

Configuration Management System for International Onboarding

Problem Statement

An international logistics company, engaged in global transactions, faces challenges in managing the onboarding requirements for organizations across different countries. Each country has distinct regulatory and compliance requirements for business onboarding. For instance, onboarding a business in India necessitates details such as Business Name, PAN, and GSTIN numbers, whereas onboarding a business in the USA might require different registration numbers and additional details.

1. Create a configuration for a country's onboarding requirements.
2. Retrieve the configuration requirements of a country.
3. Update the configuration requirements of a country.
4. Delete the configuration requirements of a country.

Introduction

This report details the development of a robust and scalable Configuration Management System using FastAPI to manage onboarding requirements for organizations from different countries. The system ensures smooth international transactions by addressing the unique onboarding needs of each country. The key functionalities include CRUD operations (Create, Read, Update, Delete) for managing configurations specific to each country. This system is particularly useful for international logistics companies handling varied onboarding requirements across countries.

Database Schema

For the given problem statement, we need a database schema that can efficiently handle the diverse and dynamic onboarding requirements of different countries. The database schema should be able to store the country-specific configuration details, which include the country code, business name, and other dynamic fields that might be required for onboarding.

Here is a detailed database schema for the Configuration Management System:

The table will store the configuration details for each country. Each record will represent the onboarding requirements for a particular business in a specific country.

Table : Config

id: A unique identifier for each configuration. This is the primary key.

country_code: This field stores the ISO country code (e.g., IN for India, US for the USA). It is used to identify the country for which the configuration is applicable.

business_name: The name of the business being onboarded. This, along with the country code, forms a unique combination to ensure that each business's configuration is unique within different country.

data: A JSONB field to store the dynamic onboarding requirements. The use of JSONB allows flexibility to store various fields as required by different countries.

A unique constraint is enforced on the combination of **country_code** and **business_name** to ensure each business within a country has a unique configuration

Installation

1. Clone the Repository:

```
git clone <repository_url>
```

```
cd <project_directory>
```

2. Install Dependencies:

```
pip install -r requirements.txt
```

3. Configure Environment Variables:

Set up the database connection in a **.env** file or directly in your environment.

```
DATABASE_URL=postgresql://<username>:<password>@localhost/<database_name>
```

Running the Application

1. Run the application using below command-

```
uvicorn app.main:app --reload
```

2. Verify Setup:

Open your browser and navigate to <http://127.0.0.1:8000/docs> to access the interactive API documentation provided by FastAPI.


FastAPI Endpoints

The FastAPI application provides the following endpoints:

- **/create_configuration [POST]**

Creates a new configuration for a country's onboarding requirements.

Request Body:



POST /create_configuration Create Configuration

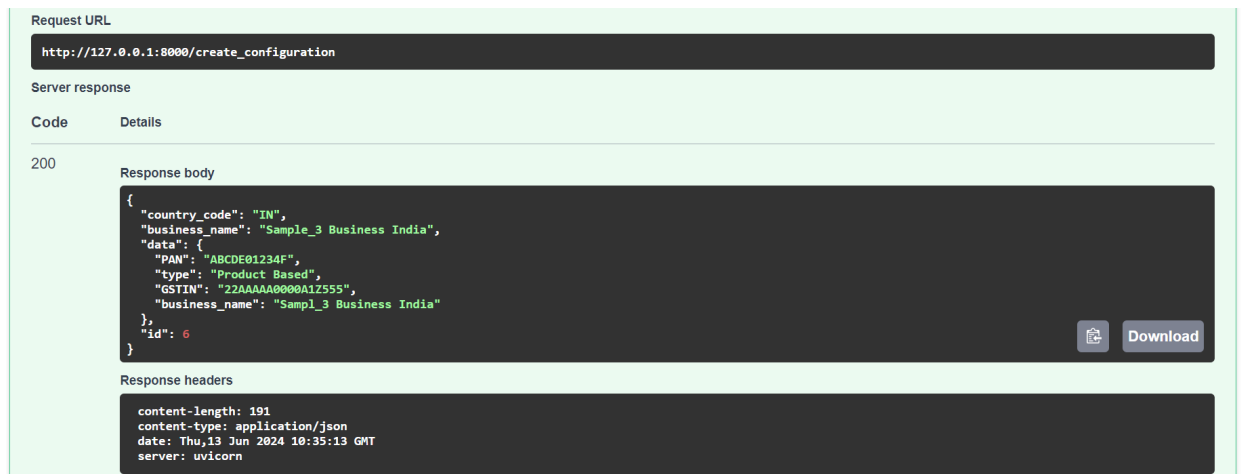
Parameters

No parameters

Request body **required** application/json

```
{
  "country_code": "IN",
  "business_name": "Sample Business India",
  "data": {
    "PAN": "ABCDE1234F",
    "type": "Product Based",
    "GSTIN": "22AAAAA0000A1Z55",
    "business_name": "Sample Business India"
  }
}
```

