

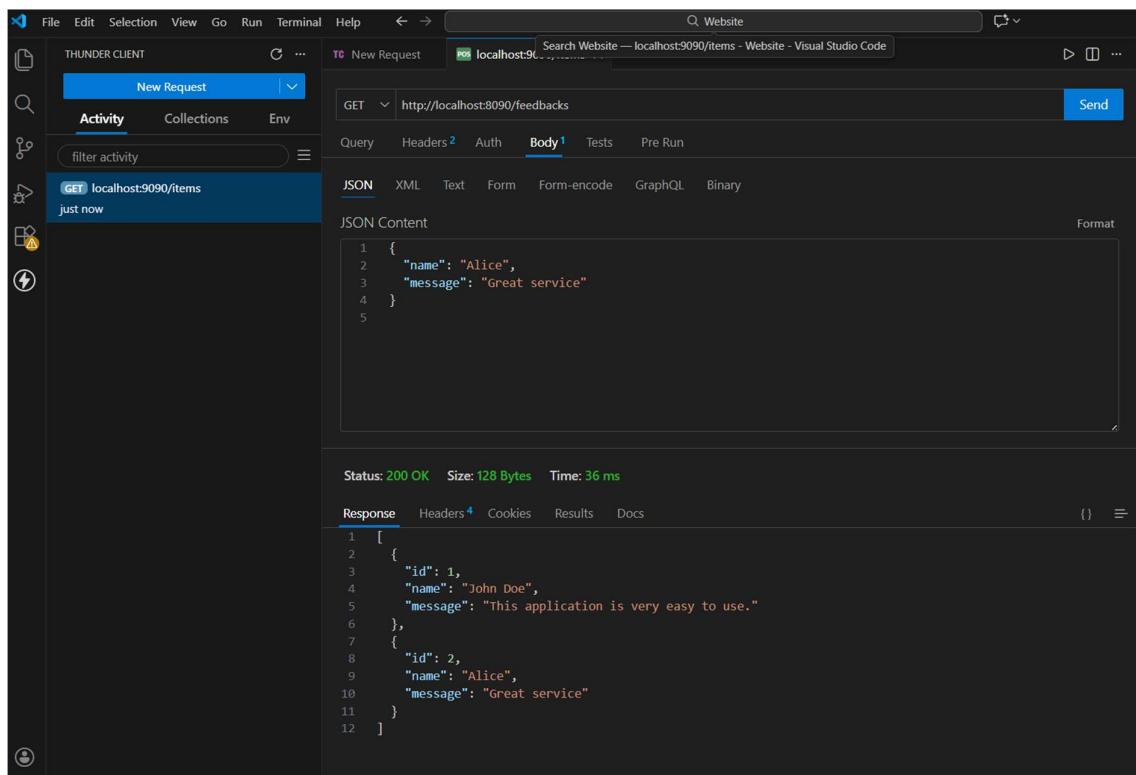
Roll No – 202401100400107

Name – Keshav Agarwal

Github – <https://github.com/Virtual-box-KA/java-ese>

Screenshots of code –

1. Post

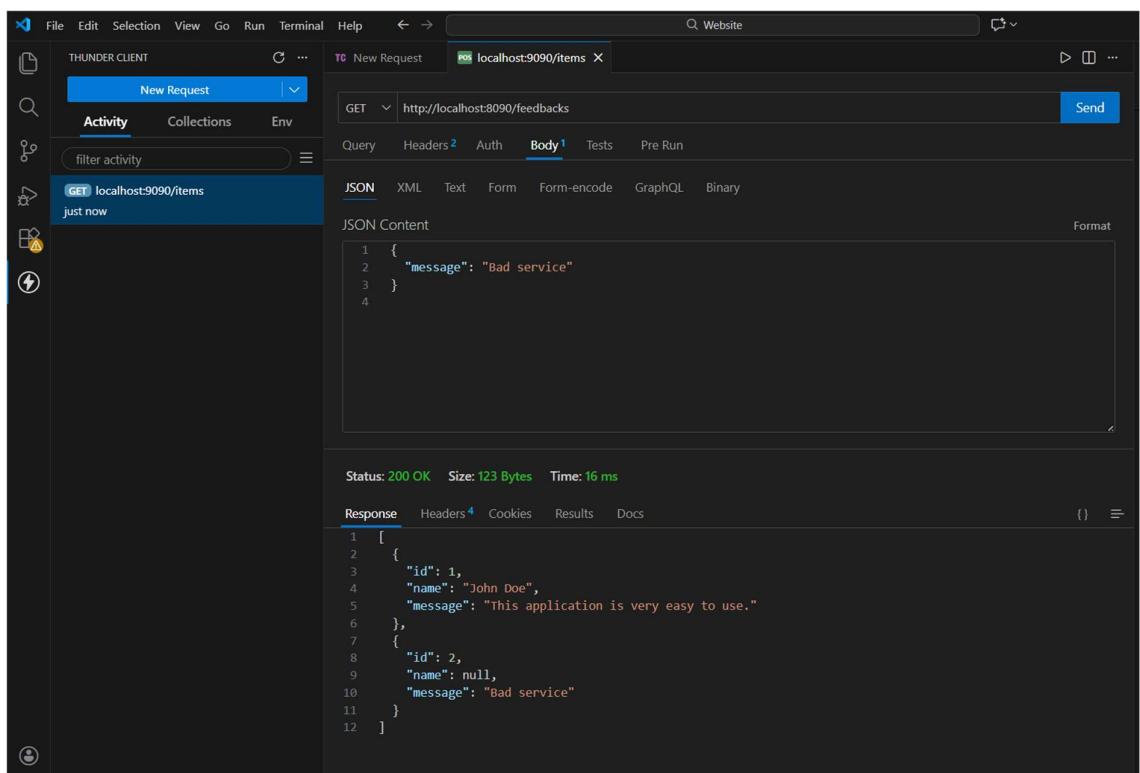


The screenshot shows the Thunder Client interface. A new request is being made to `http://localhost:8090/feedbacks` using a `POST` method. The `Body` tab is selected, displaying the following JSON content:

```
1 {
2   "name": "Alice",
3   "message": "Great service"
4 }
```

The response status is `200 OK`, size is `128 Bytes`, and time is `36 ms`. The response body shows a list of two items:1 [
2 {
3 "id": 1,
4 "name": "John Doe",
5 "message": "This application is very easy to use."
6 },
7 {
8 "id": 2,
9 "name": "Alice",
10 "message": "Great service"
11 }
12]

