Container Optimizer API - Project Overview

This is a Spring Boot microservice designed for container-routing optimization.

It exposes an API endpoint (POST /pickSpot) to compute the best container move by analyzing a given yardMap.

The service calculates the lowest score spot, helping with efficient container placement in the yard.

API Endpoint: POST /pickSpot

Request Body:

```
{
  "yardMap": [
    [5, 8, 9],
    [4, 3, 6],
    [7, 2, 1]
  ]
}
```

This is a JSON format where the 'yardMap' represents the container yard as a 2D array.

The algorithm will calculate the lowest score by processing each container slot.

Response Example:

Response Body:

```
{
  "row": 2,
  "column": 2,
  "score": 1
}
```

The response returns the optimal row, column, and score of the container yard slot.

Service Logic

The PickSpotService finds the optimal container spot by looping through the yardMap

and comparing the scores. It returns the row, column, and score of the optimal spot.

The algorithm considers all available slots and finds the one with the lowest score.

Code Snippets

```
// PickSpotRequest.java - Model class to map the request JSON
public class PickSpotRequest {
    private List<List<Integer>> yardMap;
    // Getters and setters for yardMap
}

// PickSpotController.java - Controller class to handle POST requests
@RestController
public class PickSpotController {
    @PostMapping("/pickSpot")
    public PickSpotResponse pickSpot(@RequestBody PickSpotRequest request) {
        return pickSpotService.findOptimalSpot(request);
    }
}
```

```
// PickSpotService.java - Business logic to find the optimal spot
public PickSpotResponse findOptimalSpot(PickSpotRequest request) {
    int[][] yardMap = convertListToArray(request.getYardMap());
    // Loop through yardMap to find the lowest score and return the optimal spot
}
```