Response:



Request URL

http://127.0.0.1:8000/create_configuration

Server response

Code Details

200

Response body

```
{
  "country_code": "IN",
  "business_name": "Sample_3 Business India",
  "data": {
    "PAN": "ABCDE01234F",
    "type": "Product Based",
    "GSTIN": "22AAAAA0000A1Z555",
    "business_name": "Samp1_3 Business India"
  },
  "id": 6
}
```

Response headers

```
content-length: 191
content-type: application/json
date: Thu, 13 Jun 2024 10:35:13 GMT
server: uvicorn
```

Error Handling:

When fields are missing or invalid-

- 400: Bad Request if required fields are missing or invalid.

- **/get_configuration/{country_code}/{business_name} [GET]**

Retrieves the configuration requirements for a specific country and business.

Request Body:

The screenshot shows a REST client interface with a blue header bar. The header bar contains a 'GET' button, the endpoint path '/configuration/{country_code}/{business_name}', and the text 'Get Configuration Api'. Below the header bar is a 'Parameters' section with a 'Cancel' button. The parameters are listed in a table with two columns: 'Name' and 'Description'. The first parameter is 'country_code', which is a required string (path) with the value 'IN'. The second parameter is 'business_name', which is a required string (path) with the value 'Sample Business India'. At the bottom of the parameters section are two buttons: 'Execute' and 'Clear'.

Name	Description
country_code * required string (path)	IN
business_name * required string (path)	Sample Business India

Response:

The screenshot shows a REST client interface with a blue header bar. The header bar contains the text 'Request URL'. Below the header bar is a dark bar with the request URL 'http://127.0.0.1:8000/configuration/IN/Sample%20Business%20India'. Below the dark bar is a 'Server response' section. The 'Server response' section has a table with two columns: 'Code' and 'Details'. The 'Code' column has the value '200'. The 'Details' column has the text 'Response body'. Below the 'Response body' text is a dark box containing the JSON response. The JSON response is: { "country_code": "IN", "business_name": "Sample Business India", "data": { "PAN": "ABCDE1234F", "type": "Product Based", "GSTIN": "22AAAAA0000A1Z5", "business_name": "Sample Business India" }, "id": 1 }. To the right of the JSON response is a 'Download' button. Below the JSON response is a 'Response headers' section. The 'Response headers' section has a dark box containing the headers: content-length: 185, content-type: application/json, date: Thu, 13 Jun 2024 10:37:17 GMT, server: uvicorn.

Request URL

http://127.0.0.1:8000/configuration/IN/Sample%20Business%20India

Server response

Code	Details
200	<p>Response body</p> <pre>{ "country_code": "IN", "business_name": "Sample Business India", "data": { "PAN": "ABCDE1234F", "type": "Product Based", "GSTIN": "22AAAAA0000A1Z5", "business_name": "Sample Business India" }, "id": 1 }</pre> <p>Download</p> <p>Response headers</p> <pre>content-length: 185 content-type: application/json date: Thu, 13 Jun 2024 10:37:17 GMT server: uvicorn</pre>

Error Handling:

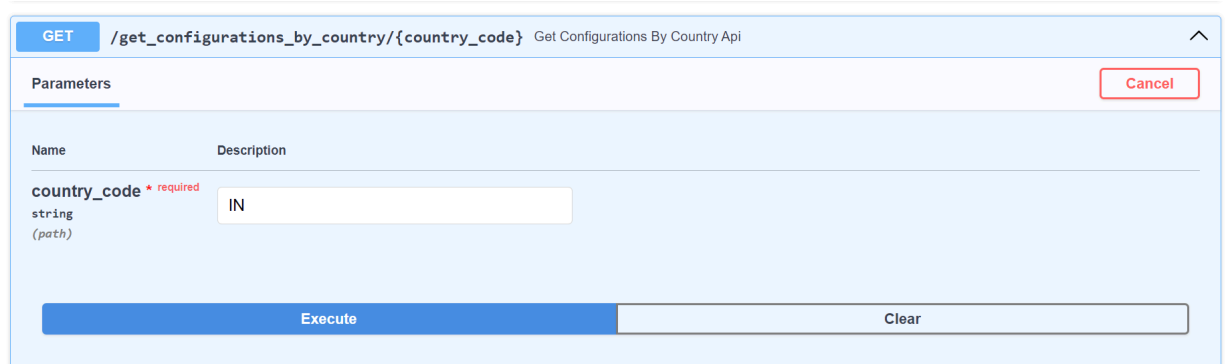
When the country code and business name mentioned are not found.

