



Project Explorer ×

▼ Pentonix

- > JRE System Library [JavaSE-7]
- ▼ src
 - ▼ Assesment
 - > Task2.java
 - > Task3.java
 - > module-info.java
 - pom.xml

Task2.java × Task3.java

```
1 package Assesment;
2
3 import com.sun.net.httpserver.HttpServer;
4
5
6
7
8
9
10
11
12
13 public class Task2
14 {
15     public static class Employee
16     {
17         private int eno;
18         private String ename;
19         private int dno;
20         private int salary;
21
22         public Employee(int eno, String ename, int dno, int salary)
23         {
24             this.eno = eno;
25             this.ename = ename;
26             this.dno = dno;
27             this.salary = salary;
28         }
29
30         public int getEno()
31         {
32             return eno;
33         }
34
35         public String getEname()
36         {
37             return ename;
38         }
39
40         public int getDno()
41         {
42             return dno;
43         }
44
45         public int getSalary()
46         {
47             return salary;
48         }
49     }
50
51     public static class Department
52     {
53         private int dno;
54         private String dname;
```

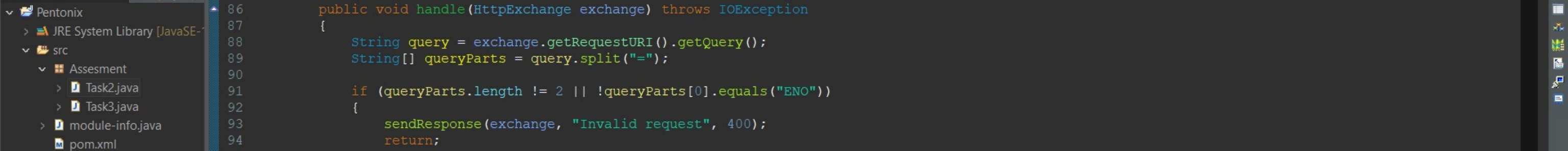


Project Explorer

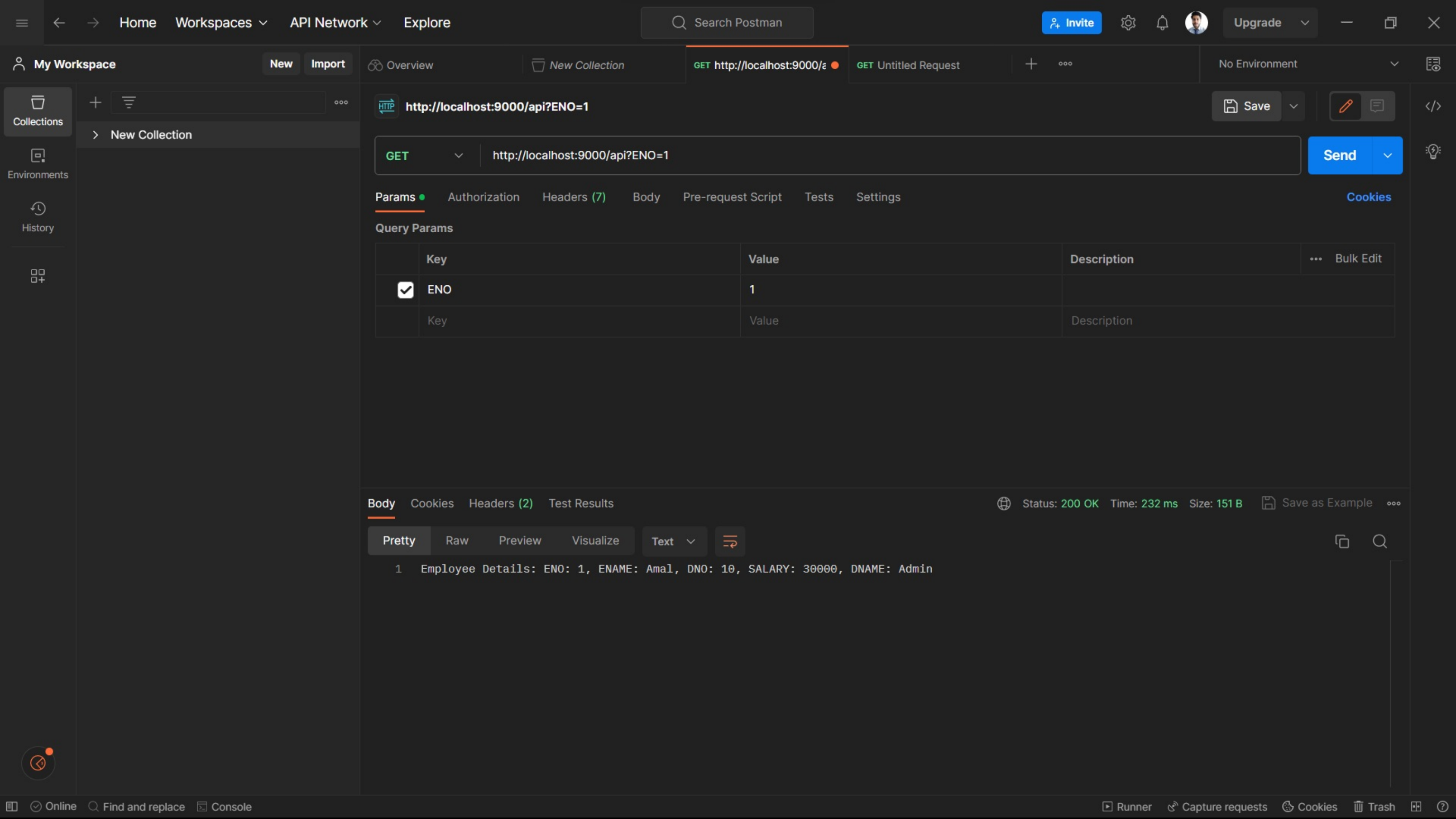
Pentonix

- > JRE System Library [JavaSE-7]
- > src
 - > Assesment
 - > Task2.java
 - > Task3.java
 - > module-info.java
 - pom.xml

```
Task2.java x Task3.java
54     private String dname;
55     public Department(int dno, String dname)
56     {
57         this.dno = dno;
58         this.dname = dname;
59     }
60     public int getDno()
61     {
62         return dno;
63     }
64
65     public String getDname()
66     {
67         return dname;
68     }
69 }
70 private static Map<Integer, String> departmentDatabase = new HashMap<>();
71 private static Map<Integer, Employee> employeeDatabase = new HashMap<>();
72
73 public static void main(String[] args) throws IOException
74 {
75     // Create an HTTP server
76     HttpServer server = HttpServer.create(new InetSocketAddress(9000), 0);
77     server.createContext("/api", new DepartmentHandler());
78     server.start();
79
80     System.out.println("Server started on port 9000...");
81 }
82
83 static class DepartmentHandler implements HttpHandler
84 {
85     @Override
86     public void handle(HttpExchange exchange) throws IOException
87     {
