Here's a **simple end-to-end case study** for Camunda 8 that demonstrates the use of:

- **Message-driven flows**
- **Send and Receive Tasks**
- **User Task with External Angular Form**
- **Java Client (Camunda SDK) to interact with process**
- **Modelled using Camunda SaaS Web Modeler**

🎯 Case Study: Loan Application Approval Process

Actors:

- Applicant (fills form via Angular UI)
- Loan Officer (reviews and approves/rejects)

✅ Process Overview (Modeled in Camunda Web Modeler - SaaS)

BPMN Elements:

Start Event → Send Task → Receive Task → User Task (External Form) → Exclusive Gateway

→ [Approved] → End Event

→ [Rejected] → End Event

🔁 Flow Explanation:

1. **Send Task**: Triggers a message (e.g., loanApplicationRequest) via Java client (simulating backend submission).
2. **Receive Task**: Waits for an external system to respond (e.g., credit score system or notification).
3. **User Task**: Loan Officer reviews loan using **Angular Form** (connected via Tasklist API).
4. **Gateway**: Based on officer input, routes to approval or rejection path.

🔧 Implementation Details

1️⃣ Model in Camunda Web Modeler (SaaS)

- **Start Event**

- **Send Task**: Set to send message loanApplicationRequest
- **Receive Task**: Waits for loanApplicationResponse
- **User Task**: Assign form key (e.g., embedded:app:forms/loan-review-form.html)
- **Gateway**: approved == true
- **End Events**: Named accordingly

2️⃣ Java Client Logic (Using Camunda SDK)

```
ZeebeClient client = ZeebeClient.newClientBuilder()
        .gatewayAddress("your-camunda-cloud.broker-address:443")
        .usePlaintext()
        .build();

// Send message to trigger Receive Task
client.newPublishMessageCommand()
    .messageName("loanApplicationResponse")
    .correlationKey("loan123")
    .variables("{\"creditScore\": 750}")
    .send()
    .join();

// Start process instance
client.newCreateInstanceCommand()
    .bpmnProcessId("loan_approval_process")
    .latestVersion()
    .variables(Map.of(
       "loanId", "loan123",
       "applicantName", "John Doe",
       "loanAmount", 50000
```

```
  ))
  .send()
  .join();
```

3️⃣ Angular External Form (For User Task)

```
<!-- loan-review-form.component.html -->
<form (ngSubmit)="submit()" *ngIf="task">
 <label>Name: {{ task.variables.applicantName }}</label><br>
 <label>Amount: {{ task.variables.loanAmount }}</label><br>

 <label>Approve?</label>
 <select [(ngModel)]="formData.approved" name="approved">
  <option [value]="true">Yes</option>
  <option [value]="false">No</option>
 </select><br>

 <button type="submit">Submit</button>
</form>
// loan-review-form.component.ts
submit() {
 this.taskService.completeTask(this.task.id, {
  variables: {
   approved: { value: this.formData.approved, type: "Boolean" }
  }
 }).subscribe(() => alert("Task completed"));
}
```

Use **Camunda Tasklist REST API** or **Camunda Form JS SDK** to load/complete the task.

🔌 Required Integration Points

Component	Details
Camunda Web Modeler	For designing and deploying process
Zeebe Java Client	Start process instance, send/receive message
Angular App	Hosts external user task form
Camunda Tasklist API	Used by Angular app to fetch & complete user task

---

✅ Outcome

This flow demonstrates:

- **Message initiation via Java client**
- **Message correlation for Receive Task**
- **Human Task completion via Angular external form**
- **Decision branching based on form data**

📦 Optional Enhancements

- Store form data in PostgreSQL
- Add retry/boundary events for Receive Task
- Integrate email notifications