**Spot Draft - Junior Integration Assignment Documentation**

**BRINDHA S**

**Table of Contents**

**1.Introduction**

**2.Integration Overview**

2.1. Asana

2.2. Airtable

**3.Requirements Analysis**

3.1. Integration Requirements

3.2. Technical Requirements

**4.Architecture and Design**

4.1. System Architecture

4.2. Data Mapping

**5.Implementation**

5.1. Language and Framework Selection

5.2. Asana Webhooks

5.3. Airtable API Integration

5.4. Code Explanation

**6.Deployment**

6.1. Prerequisites

6.2. Deployment Steps

**7.Testing and Validation**

7.1. Test Cases

7.2. Validation Results

**8.ConclusioN**

1. **Introduction**

This documentation presents the integration between Asana and Airtable, developed as part of the SpotDRAFT - Junior Integration Assignment. The marketing agency faces challenges in maintaining a seamless workflow between project management and data organization. The integration will automatically copy tasks from Asana to Airtable, enabling better data organization and analysis.

**2. Integration Overview**

**2.1. Asana**

Asana is a popular project management tool used by the marketing agency to track tasks, manage projects, and collaborate with team members. It offers a robust API that allows developers to interact with Asana data and events, including creating webhooks to receive real-time notifications about task creation.

**2.2. Airtable**

Airtable is a flexible and powerful database tool used by the marketing agency for storing and analyzing data. It provides an API that enables developers to read and write data to Airtable bases, allowing seamless integration with other services.

**3. Requirements Analysis**

**3.1. Integration Requirements**

The integration must copy new tasks created in Asana to Airtable. The copied tasks should be stored as new rows in an Airtable table named "Asana Tasks" with specific columns, including Task ID, Name, Assignee, Due Date, and Description.

**3.2. Technical Requirements**

**Language/Framework:** The integration service will be developed using a language and framework of the developer's choice.

**Webhooks:** Asana webhooks will be utilized to trigger the integration whenever a new task is created in Asana.

**Airtable API**: The Airtable API will be used to write new records to the "Asana Tasks" table in Airtable.

SYSTEM REQUIREMENTS:

* Node.js
* Vs code
* Npm express
* Npm install
* Ngrok
* Postman

PACKAGES USED:

* Package.json
* Package-lock.json
* Nodejs
* Numpy
* nodemon

**4. Architecture and Design**

**4.1. System Architecture**

The integration service will be built as a standalone application that listens for incoming Asana webhook events. When a new task event is received, the service will fetch the relevant task details from Asana and then create a new record in the "Asana Tasks" table in Airtable using the Airtable API.

**4.2. Data Mapping**

The following data will be mapped from Asana to Airtable:

Asana Field Airtable Column

Task ID Task ID

Name Name

Assignee Assignee

Due Date Due Date

Description Description

**5. Implementation**

**5.1. Language and Framework Selection**

The developer is free to choose the language and framework that they are comfortable with. Popular choices include Python with Flask or Django, Node.js with Express, Ruby on Rails, etc.

**5.2. Asana Webhooks**

To set up Asana webhooks, the developer will need to create a webhook subscription to the desired project or workspace in Asana. Asana will then send a POST request to the integration service whenever a new task is created.

**5.3. Airtable API Integration**

The integration service will interact with the Airtable API using HTTP requests. It will make use of the Airtable API documentation to authenticate the service and write new records to the "Asana Tasks" table.

**5.4. Code Explanation**

The integration code will be thoroughly documented, explaining each function's purpose, input, and output. It will also include comments to enhance code readability.

**6. Deployment**

**6.1. Prerequisites**

The deployment documentation will outline the required dependencies, such as the language runtime, any necessary libraries, and access to Asana and Airtable accounts with appropriate API keys.

**6.2. Deployment Steps**

The deployment steps will guide the user through the process of setting up the integration service on their server or cloud platform, including configuring webhooks in Asana and connecting to Airtable.

**7. Testing and Validation**

**7.1. Test Cases**

The documentation will include a set of test cases covering various scenarios to validate the integration's functionality, including task creation, data mapping, and error handling.

**7.2. Validation Results**

The validation results will be recorded, demonstrating the successful integration of Asana tasks into Airtable and highlighting any potential issues that were resolved during testing.

**8. Future Enhancements**

The documentation will suggest possible future enhancements, such as bidirectional syncing between Asana and Airtable, support for task updates and deletions, or additional data fields.

**9. Conclusion**

The documentation will conclude by summarizing the key points covered in the SpotDRAFT - Junior Integration Assignment. It will reiterate the benefits of the integration between Asana and Airtable and express appreciation for the opportunity to work on this project.