2. Get

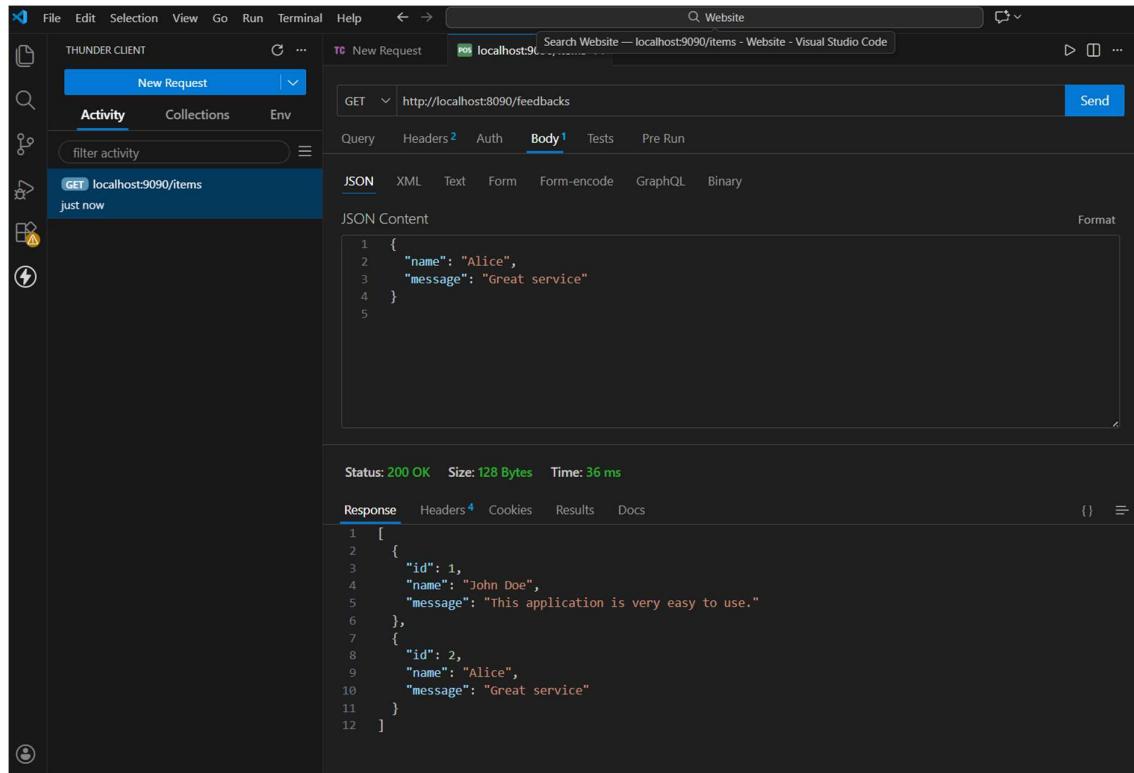


The screenshot shows the Thunder Client interface. A new request is being made to `http://localhost:8090/items` using a `GET` method. The `Body` tab is selected, displaying the following JSON content:

```
1 {
2   "message": "Bad service"
3 }
```

The response status is `200 OK`, size is `123 Bytes`, and time is `16 ms`. The response body shows a list of two items:1 [
2 {
3 "id": 1,
4 "name": "John Doe",