- 404: Not Found if the configuration does not exist.

- **/get_configurations/{country_code} [GET]**

Retrieves all configurations for a specific country.

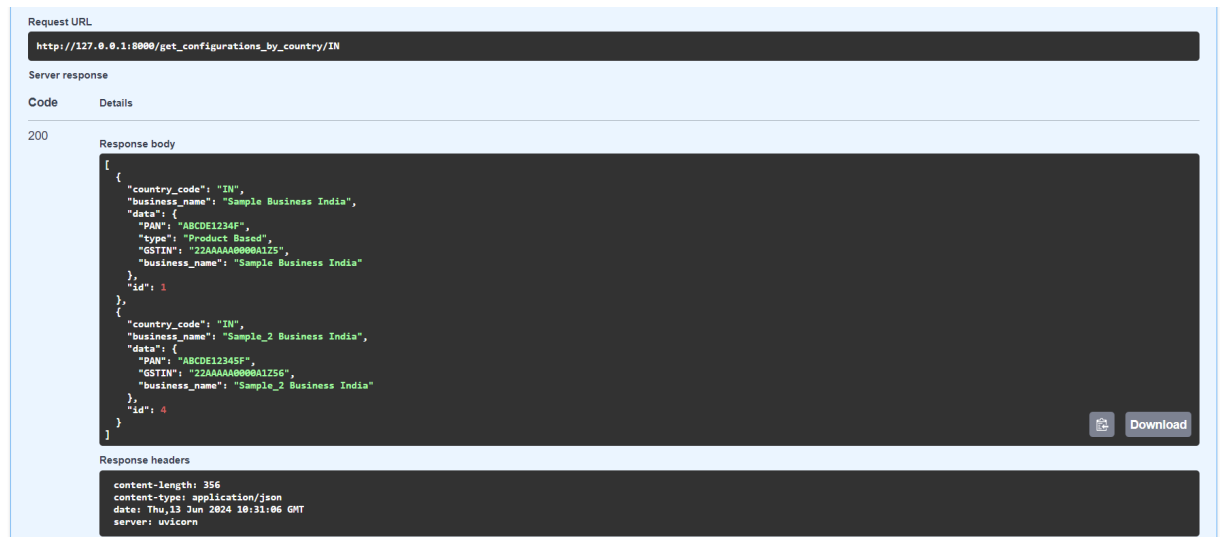
Request Body:



The image shows an API client interface for a GET request to the endpoint `/get_configurations_by_country/{country_code}`. The interface includes a "Parameters" tab with a table for request parameters. The table has two columns: "Name" and "Description". A single parameter is listed: `country_code` (string, path), marked as required. The value `IN` is entered in the adjacent input field. At the bottom, there are "Execute" and "Clear" buttons.

Name	Description
<code>country_code</code> * required string (path)	IN

Response:



The image shows the response view of the API client. The "Request URL" is `http://127.0.0.1:8000/get_configurations_by_country/IN`. The "Server response" section shows a status code of 200. The "Response body" is a JSON array containing two configuration objects. The "Response headers" section shows the following headers: `content-length: 356`, `content-type: application/json`, `date: Thu, 13 Jun 2024 10:31:06 GMT`, and `server: uvicorn`.

```
{
  {
    "country_code": "IN",
    "business_name": "Sample Business India",
    "data": {
      "PAN": "ABCD1234E",
      "type": "Product Based",
      "GSTIN": "22AAAAA0000A1Z5",
      "business_name": "Sample Business India"
    },
    "id": 1
  },
  {
    "country_code": "IN",
    "business_name": "Sample_2 Business India",
    "data": {
      "PAN": "ABCD12345F",
      "GSTIN": "22AAAAA0000A1Z56",
      "business_name": "Sample_2 Business India"
    },
    "id": 2
  }
}
```

Error Handling:

