

Session Notes 2 – ECSE 429 (Interoperability): Marcello

Charter

- Test how todos, projects, and categories work together through relationships
- Identify interoperability issues and data consistency problems
- Test cross-entity operations and relationship management
- Verify bidirectional relationship integrity across all entity types

Build

- runTodoManagerRestAPI-1.5.5.jar

Area

- INTEROPERABILITY between todos, projects, and categories

Test Environment

- macOS
- Postman v11
- API running locally via `java -jar runTodoManagerRestAPI-1.5.5.jar`

Date & Time

- October 1st, 2025 - Start time: 9:28pm
- Session Duration: 45 minutes

Tester

- Marcello Audino – [261029024] - [marcello.audino@mail.mcgill.ca]

9:28 PM

Starting interoperability testing session. Preparing clean test data.

9:30 PM

Created test data: Todo 29 “InteropCatTodo” (POST /todos 201), Category 7 “C_Interop” (POST /categories 201), Project 3 “P_Interop” (POST /projects 201).

9:32 PM

GET /todos/29/categories returned 200 OK with an empty list.

9:33 PM

POST /todos/29/categories with body { "id": "7" } returned 201 Created. Link between todo and category established.

9:34 PM

GET /categories/7/todos returned 200 OK and includes todo 29.

9:36 PM

GET /todos/29/tasksof returned 200 OK with an empty list.

9:37 PM

POST /todos/29/tasksof with body { "id": "3" } returned 201 Created. Link between todo and project established.

9:38 PM

GET /projects/3/tasks returned 200 OK and includes todo 29.

9:45 PM

GET /todos/29/categories returned 200 OK and shows category 7.

GET /todos/29/tasksof returned 200 OK and shows project 3.

9:49 PM

DELETE /projects/3 returned 200 OK.

GET /todos/29 returned 200 OK. Todo still present after project deletion.

9:51 PM

DELETE /categories/7 returned 200 OK.

GET /projects returned 200 OK and project 3 is gone.

GET /todos returned 200 OK and todo 29 remains.

9:54 PM

OPTIONS /projects returned 200 OK with allowed methods in headers.

HEAD /todos returned 200 OK with headers only and no body.

Session Summary Findings

Key Issues

- Deleting a project or a category did not remove the linked todo. Todo 29 remained after deleting project 3 and category 7
- IDs were not reused after deletions, leaving gaps
- Error message wording varied across endpoints
- Parent–child links were not always obvious in a single response and often required additional GETs

Capabilities Identified

- Todo–category and todo–project links were created and retrieved in both directions using GET /todos/29/categories, GET /categories/7/todos, GET /todos/29/tasksof, and GET /projects/3/tasks
- HEAD and OPTIONS worked on the relevant endpoints and returned appropriate headers
- Linking by id on the relationship endpoints worked as expected

New Test Ideas

- Bulk create, link, unlink, and delete across many todos, categories, and projects to observe consistency at scale
- Re-link an orphaned todo to a new project or category and verify all reverse lookups
- Send malformed or duplicate relationship payloads to confirm validation behavior