5 "message": "This application is very easy to use."
6 },
7 {
8 "id": 2,
9 "name": null,
10 "message": "Bad service"
11 }
12]

3. Post



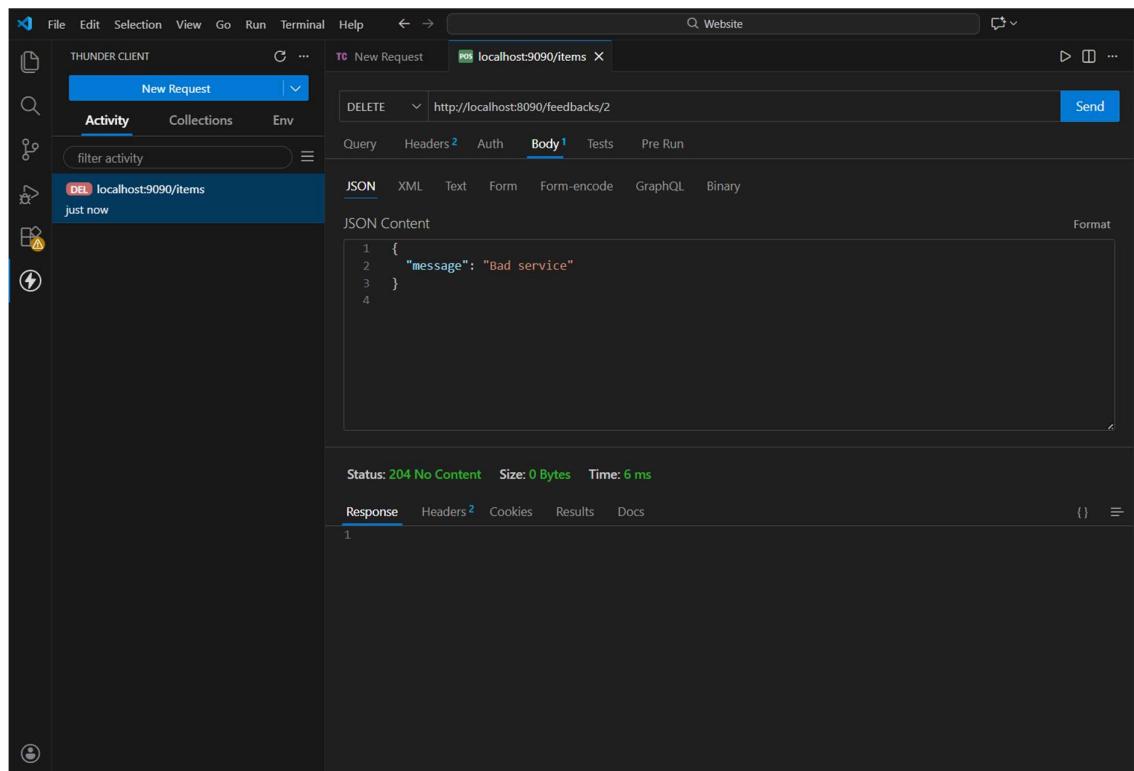
The screenshot shows the Thunder Client interface. A new request is being made to `http://localhost:8090/feedbacks` using the `POST` method. The `Body` tab is selected, displaying the following JSON content:

```
1 {  
2   "name": "Alice",  
3   "message": "Great service"  
4 }  
5
```

