**Data Synchronization Project Plan Extension: Invoice Catalog Integration**

**Project Overview:**

This extension aims to integrate an invoice catalog into existing data synchronization system between Stripe, and Django application.

**Project Phases:**

**Phase 1: Data Model and API Setup**

* 1. Invoice Model
* Create an Invoice model in your Django application with a foreign key relationship to the Customer model.

1.2 Invoice Outward Sync API

* Implement an outward sync API for invoices in your Django application.
* The API should trigger the producer to monitor changes in the Invoice model.
* The producer will add events to the existing queue for outward sync.

1.3 Invoice Inward Sync API

* Create an inward sync API endpoint for invoices.
* Expose this endpoint as a webhook in Stripe, with the event type "invoice.created."
* Implement a function to process data received from the Stripe webhook and create/update invoices in your Django application.

**Phase 2: Queue and Processing**

2.1 Queue System Extension

* Create a queue to handle invoice events.
* Create producer to push invoice-related events to the queue.

2.2 Invoice Consumer

* Create a dedicated consumer for processing invoice events from the queue.
* Implement logic to make API calls to Stripe for invoice synchronization.
* Ensure proper error handling and logging for invoice-related operations.

**Phase 3: Testing and Deployment**

3.1 Testing

* Perform extensive testing of the entire synchronization workflow, including the new invoice integration.
* Verify data integrity and synchronization between Stripe, invoices, and your Django application.