Areas of Potential Risk

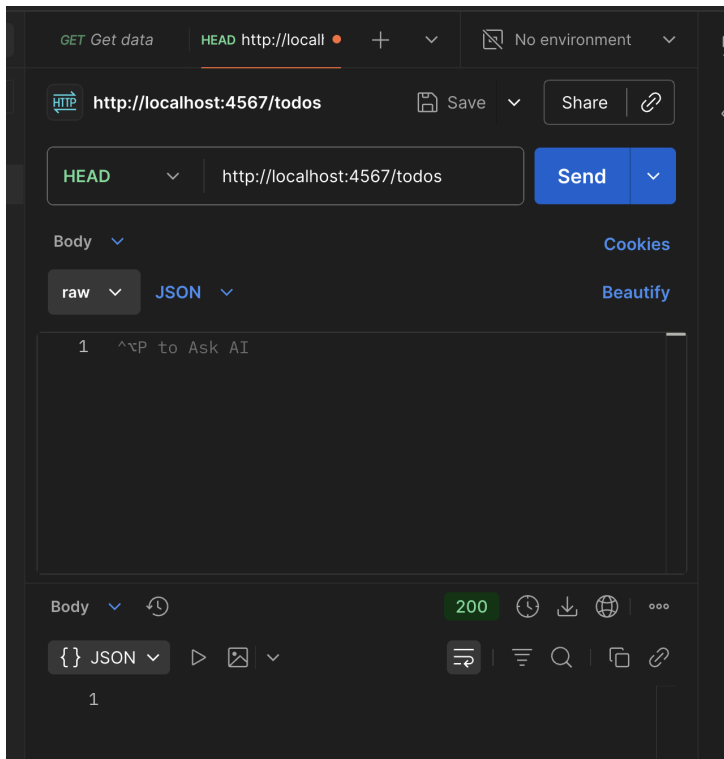
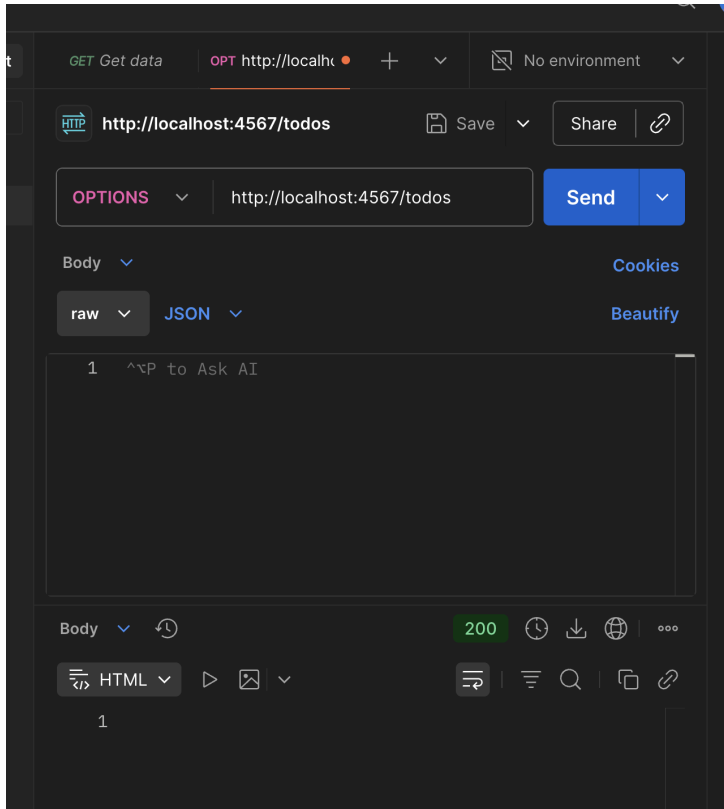
- Orphaned records after deletes can lead to inconsistent data and confusing state
- Inconsistent error wording can slow debugging and automation
- Needing multiple requests to confirm relationships increases integration complexity