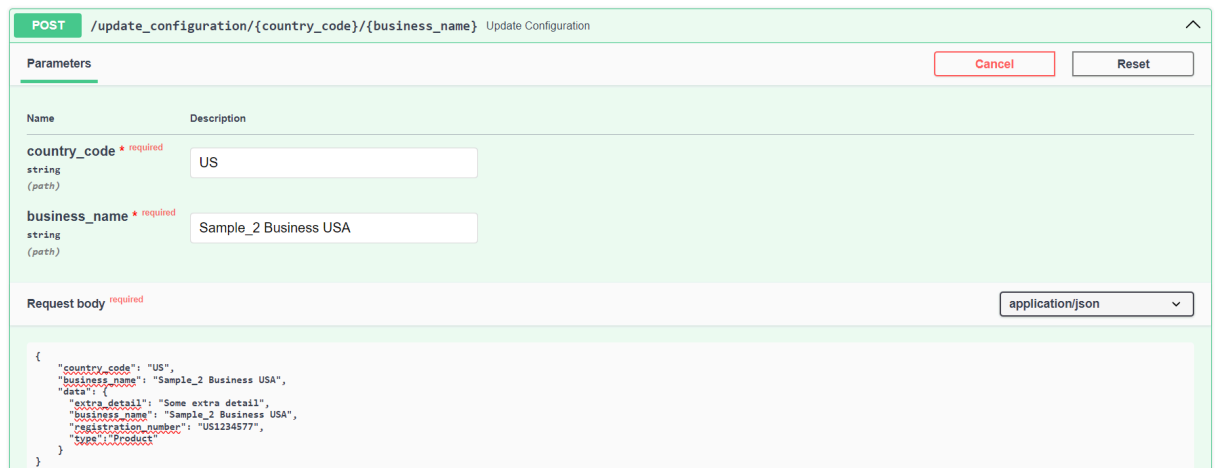
When the country code is not found.

- 404: Not Found if the configuration does not exist.

- **/update_configuration/{country_code}/{business_name} [POST]**

Updates the configuration requirements for a specific country and business.

Request Body:



POST /update_configuration/{country_code}/{business_name} Update Configuration

Parameters

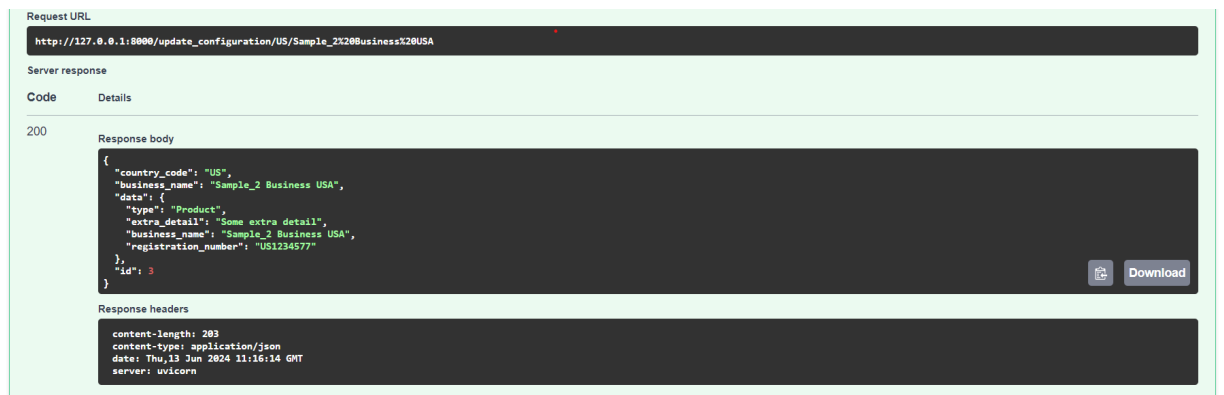
Cancel Reset

Name	Description
country_code * required string (path)	US
business_name * required string (path)	Sample_2 Business USA

Request body required application/json

```
{  "country_code": "US",  "business_name": "Sample_2 Business USA",  "data": {    "extra_detail": "Some extra detail",    "business_name": "Sample_2 Business USA",    "registration_number": "US1234577",    "type": "Product"  }}
```

Response:



Request URL

http://127.0.0.1:8000/update_configuration/US/Sample_2x288Business128USA

Server response

Code	Details
200	<p>Response body</p> <pre>{ "country_code": "US", "business_name": "Sample_2 Business USA", "data": { "type": "Product", "extra_detail": "Some extra detail", "business_name": "Sample_2 Business USA", "registration_number": "US1234577" }, "id": 3}</pre> <p>Response headers</p> <pre>content-length: 283content-type: application/jsondate: Thu, 13 Jun 2024 11:16:14 GMTserver: unicorn</pre>

Download

Error Handling:

When the business that needs to be update is not found.

