# API Documentation: Sequifi Webhook Integration

## 1. Webhook Purpose

This document outlines the integration of customer webhooks triggered by Sequifi when specific onboarding events occur for sales representatives.

## 2. Webhook Endpoint

https://test.indian.railway.app/webhooks/sequifi

## 3. Event Types and Payloads

### 3.1 Event: sales\_rep\_signup

Triggered when a sales representative has completed onboarding in Sequifi.

Sample Payload:

{  
 "type": "sales\_rep\_signup",  
 "data": {  
 "id": "1234567",  
 "department\_name": "Sales",  
 "first\_name": "John",  
 "last\_name": "Doe",  
 "office\_name": "New York",  
 "status\_name": "Active",  
 "manager\_name": "Jane Smith",  
 "email": "john.doe3@example.com",  
 "phone": "123-456-7892",  
 "signedAt": "2023-10-01T12:00:00Z",  
 "position\_name": "Sales Representative",  
 "offer\_letter\_accepted": true  
 }  
}

### 3.2 Event: offer\_letter\_signed

Triggered when a sales representative signs their offer letter. Only the id and status\_name fields are required.

Sample Payload:

{  
 "type": "offer\_letter\_signed",  
 "data": {  
 "id": "12345",  
 "status\_name": "Active"  
 }  
}

### 3.3 Event: hiredEmployee

Triggered when a sales representative completes the onboarding process.

Sample Payload:

{  
 "type": "hiredEmployee",  
 "data": {  
 "id": "12345",  
 "status\_name": "Active"  
 }  
}

## 4. Status Push Summary

Data will be pushed to the webhook under the following conditions:

1. Offer letter sent → status: Offer letter sent

2. Offer letter signed by rep → status: Onboarding

3. Rep completed onboarding → status: Active (type: hiredEmployee)

**php-assignment-1.php**  
You need to review the assessment file **php-assignment-1.php**, which contains four functions. Your task is to understand the logic in these functions and implement them as REST APIs.

Data should be passed to the APIs in **JSON format**, as described in the file.

For example, if you're calling the function applyDigiSign, which is used for offer\_letter\_signed, you'll need to pass the following parameters in the API request:

* type
* data
* id
* status\_name

The same approach should be followed for the other three functions — each one should be implemented as a separate REST API using the same logic.

**php-assignment-2.php**

In this assignment, you are required to identify the issues in the code provided in **php-assignment-2.php**. Analyze what's wrong, explain the problems, and provide the necessary corrections to fix the code.  
  
  
===============================================================

**Note:** This task must be completed using the existing code structure, specifically within Laravel functions. You need to integrate the provided API into the php-assignment 1 file.  
There's **no need to create a model**—just implement the required logic directly in php-assignment 1.  
You are **not required to create a new API** that matches the provided URL; instead, you need to **send data to the existing API** as per the documentation.

**The assignment must be completed within 24 hours. Submissions made after this period will not be considered, so please ensure you finish and submit it within the given timeframe.**