TEST

Vegan/s

# Assignment: RESTful Packaging Service

## Problem Description

Imagine you're given a package in which you are to place a number of given items. The items each have a given weight (in kg) and a given price (in €). There's a total of `n` items. Each item has a unique ID, ranging from `1` to `n`.

In general, the package will not be able to hold all of the given items. The reason is that there's a maxium weight the package can hold. It can only hold the items you place in the package if their total weight doesn't exceed this maximum weight.

It is now your job to pick a subset of the given items so that their total price is maximized and their combined weight doesn't exceed the maximum weight the package can hold.

Write a RESTful service that solves this problem. The service should have one endpoint which accepts a specification of the problem to solve, in particular

- the maximum weight the package can hold.
- a list of the items, each with their own ID, weight and price.

The service should verify that the following conditions are met:

- The maximum number of items is 15, i.e. `n <= 15`.
- The maximum weight of an item is 100 kg.
- The maximum price of an item is 100 €.
- The maximum weight the package can hold is 100 kg.

In case any one of these constraints is violated, an error message indicating the problem should be returned.

Otherwise, the service should return the IDs of those items that are to be placed in the package in order to maximize their total price, as described above.

If there's more than one combination of items having the same maximum price, then the IDs of the combination having the lowest total weight among them should be returned.

Don't forget to document the API.

## **Examples**

Here are a few examples that might help you better understand the problem.

### **Example 1**

Given a list of these 9 items

Item ID	Weight	Price
1	85.31 kg	29 €
2	14.55 kg	74 €
3	3.98 kg	16 €
4	26.24 kg	55 €
5	63.69 kg	52 €
6	76.25 kg	75 €
7	60.02 kg	74 €
8	93.18 kg	35 €
9	89.95 kg	78 €

and a maximum package weight of `75 kg`, then items `2` and `7` can be placed in the package for a combined weight of `74.57 kg` and a total price of `148 €`. This is the optimal solution given the above constraints. The service should thus return the IDs `2` and `7`.

Item ID	Weight	Price
1	53.38 kg	45 €
2	88.62 kg	98 €
3	78.48 kg	3 €
4	72.30 kg	76 €
5	30.18 kg	9 €
6	46.34 kg	48 €

and a maximum package weight of `80 kg`, the service should return the ID `4`, which is the optimal solution.

### Example 3

Given a list with this single item

Item ID	Weight	Price
1	15.3 kg	34€

and a maximum package weight of `8 kg` the service should return an error, indicating that no items can be placed in the package under the given constraints.

### Example 4

Given a list of the following 9 items

Item ID	Weight	Price
1	90.72 kg	13 €
2	33.80 kg	40 €
3	43.15 kg	10 €
4	37.97 kg	16 €
5	46.81 kg	36 €

6	48.77 kg	79 €
7	81.80 kg	45 €
8	19.36 kg	79 €
9	6.76 kg	64 €

and a maximum package weight of `56 kg` the service should return item IDs `8` and `9`, the optimal combination of items.

## Final Remarks

Feel free to use the attached [Spring Boot](https://spring.io/projects/spring-boot) skeleton project or implement the solution from scratch.

Apply best software design and development practices. Document your solution appropriately.

## How to submit your solution

- Develop on a branch and create a merge request once you are done.
- Send an email to your contact from the Vegan/S HR team, and we will have a look at your submission.

Good luck!