z",
        "endTime": "2020-03-21T00:00:00.000Z",
        "status": "BOOKED"
    }
]
</pre>
</td>
</tr>
<tr>
<td>POST
api/lm/newVehicle/:vin/:offset</td>
<td>
<h3>Path variables</h3>
vin : Vin of vehicle
offset : Delay in between subscriptions (in days)
<h3>Description</h3>
This API adds a new vehicle to the object store (linked list) and initializes it with the provided vin and offset value.
<h3>Example</h3>
/api/lm/newVehicle/office_test_miracle/1
<h3>Request body</h3>
Empty
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
<h3>Response (if vehicle not already present)</h3>
<pre>
{
    "office_test_miracle": {
        "rentalStatus": "BOOKED",
        "offset": 1,
        "next": null
    }
}
</pre>
<h3>(if vehicle present)</h3>
<pre>Vehicle already present</pre>
</td>
</tr>
<tr>
<td>POST
api/lm/addNode/:vin/:id/:after</td>
<td>
<h3>Path variables</h3>

vin : Vin of vehicle
id : Id of the rental subscription
after : Node after which insertion should be done

<h3>Description</h3>
This API updates the object store with respect to a particular vin, and adds a new node which corresponds to a new rental subscription.
<h3>Example</h3>
/api/lm/addNode/office_test_miracle/5e4e29d524aa9a0007d3b274/-1
<h3>Request body</h3>
Empty
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
Request body
<pre>
{
    "office_test_miracle": {
        "rentalStatus": "BOOKED",
        "offset": 1,
        "next": {
            "id": "5e4e29d524aa9a0007d3b274",
            "startTime": "2020-02-21T00:00:00.000Z",
            "endTime": "2020-03-21T00:00:00.000Z",
            "next": null
        }
    }
}
</pre>
</td>
</tr>
<tr>
<td>GET
api/lm/availSlots/:vin/:currTS</td>
<td>
Path variables

vin : Vin of vehicle
currTS : current timestamp to compare with

Description
This API returns the available slots with respect to the rental subscriptions of a vehicle with the specified vin from the object store. The current timestamp is also used if the vehicle is available, to check for a free slot until the next booking
Example
api/lm/availSlots/office_test_miracle/2020-02-19T00:00:00.000Z
</td>
</tr>
</table>

	Request body
	Empty
	
	Response
	[
	{
	"after": "Current",
	"from": "2020-02-19T00:00:00.000Z",
	"till": "2020-02-21T00:00:00.000Z"
	},
	{
	"after": "5e4e29d524aa9a0007d3b274",
	"from": "2020-03-22T00:00:00.000Z",
	"till": "NextNewBooking"
	}
	]

GET **api/lm/getObjectStore**	Path variables - none Description This API returns the entire object store with details of the rental subscriptions corresponding to all the vin's. Request body Empty

<table>
<tr>
<td></td>
<td>
Response
<pre>
{
    "office_test_miracle": {
        "rentalStatus": "BOOKED",
        "offset": 1,
        "next": {
            "id": "5e4e29d524aa9a0007d3b274",
            "startTime": "2020-02-21T00:00:00.000Z",
            "endTime": "2020-03-21T00:00:00.000Z",
            "next": null
        }
    }
}
</pre>
</td>
</tr>
<tr>
<td>GET
api/lm/getAvail/:vin/:fromTS/:forDays</td>
<td>
Path variables

vin : Vin of vehicle
fromTS : timestamp from which vehicle is required
forDays : number of days for which vehicle is required

Description

This API checks and returns suitable slots for a new rental subscription based on availability, given the period of requirement.

Example

localhost:3000/api/lm/getAvail/office_test_miracle/2020-02-20T00:00:00.000Z/2
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
<h3>Request body</h3>
avail : available slots for rental subscriptions
<pre>
{
        "avail": [
           {
              "after": "Current",
              "from": "2020-02-19T00:00:00.000Z",
              "till": "2020-02-21T00:00:00.000Z"
           },
           {
              "after": "5e4e29d524aa9a0007d3b274",
              "from": "2020-03-22T00:00:00.000Z",
              "till": "NextNewBooking"
           }
        ]
}
</pre>
<h3>Response</h3>
<pre>
[
    {
        "after": "5e4e29d524aa9a0007d3b274",
        "from": "2020-03-22T00:00:00.000Z",
        "till": "NextNewBooking"
    }
]
</pre>
</td>
</tr>
<tr>
<td>POST
api/lm/deleteNode/:vin/:id</td>
<td>
<h3>Path variables</h3>

vin : Vin of vehicle
id : Id of the rental subscription

<h3>Description</h3>
This API updates the object store and deletes a node corresponding to the vehicle vin and the id of the subscription
<h3>Example</h3>
/api/lm/deleteNode/office_test_miracle/5e4e29d524aa9a0007d3b274
<h3>Request body</h3>
Empty
</td>
</tr>
</table>

	Response `Node deleted`
POST **api/lm/updateObjStore/:vin/:currTS**	Path variables - vin : Vin of vehicle - currTS : current timestamp compared to which deletion to be performed Description This API updates the object store and deletes nodes from the object store that not valid in terms of endTime. Example /api/lm/deleteNode/office_test_miracle/2020-03-21T00:00:00.000Z Request body Empty Response `Updated`