Yuki Integration MVP - Technical Documentation

---

1. Introduction

This document provides an overview of the integration between Yuki and Billify. It outlines the implementation steps, API endpoints, data flow, and synchronisation processes for retrieving purchase and sales invoices.

---

2. System Overview

2.1 Objective

The goal of this integration is to connect Billify with Yuki to:

- Retrieve purchase and sales invoices.
- Sync invoice payment statuses.
- Store essential invoice data in Billify's database.
- Update invoice data periodically.

2.2 Tech Stack

- **Backend:** Django (Django REST Framework)
- **Database:** PostgreSQL
- **Task Queue:** Celery (for periodic synchronization)
- **Authentication:** API Key-based authentication with Yuki

---

3. Yuki API Integration

3.1 Authentication

To interact with Yuki's API, authenticate using an **API key**. The key is used to generate a session ID for subsequent requests.

Endpoint:
POST https://api.yukiworks.be/ws/Sales.asmx

Authentication Request (XML)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:they="http://www.theyukicompany.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <they:Authenticate>
         <they:accessKey>{{API_KEY}}</they:accessKey>
      </they:Authenticate>
   </soapenv:Body>
</soapenv:Envelope>
```

Response:

Returns a session ID valid for 24 hours.

---

4. Fetching Invoice Data

4.1 Retrieve Sales Invoices

Endpoint:

```
POST https://api.yukiworks.be/ws/Sales.asmx
```

Request XML:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <GetSalesInvoices>
         <sessionID>{{SESSION_ID}}</sessionID>
      </GetSalesInvoices>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Fields:

- Invoice Number
- Issue Date
- Due Date
- Total Amount
- Payment Status

4.2 Retrieve Purchase Invoices

Endpoint:
POST https://api.yukiworks.be/ws/Purchase.asmx

Request XML:
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <GetPurchaseInvoices>
         <sessionID>{{SESSION_ID}}</sessionID>
      </GetPurchaseInvoices>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Fields:

- Invoice Number
- Supplier Name
- Total Amount
- Payment Status

---

5. Data Management

5.1 Database Schema

Invoices Table
```
CREATE TABLE invoices (
   id SERIAL PRIMARY KEY,
   invoice_number VARCHAR(50) UNIQUE,
   type VARCHAR(20), -- 'purchase' or 'sales'
   issue_date DATE,
   due_date DATE,
   total_amount DECIMAL(10,2),
   payment_status VARCHAR(20),
   yuki_invoice_id VARCHAR(50) UNIQUE
);
```

5.2 Duplicate Prevention

- Before inserting an invoice, check if `yuki_invoice_id` exists.
- If found, update the existing record instead of inserting a new one.

5.3 Periodic Synchronization

A scheduled task (Celery) runs every minute to:

1. Fetch sales and purchase invoices from Yuki.
2. Update the Billify database with the latest invoice data.

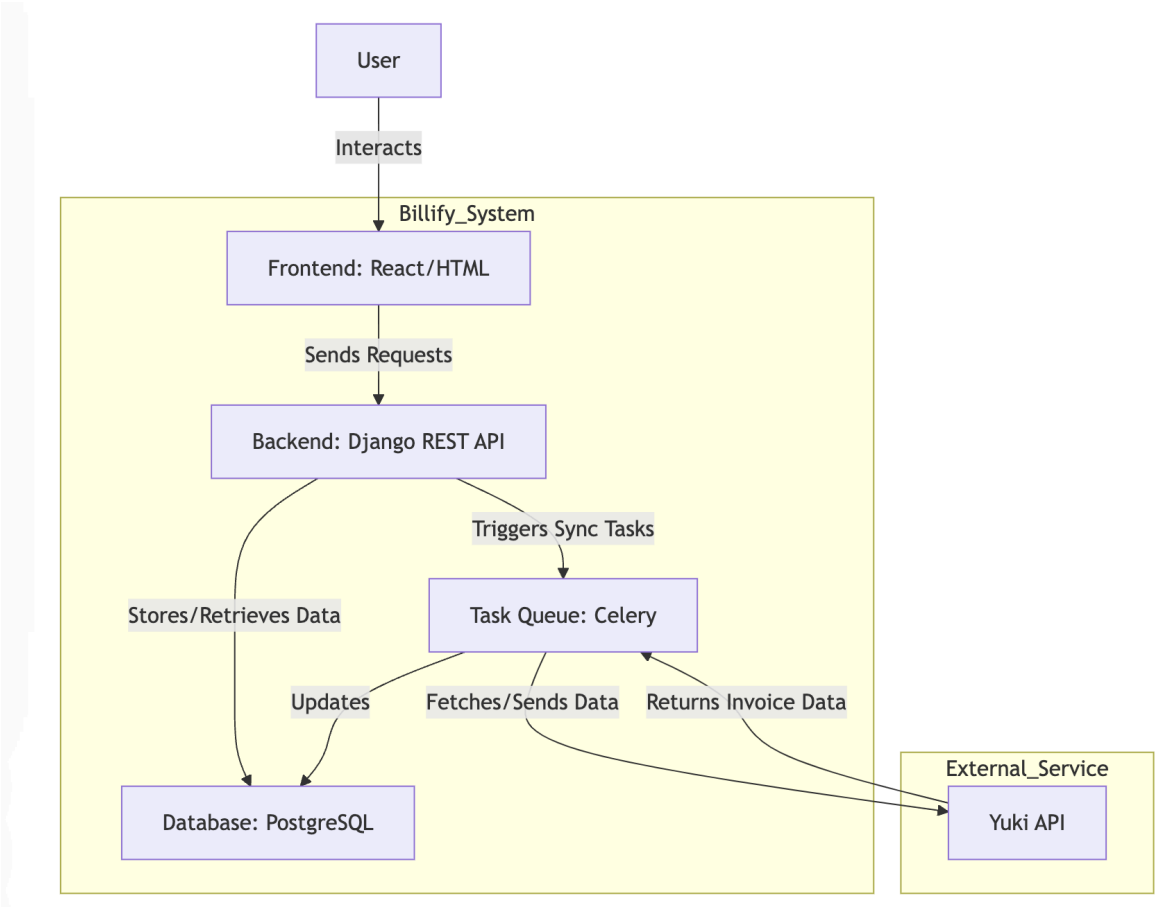
Celery Task Implementation:

```
from celery import shared_task
import requests

def fetch_invoices():
    # Fetch and process invoices from Yuki
    pass

@shared_task
def sync_invoices():
    fetch_invoices()
```

Flow Diagram:



Backend Schema:

```
class Invoice(models.Model):
    INVOICE_TYPES = [
        ('purchase', 'Purchase Invoice'),
        ('sales', 'Sales Invoice'),
    ]

    PAYMENT_STATUSES = [
        ('pending', 'Pending'),
        ('paid', 'Paid'),
        ('overdue', 'Overdue'),
        ('cancelled', 'Cancelled'),
        ('failed', 'Failed'),
    ]

    invoice_number = models.CharField(max_length=50, unique=True)
    type = models.CharField(max_length=20, choices=INVOICE_TYPES)
    issue_date = models.DateField()
    due_date = models.DateField()
    total_amount = models.DecimalField(max_digits=10,
decimal_places=2)
    payment_status = models.CharField(max_length=20,
choices=PAYMENT_STATUSES)
    yuki_invoice_id = models.CharField(max_length=50, unique=True)

    def __str__(self):
        return f"{self.invoice_number} ({self.type})"
```