88         String query = exchange.getRequestURI().getQuery();
89         String[] queryParts = query.split("=");
90
91         if (queryParts.length != 2 || !queryParts[0].equals("ENO"))
92         {
93             sendResponse(exchange, "Invalid request", 400);
94             return;
95         }
96
97         int eno = Integer.parseInt(queryParts[1]);
98         Employee employee = employeeDatabase.get(eno);
99     }
```



```
85  @Override
86  public void handle(HttpExchange exchange) throws IOException
87  {
88      String query = exchange.getRequestURI().getQuery();
89      String[] queryParts = query.split("=");
90
91      if (queryParts.length != 2 || !queryParts[0].equals("ENO"))
92      {
93          sendResponse(exchange, "Invalid request", 400);
94          return;
95      }
96
97      int eno = Integer.parseInt(queryParts[1]);
98      Employee employee = employeeDatabase.get(eno);
99
100     if (employee != null)
101     {
102         String departmentName = departmentDatabase.get(employee.getDno());
103         Department department = new Department(employee.getDno(), departmentName);
104
105         sendResponse(exchange,
106             "Employee Details: ENO: " + employee.getEno()
107             + ", SALARY: " + employee.getSalary()
108             + ", DNAME: " + department.getDname()
109             + ", ENAME: " + employee.getEname()
110             + ", DNO: " + employee.getDno()
111             , 200);
112     } else
113     {
114         sendResponse(exchange, "Employee not found", 404);
115     }
116 }
117
118 private void sendResponse(HttpExchange exchange, String response, int statusCode) throws IOException
119 {
120     exchange.getResponseHeaders().set("Content-Type", "text/plain");
121     OutputStream os = exchange.getResponseBody();
122     os.write(response.getBytes());
123     os.close();
124 }
125 }
126
127
128 }
129
130
```

My Workspace

NewImport

Collections

Environments

History

+

New Collection

Overview

New Collection

GET http://localhost:9000/api?ENO=1

GET Untitled Request

+⋮

No Environment ▾

Save ▾

HTTP

http://localhost:9000/api?ENO=1

Send ▾

Params ●

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/>	ENO	1		
	Key	Value	Description	

Body

Cookies

Headers (2)

Test Results

🌐 Status: 200 OK Time: 232 ms Size: 151 B Save as Example ⋮

Pretty

Raw

Preview

Visualize

Text ▾

1 Employee Details: ENO: 1, ENAME: Amal, DNO: 10, SALARY: 30000, DNAME: Admin