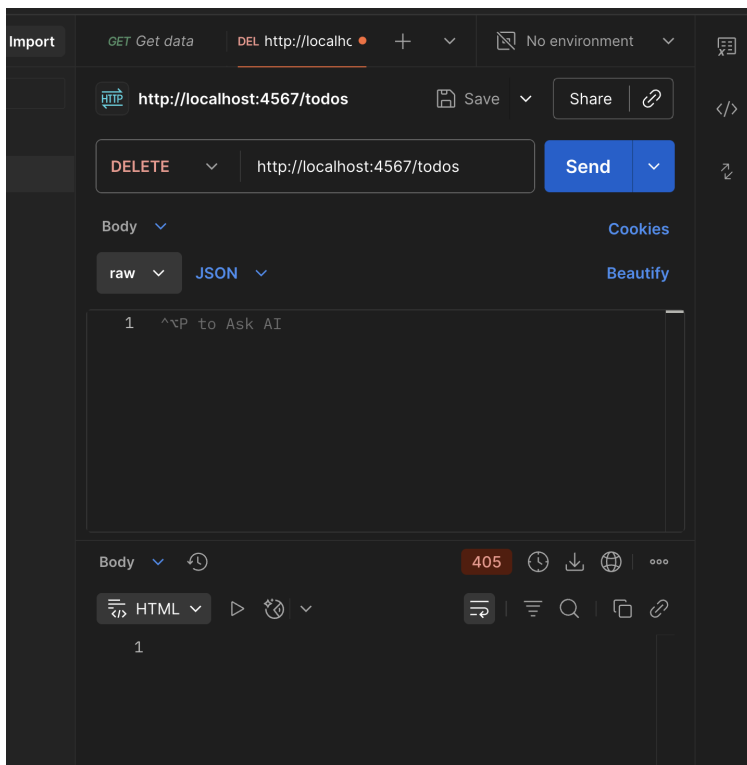
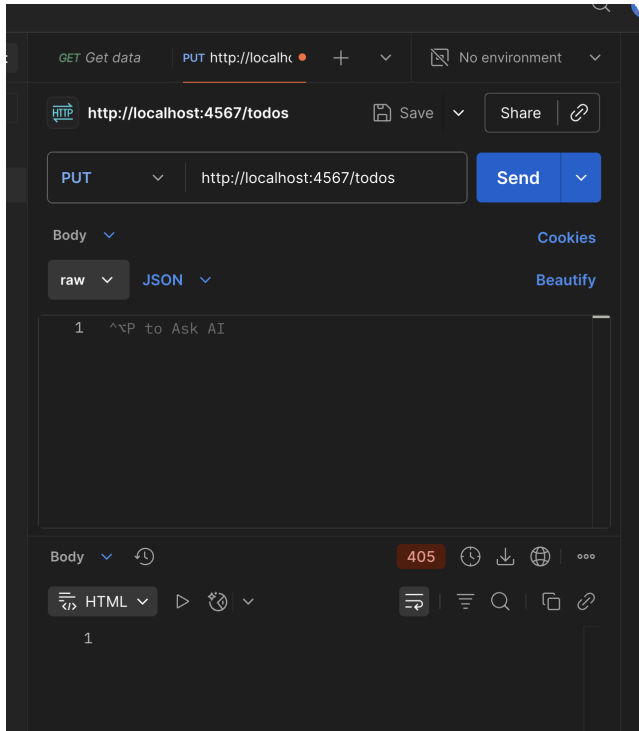
Decisions Made

