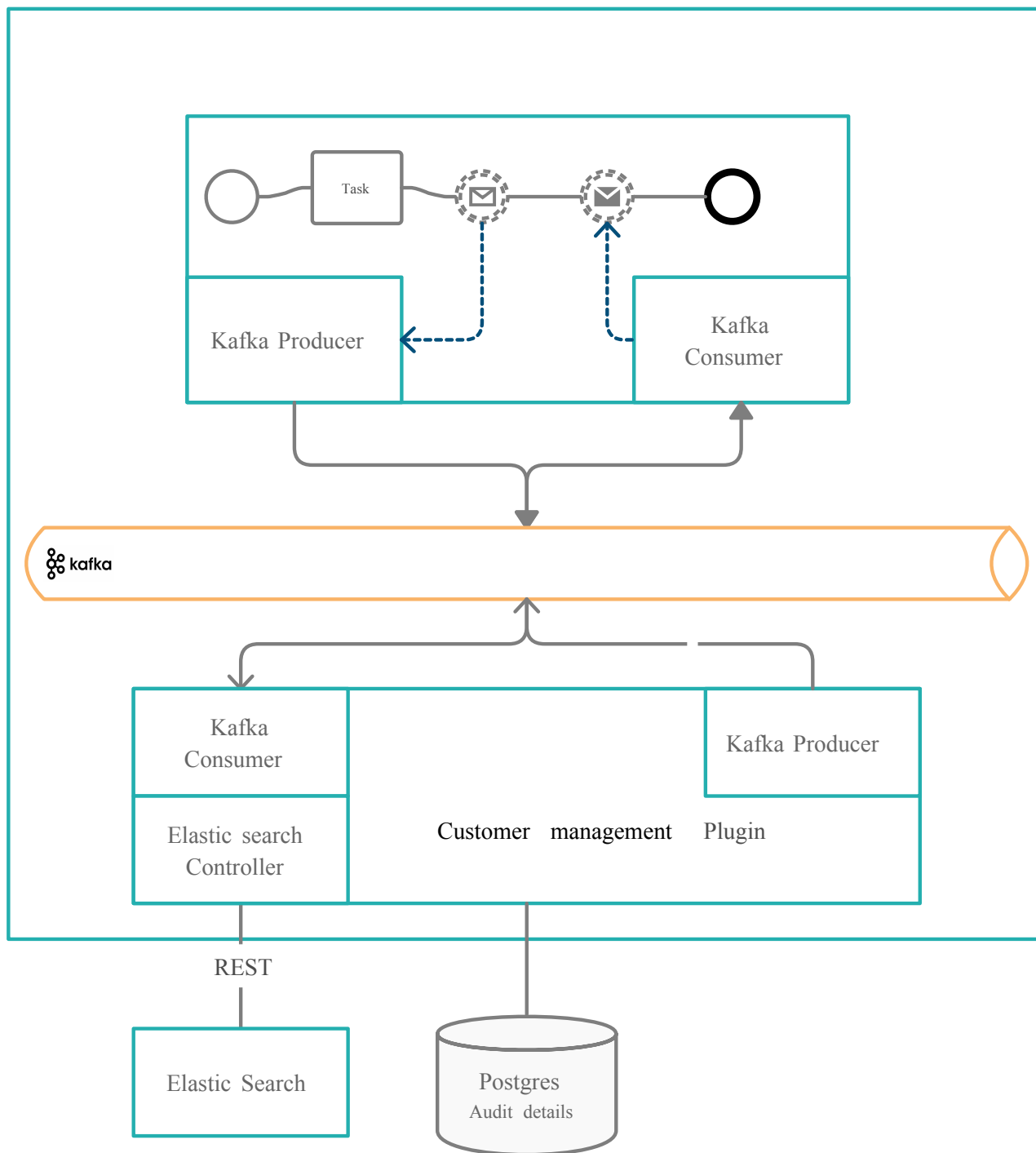# FLOWx.AI

PLATFORM DEEP DIVE / Plugins / 📦 Custom Plugins / 📦 Customer management

Contents

PLATFORM DEEP DIVE / Plugins / 📦 Custom Plugins / 📦 Customer management / Using the customer management plugin

The customer management plugin offers the possibility of retrieving customer details from an elasticSearch engine.

The plugin listens for incoming requests on a Kafka topic and sends the reply to the Engine on an outgoing topic.

> » Kafka topics for Customer Management

Kafka topics for customer management

Customer Search

> ⓘ **INFO**
>
> The kafka topics used for the Customer Management plugin can be
> defined/overwritten using the following environment variables (that can be
> found in the deployment of the service):
>
> - `KAFKA_TOPIC_CUSTOMER_SEARCH_IN` - used to search customers in
>   the customer management plugin
> - `KAFKA_TOPIC_CUSTOMER_SEARCH_OUT` - used to get the response from
>   the customer management plugin to the Engine.

The request sent to the plugin can use any key that was previously configured in
the elasticsearch index where the customers are saved.

