Homework 1

Create API for following things

- Add a new formula.
- See the inputs and outputs of a specific formula.
- List which plans contain a specific formula.
- Add a new plan.
- Append a formula to the end of a plan.
- Replace a formula anywhere in the sequence of formulas associated with a plan.
- List all the formulas contained in a plan.
- Delete a plan.

API Reference

1: Add a new formula.

```
POST /formula
```

Parameter	Description
-	-

Request Body	Description
JSON	A json object containing input resources with quantity and output resources with quantity

Request Body Example:

```
{
   "inputs": [

       { "resource": "iron ore", "quantity": 3 },
       { "resource": "coal", "quantity": 1.5 }
   ],
   "outputs": [
       { "resource": "iron bar", "quantity": 10000 }
   ]
}
```

Response codes	Description
200	New formula Successfully added
400	The formula did not added

Response Body	Description
JSON	A json object containing newly added formula.

2: See the inputs and outputs of a specific formula.

```
GET /formulas/:fid/details
```

Parameter	Description
:fid	Non negative formula ID

Request Body	Description
-	-

Response codes	Description
200	Successfully returned Formula Details
400	Invalid ID

Response Body	Description
JSON	A json objetct containing formula details.

3: List which plans contain a specific formula.

```
GET /plan/formula/:fid
```

Parameter	Description
:fid	Non negative formula ID

Request Body	Description
-	-

Response codes	Description
200	Successfully returned list of plan
400	Invalid ID or Formula does not exist

Response Body	Description
JSON	A json object containing list of plans detail.

4: add new plan

```
POST /plan
```

Parameter	Description
-	-

Request Body	Description
JSON	A json object containing list of formulas.

Request Body Example:

```
{
    "formulas": [0,3]
}
```

Response codes	Description
200	Successfully returned details of plan

Response Body	Description
JSON	A json object containing details of plan.

5: append a formula to end of a plan

```
POST /plan/:pid/add_Formula
```

Parameter	Description
:pid	Non negative plan ID

Request Body	Description
JSON	A json object containing formula ID to be added to end of list.

Request Body Example:

```
{
    "formula": 3
}
```

Response codes	Description
200	Successfully added a formula at the end of formula list in a plan
400	Invalid ID or plan does not exist

Response Body	Description
JSON	A json object containing updated plan details.

6: Replace a formula anywhere in the sequence of formulas associated with a plan.

```
PUT /plan/:pid/formula/:fid
```

Parameter	Description
:pid	Non negative plan ID

Request Body	Description

Request Body	Description
JSON	A json object containing formula ID to be updated.

Request Body Example:

```
{
    "formula": 2
}
```

Response codes	Description
200	Successfully updated list of formulas in a plan
400	Invalid ID or plan does not exist

Response Body	Description
JSON	A json object with update plan and formula list.

7: list of formulas containing in a plan.

```
GET /plan/:pid/formulas
```

Parameter	Description
:pid	Non negative plan ID

Request Body	Description
-	-

Response codes	Description
200	Successfully returned list of formulas in a plan
400	Invalid ID or plan does not exist

Response Body	Description
JSON	A json object containing details of all the formulas in that plan.

8: Delete a plan

```
DELETE /plan/:pid
```

Parameter	Description
:pid	Non negative plan ID

Request Body	Description
-	-

Response codes	Description
200	Successfully deleted a plan
400	Invalid ID or plan does not exist

Response Body	Description
JSON	A json object containing recently deleted plan.