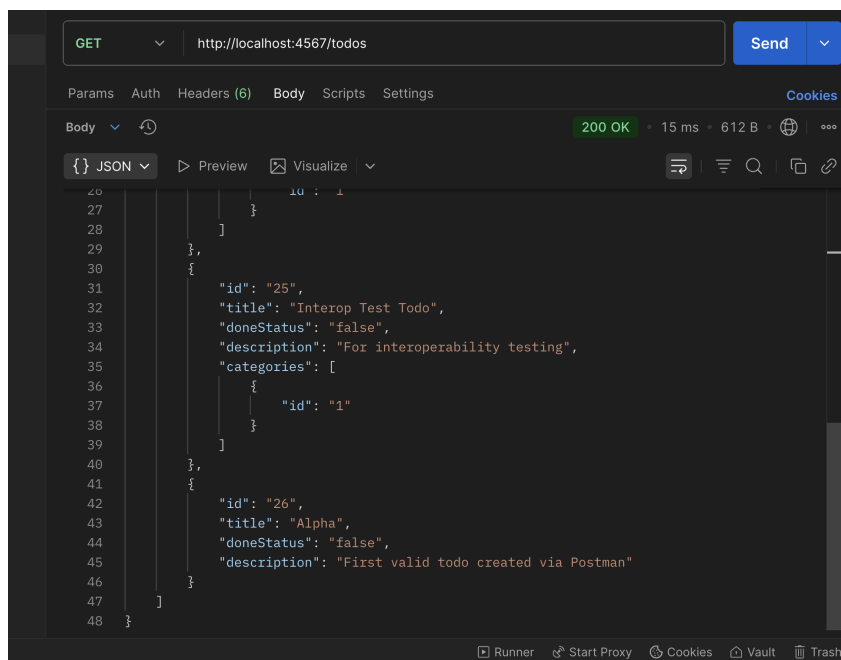
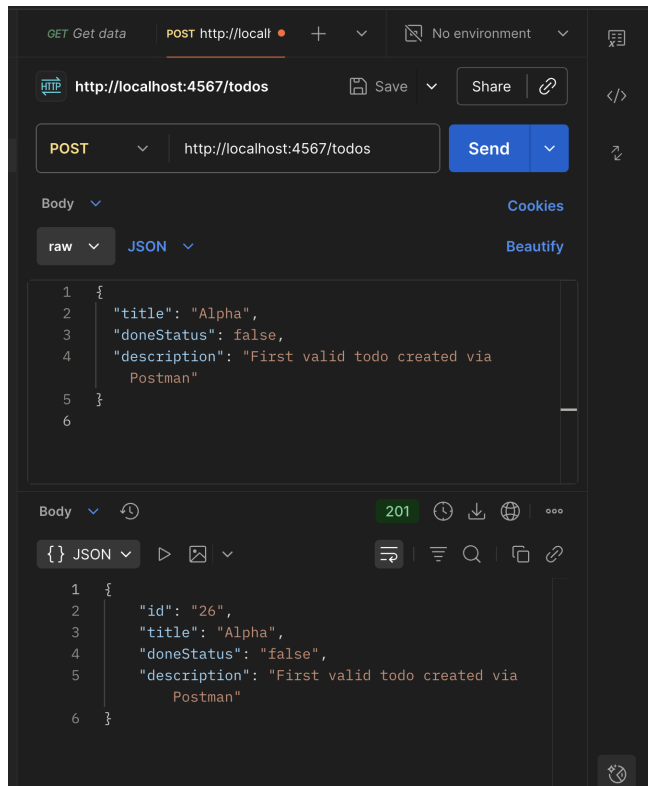
- Focus next on delete behavior and re-linking orphans and capture exact error texts for inconsistent cases
- Continue documenting with both forward and reverse lookups, including GET /projects/{id}/tasks