Example of an elastic search index:

```
{
  "settings": {
    "analysis": {
      "normalizer": {
        "lowercase_normalizer": {
          "type": "custom",
          "filter": ["lowercase"]
```

```
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "CIF":{
        "type": "keyword",
        "normalizer": "lowercase_normalizer"
      },
      "ClientUniqueIdentifier": {
        "type": "keyword",
        "normalizer": "lowercase_normalizer"
      },
      "CNPFlex":{
        "type": "text"
      },
      "ClientType":{
        "type": "text"
      },
      "ClientCategory":{
        "type": "text"
      },
      "FirstName":{
        "type": "text",
        "fields": {
          "keyword":{
            "type": "keyword",
            "normalizer": "lowercase_normalizer"
          }
        }
      },
      "LastName":{
        "type": "text",
        "fields": {
```

```json
        "keyword":{
          "type": "keyword",
          "normalizer": "lowercase_normalizer"
        }
      }
    },
    "CompanyName":{
      "type": "text",
      "fields": {
        "keyword":{
          "type": "keyword",
          "normalizer": "lowercase_normalizer"
        }
      }
    },
    "DateOfBirth":{
      "type": "date",
      "format": "dd.MM.yyyy"
    },
    "IDDocType":{
      "type": "text"
    },
    "IDSeries":{
      "type": "text"
    },
    "IDNumber":{
      "type": "text"
    },
    "IDIssueDate": {
      "type": "date",
      "format": "dd.MM.yyyy"
    },
    "IDExpiryDate":{
      "type": "date",
      "format": "dd.MM.yyyy"
```

```
      },
      "LegalForm":{
        "type": "text"
      },
      "CreatedDatePJ":{
        "type": "date",
        "format": "dd.MM.yyyy"
      },
      "ClientClosedDate": {
        "type": "date",
        "format": "dd.MM.yyyy"
      },
      "LastModifiedDate":{
        "type": "date",
        "format": "dd.MM.yyyy"
      },
      "ListID":{
        "type": "text"
      },
      "MobilePhone":{
        "type": "text"
      }
    }
  }
}
```

With this index configuration we can search for customers using any key:

Key examples

Example 1 - using only the "ClientUniqueIdentifier" key:

```
{
"ClientUniqueIdentifier": "1900101223344"
}
```

Example 2 - using "FirstName" and "LastName" keys:

```
{
"FirstName": "TestFirstName",
"LastName": "Test Last Name"
}
```

Example 3 - using "FirstName", "DateOfBirth" and "LegalForm" keys:

```
{
"FirstName": "TestFirstName",
"DateOfBirth": "01.01.1990",
"LegalForm": "PF"
}
```

Keys description

- **customers** - list of customers found in the customer management, in the used elasticsearch index, maximum 10 results
- **hasMore** - boolean, true if number of results are bigger than 10, false if the number of results are equal or smaller than 10
- **error** - error description if the request returned an error

Topic name example:

```
ro.flowx.updates.sandbox.customer.management.response
```

Sent body example:

```
"searchResults" : {
    "customers" : [ {
      "id" : "CL12345",
      "firstName" : "John Doe",
      "lastName" : "Doe",
      "birthDate" : "27.02.1982",
      "cui" : "1820227103865_84",
      "companyName" : "",
      "clientCategory" : "PF",
      "clientType" : "PF",
      "idSeries" : "RT",
      "idNumber" : "879948",
      "idDocType" : "CI",
      "idExpiryDate" : "27.02.2023",
      "legalForm" : "",
      "listId" : "4691602",
      "mobilePhone" : "0711111111",
      "attributes" : null,
      "type" : "PF"}],
    "hasMore" : false,
    "error" : null
  }
```

Was this page helpful?

PLATFORM DEEP DIVE / Plugins / 📦 Custom Plugins / 📦 Customer management / Customer management plugin example

Integrate a customer search in a business flow

Follow the next steps to use the user personal number to perform a search query in the customer management plugin.

1. First make sure the details about customers are indexed in the search engine (for example, elasticSearch).

2. Open

   The fallback content to display on prerendering

   web app and create a

   The fallback content to display on prerendering

   .

3. Add a **Kafka send event** node.

4. Configure the **Kafka send event** node by adding the following elements:

- Kafka topic - defined on the `KAFKA_TOPIC_CUSTOMER_SEARCH_IN` environment variable

- Message body (example of identifiers for an indexed customers):

Parameters

[**Custom**] [From integration]

Topics

```
customerSearchTopic-qa
```

Message

```
1   {
2       "id": "${clientIdentification.cif}",
3       "cui": "${clientIdentification.uniqueCode}",
4       "firstName": "${clientIdentification.firstName}",
5       "lastName": "${clientIdentification.lastName}",
6       "birthDate": "${clientIdentification.dateOfBirth}",
7       "companyName": "${clientIdentification.identificationData.name}"
8   }
```

Advanced configuration

Show Headers 🔵

```
1   {"processInstanceId": ${processInstanceId}}
```

> ⓘ **INFO**
>
> For more examples of keys, check **Using the customer management plugin**.

5. Add a **Kafka receive event**.

6. Configure the topic on which you want to receive the response from the CRM, on the value of `KAFKA_TOPIC_CUSTOMER_SEARCH_OUT` environment variable.

Data stream topics

| Custom | From integration |

Topic Name

ai.flowx.updates.qa.customer.management.response

Key Name

receiveReply

🗑

Response example:

```
"searchResults" : {
    "customers" : [ {
      "id" : "ID3456",
      "firstName" : "Jane Doe",
      "lastName" : "Doe",
      "birthDate" : "27.02.1980",
      "cui" : "1820227103840_84",
      "companyName" : "",
      "clientCategory" : "PF_INTL",
      "clientType" : "PF",
      "idSeries" : "RT",
      "idNumber" : "879948",
      "idDocType" : "CI",
      "idExpiryDate" : "27.02.2023",
      "legalForm" : "",
      "listId" : "4691602",
      "mobilePhone" : "0711111111",
```

```
        "attributes" : null,
        "type" : "PF"}],
    "hasMore" : false,
    "error" : null
}
```

Was this page helpful?