- 404: Not Found if the configuration does not exist.

- **/delete_configuration/{country_code}/{business_name} [DELETE]**

Deletes the configuration requirements for a specific country and business.

Request Body:

The screenshot shows a REST client interface for the endpoint `/delete_configuration/{country_code}/{business_name}` with the method `DELETE`. The interface includes a 'Parameters' tab with a table of path parameters:

Name	Description
<code>country_code</code> * required string (path)	IN
<code>business_name</code> * required string (path)	Sample Business 2

At the bottom, there are 'Execute' and 'Clear' buttons.

Response:

The screenshot shows the response details for the DELETE endpoint. The status code is 200. The response body is a JSON object:

```
{
  "country_code": "IN",
  "business_name": "Sample Business 2",
  "data": {
    "data": {
      "PAN": "ABKDE1234G",
      "type": "product",
      "GSTIN": "22AZZZ2000A126",
      "business_name": "Sample Business 2"
    }
  },
  "id": 7
}
```

The response headers are:

```
content-length: 179
content-type: application/json
date: Thu, 13 Jun 2024 12:12:54 GMT
server: uvicorn
```

Error Handling:

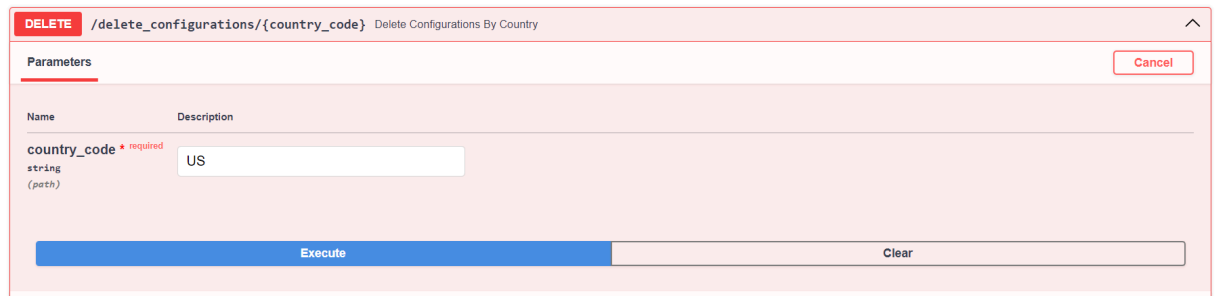
When country code and business name is not found.

- 404: Not Found if the configuration does not exist.

- **/delete_configuration/{country_code} [DELETE]**

Deletes the configuration requirements for a specific country and business.

Request Body:



The image shows an API client interface for a DELETE request. The title bar indicates the method is DELETE and the endpoint is /delete_configurations/{country_code} with the description 'Delete Configurations By Country'. Below the title bar, there is a 'Parameters' tab. Under this tab, there is a table with two columns: 'Name' and 'Description'. The first row shows 'country_code' with a red asterisk and 'required' next to it, indicating it is a required parameter. The type is 'string' and the location is '(path)'. The value 'US' is entered in the input field. At the bottom of the interface, there are two buttons: 'Execute' (blue) and 'Clear' (white).

Response:



The image shows the response section of the API client. The 'Request URL' is http://127.0.0.1:8000/delete_configurations/US. The 'Server response' section shows a status code of 200. The 'Response body' is a JSON object:

```
{  "country_code": "US",  "business_name": "Sample_2 Business USA",  "data": {    "type": "Product",    "extra_detail": "Some extra detail",    "business_name": "Sample_2 Business USA",    "registration_number": "US1234577"  },  "id": 3}
```

. There is a 'Download' button next to the response body. The 'Response headers' section shows:

```
content-length: 285content-type: application/jsondate: Thu, 13 Jun 2024 11:23:21 GMTserver: uvicorn
```

Error Handling:

When the country code is not found.

- 404: Not Found if the configuration does not exist.

Conclusion

The proposed FastAPI application successfully addresses the needs of the international logistics company by providing a robust and scalable system for managing country-specific onboarding configurations. The solution leverages modern technologies such as FastAPI, PostgreSQL, and SQLAlchemy to ensure efficient CRUD operations. Comprehensive error handling, data validation using Pydantic schemas, and a modular code structure contribute to the system's reliability, maintainability, and security.

By implementing this system, the company can streamline its onboarding process for organizations from different countries, ensuring compliance with varying regulatory requirements and improving overall operational efficiency. The flexibility to update all existing configurations for a country in response to changes in onboarding templates further enhances the system's adaptability to evolving business needs.