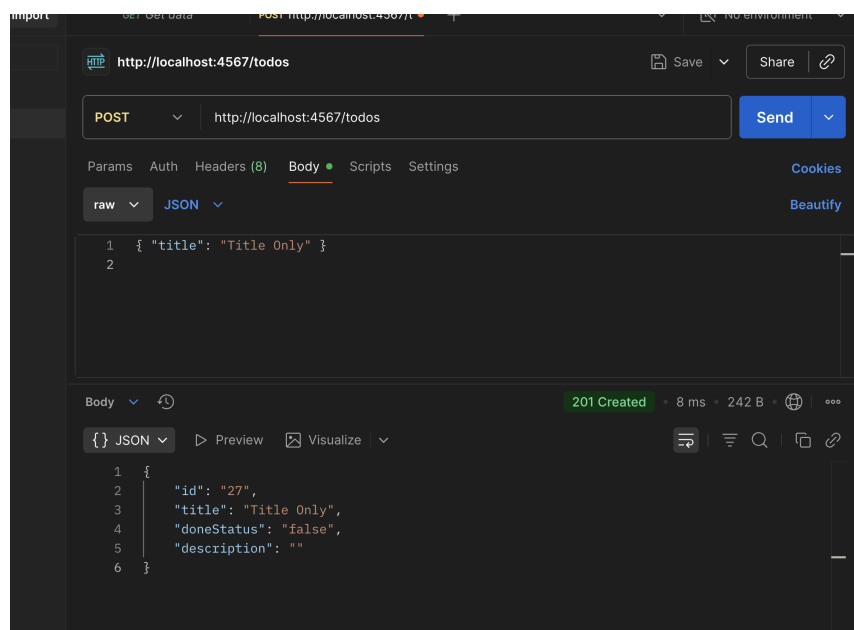
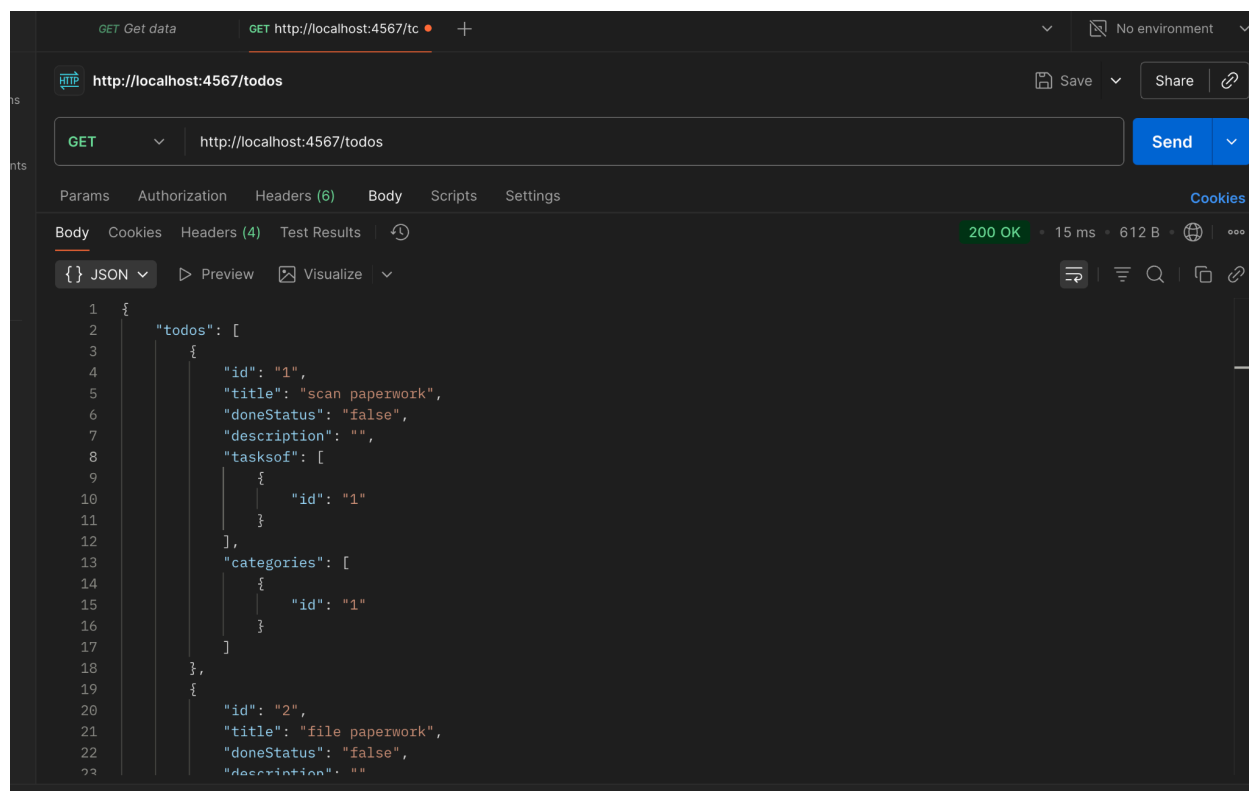
Files Referenced

