Homework 1 APIs Documentation

Sizhe Liu

Overview:

* The server is implemented on app.js file only, which is allowed in homework 1.

Important Variables:

* ‘formulaId’ is for formula ID. It is unique and global. It will only be incremented.
* ‘planId’ is for plan ID. Same rule applies.

Data Structures:

* ‘plans’ is Map(). The key is the plan ID (number). The value is an array that holds formula IDs (numbers) that belong to a given plan. A plan cannot have duplicate formula. For example, if plan 0 already contains formula 1, you cannot add formula 1 to the plan again. Neither can you replace another formula other than 1 to formula 1 because that will introduce duplication. For example, if plan 1 has [0, 1], you cannot replace formula 0 to formula 1. The format of plans Map() looks like this as an example:
* {  
   0: [4,2,1]   
  }
* ‘formulas’ is Map(). The key is the formula ID (number). The value is an object. The object itself looks like the following:
* const f1 = {  
   "inputs": [{"name": "butter", "number": 2}, {"name": "egg", "number": 1},  
   {"name": "sugar", "number": 1}, {"name": "flour", "number": 2},  
   {"name": "baking soda", "number": 2}],  
   "outputs": [{"name": "cookies", "number": 36}]  
  }

API:

|  |
| --- |
| POST /formulas  add a new formula to the pool of formulas.  Path parameters: None  Request body:   * a Json object that contains the input and output of a formula. Example:   {      "inputs": [{"name": "wheel", "number": 4}, {"name": "steel frame", "number": 1}],      "outputs": [{"name": "car", "number": 1}]  }  Response codes and body contents   * 201 – the formula is created.   Respond body:   * An object (array) that contains the id of the newly created formula. |

|  |
| --- |
| GET /formulas/:id  See the inputs and outputs of a specific formula.  Path parameters:   * id: non-negative integer, determines which formula you want to inspect.   Request body: None  Response codes and body contents   * 200 - all OK. * 400 - the formula does not exist.   Respond body:   * An object that contains the content of the formula (input and output). |

|  |
| --- |
| GET /plans/formulas/:id  List which plans contain a specific formula.  Path parameters:   * id: non-negative integer, determines which formula you are looking for from all the plans.   Request body: None  Response codes and body contents   * 200 - all OK.   Respond body:   * An object (array) that contains all the plans (id in number) that has this formula. |

|  |
| --- |
| POST /plans  Add a new plan.  Path parameters: None  Request body: None  Response codes and body contents   * 201 - a new plan is created.   Respond body:   * An object (array) that contains the id of the newly created plan. |

|  |
| --- |
| POST /plans/:id/formula  Append a formula to the end of a plan.  Path parameters:   * id: non-negative integer, determines which plan to add a formula to.   Request body:   * a Json object: key = ‘id’, value = formula id in integer. Example:   {      "id": 4  }  Response codes and body contents   * 200 - the formula is added to the end of a plan. * 400 - request body is empty || the plan does not exist || the formula does not exist || the formula exists but already in this plan.   Respond body: None |

|  |
| --- |
| PATCH /plans/:id/formula  Replace a formula anywhere in the sequence of formulas associated with a plan.  Path parameters:   * id: non-negative integer, determines which plan you want to access.   Request body:   * a Json object that contains: an object of old formula id (to be replaced) and an object of new formula id (the replacement). Example:   {      "oldId": 1,      "newId": 0  }  Response codes and body contents   * 200 - all OK. * 400 - the request body is empty or does not contain both old id and new id || the plan does not exist || the old formula (id) does not exist in the plan || the new formula (id) does not exist in the database (memory for this homework) || the new formula (id) is already in the plan.   Respond body: None |

|  |
| --- |
| GET /plans/:id  List all the formulas contained in a plan.  Path parameters:   * id: non-negative integer, determines which plan.   Request body: None  Response codes and body contents   * 200 - all OK. * 400 - this plan does not exist.   Respond body:   * An object (array) that contains all the formulas in this plan. |

|  |
| --- |
| DELETE /plans/:id  Delete a plan.  Path parameters:   * id: non-negative integer, determines which plan.   Request body: None  Response codes and body contents   * 204 - this plan is deleted. * 400 - this plan does not exist.   Respond body: None |