The response status is `200 OK`, size is `128 Bytes`, and time is `36 ms`. The response body contains a list of two items:

```
1 [  
2   {  
3     "id": 1,  
4     "name": "John Doe",  
5     "message": "This application is very easy to use."  
6   },  
7   {  
8     "id": 2,  
9     "name": "Alice",  
10    "message": "Great service"  
11  }  
12 ]
```

4. Delete



The screenshot shows the Thunder Client interface. A new request is being made to `http://localhost:8090/feedbacks/2` using the `DELETE` method. The `Body` tab is selected, displaying the following JSON content:

```
1 {  
2   "message": "Bad service"  
3 }  
4
```

The response status is `204 No Content`, size is `0 Bytes`, and time is `6 ms`. The response body is empty.

Code –

1. Controller –

```
package com.ese.Feedback1.controller;

import com.ese.Feedback1.model.Feedback;
import com.ese.Feedback1.service.FeedbackService;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/feedbacks")
public class FeedbackController {

    private final FeedbackService feedbackService;

    public FeedbackController(FeedbackService feedbackService) {
        this.feedbackService = feedbackService;
    }

    // POST /feedbacks
    @PostMapping
    public ResponseEntity<Feedback> createFeedback(@RequestBody
Feedback feedback) {
        return
ResponseEntity.ok(feedbackService.createFeedback(feedback));
    }

    // GET /feedbacks
    @GetMapping
    public ResponseEntity<List<Feedback>> getAllFeedbacks() {
        return ResponseEntity.ok(feedbackService.getAllFeedbacks());
    }

    // PUT /feedbacks/{id}
    @PutMapping("/{id}")
    public ResponseEntity<Feedback> updateFeedback(@PathVariable Long
id,
                                                @RequestBody
Feedback feedback) {
        Feedback updated = feedbackService.updateFeedback(id,
feedback);
        if (updated == null) {
            return ResponseEntity.notFound().build();
        }
        return ResponseEntity.ok(updated);
    }

    // DELETE /feedbacks/{id}
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteFeedback(@PathVariable Long id)
{
        if (feedbackService.deleteFeedback(id)) {
            return ResponseEntity.noContent().build();
        }
        return ResponseEntity.notFound().build();
    }
}
```

2. Model-

```
package com.ese.Feedback1.model;

public class Feedback {

    private Long id;
    private String name;
    private String message;

    public Feedback() {}

    public Feedback(Long id, String name, String message) {
        this.id = id;
        this.name = name;
        this.message = message;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

3. View –

```
package com.ese.Feedback1.service;

import com.ese.Feedback1.model.Feedback;
import org.springframework.stereotype.Service;

import java.util.*;
import java.util.concurrent.atomic.AtomicLong;

@Service
public class FeedbackService {

    private final Map<Long, Feedback> feedbackStore = new
HashMap<>();
    private final AtomicLong idCounter = new AtomicLong(1);

    // CREATE
    public Feedback createFeedback(Feedback feedback) {
        Long id = idCounter.getAndIncrement();
        feedback.setId(id);
        feedbackStore.put(id, feedback);
        return feedback;
    }

    // READ ALL
    public List<Feedback> getAllFeedbacks() {
        return new ArrayList<>(feedbackStore.values());
    }

    // UPDATE
    public Feedback updateFeedback(Long id, Feedback feedback) {
        if (!feedbackStore.containsKey(id)) {
            return null;
        }
        feedback.setId(id);
        feedbackStore.put(id, feedback);
        return feedback;
    }

    // DELETE
    public boolean deleteFeedback(Long id) {
        return feedbackStore.remove(id) != null;
    }
}
```