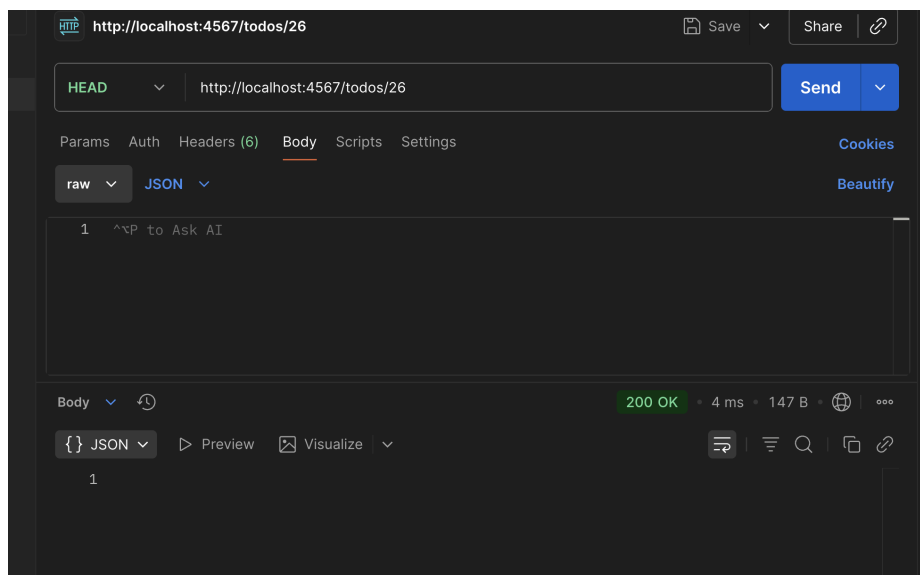
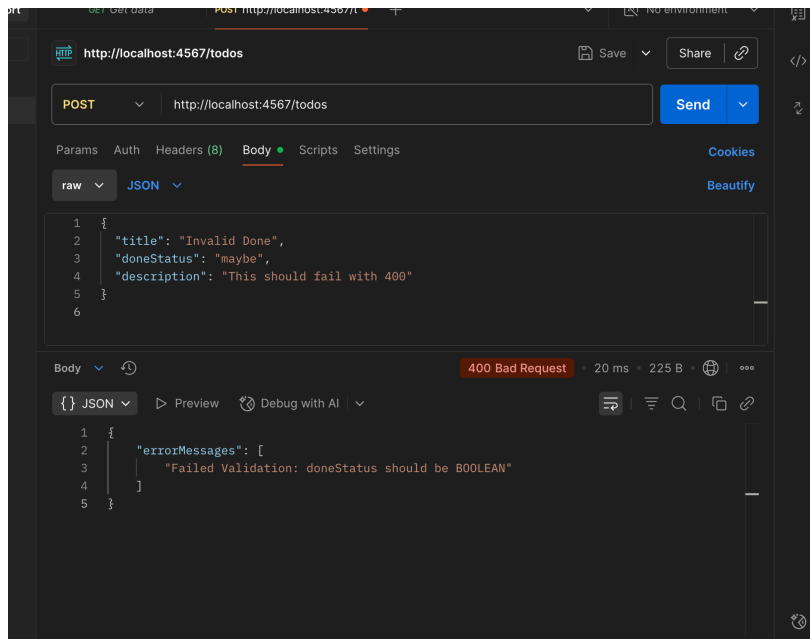
- Postman screenshots for create and link operations, reverse lookups, delete checks, and HEAD or OPTIONS
- Postman collection used for the interoperability tests











HTTP <http://localhost:4567/todos/26> Save Share

GET ▼ <http://localhost:4567/todos/26> Send ▼

Params Auth Headers (6) **Body** Scripts Settings Cookies Beautify

raw ▼ JSON ▼

```
1 ^XP to Ask AI
```

Body ▼ ↺ 200 OK • 21 ms • 260 B • 🌐 ⋮

{} JSON ▼ ▶ Preview 🖼 Visualize ▼ ↺ ☰ 🔍 📄 🔗

```
1 {
2   "todos": [
3     {
4       "id": "26",
5       "title": "Alpha",
6       "doneStatus": "false",
7       "description": "First valid todo created via Postman"
8     }
9   ]
10 }
```

HTTP <http://localhost:4567/todos/26> Save ▼ S

PUT ▼ <http://localhost:4567/todos/26> S

Params Auth Headers (8) **Body** ● Scripts Settings

raw ▼ JSON ▼

```
1 {
2   "title": "Alpha (Updated)",
3   "description": "Updated via PUT"
4 }
5
```

Body ▼ ↺ 200 OK • 8 ms • 237 B

{} JSON ▼ ▶ Preview 🖼 Visualize ▼ ↺ ☰ 🔍 📄 🔗

```
1 {
2   "id": "26",
3   "title": "Alpha (Updated)",
4   "doneStatus": "false",
5   "description": "Updated via PUT"
6 }
```

