

## Fullstack Engineer Assessment

# backend

## DEVELOPMENT

**2-3 hrs**  
*ESTIMATED EFFORT*

**24-36 hrs**  
*EXPECTED DEADLINE*



# Backend Development Assessment

## Scenario

You're building a service that receives transaction webhooks from external payment processors (like RazorPay). Your service needs to acknowledge webhooks immediately and process transactions reliably in the background.

## Requirements

### 1. API Details

Create an endpoint that accepts transaction webhooks:

**Webhook Endpoint:** POST /v1/webhooks/transactions

**Request Body:**

```
{
  "transaction_id": "txn_abc123def456",
  "source_account": "acc_user_789",
  "destination_account": "acc_merchant_456",
  "amount": 1500,
  "currency": "INR"
}
```

Create a health check endpoint:

**Health Check Endpoint:** GET / (root path)

**Response:**

```
{
  "status": "HEALTHY",
  "current_time": "2024-01-15T10:30:00Z",
}
```

Create an endpoint to retrieve transaction status *[for testing]*:

**Query Endpoint:** GET /v1/transactions/{transaction\_id}

**Response:**

```
[{
  "transaction_id": "txn_abc123def456",
  "source_account": "acc_user_789",
  "destination_account": "acc_merchant_456",
  "amount": 150.50,
  "currency": "USD",
  "status": "PROCESSED", // or "PROCESSING"
  "created_at": "2024-01-15T10:30:00Z",
  "processed_at": "2024-01-15T10:30:30Z" // null if still processing
}]
```



# Backend Development Assessment

## 2. Response Requirements

- Must return 202 Accepted status code
- Must respond within 500ms regardless of processing complexity
- Response body can be empty or contain a simple acknowledgment

## 3. Background Processing

- Process each transaction after receiving the webhook
- Processing must include a 30-second delay (simulate external API calls)
- Store the final result in persistent storage

## 4. Idempotency

- Multiple webhooks with the same transaction\_id should only result in one processed transaction
- Your service should handle this gracefully without errors or duplicate processing

## 5. Data Storage

Store processed transactions with their status and timing information.

## Success Criteria 🎯

Your solution will be tested with:

1. **Single Transaction:** Send one webhook → verify it's processed after ~30 seconds
2. **Duplicate Prevention:** Send the same webhook multiple times → verify only one transaction is processed
3. **Performance:** Webhook endpoint responds quickly even under processing load
4. **Reliability:** Service handles errors gracefully and doesn't lose transactions

## Technical Freedom 🚀

- Use any **Python** framework
- Use any **cloud-based database**
- Use any **libraries or managed services** that help you build quickly and reliably
- Deploy **service on the cloud**

## Deliverables 📦

- **Public GitHub repo** of working Python application. Include the following in README
  - Instructions to run and test your service
  - Brief explanation of your technical choices
- ☒ **Public API endpoint** deployed on cloud

