

Validation Configuration

Last Modified on 01/29/2025 7:52 am EST

Overview

The invoice-validation.yaml and linelitem-validation.yaml files configures the rules for validation of the extracted header and line-item data and the enrichment data.

The linelitem-validation.yaml validates each line separately, looking for the missing values.

Template

invoice-validation.yaml

```
kind: ruleSet
metadata:
  name: extraction/v1/documents/invoice-validation
spec:
  nodes:
  - if: 'IsNullOrEmpty(it["DocumentType.Value"])'
    then:
      name: DocumentType
      message: Required field
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: 'IsNullOrEmpty(it["InvoiceId.Value"])'
    then:
      name: InvoiceId
      message: Required field
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: 'IsNullOrEmpty(it["InvoiceDate.Value"])'
    then:
      name: InvoiceDate
      message: Required field
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: 'IsNullOrEmpty(it["InvoiceTotal.Value"])'
    then:
      name: InvoiceTotal
      message: Required field
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: '!IsNullOrEmpty(it["DocumentType.Value"]) && !IsMatch(it["DocumentType.Value"], "^[a-zA-Z0-9 ]*$")'
    then:
      name: DocumentType
      message: Only alphanumeric characters allowed
      value: Value
      severity: Severity
      source: Source
```

source: Source

tags: Tags

```
- if: !NullOrEmpty(it["InvoiceId.Value"]) && !Matches(it["InvoiceId.Value"], "^[a-zA-Z0-9 ]*$")
then:
  name: InvoiceId
  message: Only alphanumeric characters allowed
  value: Value
  severity: Severity
  source: Source
  tags: Tags

- if: |
  !string.IsNullOrEmpty(it["TotalTax.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["miscCharges.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["pickupAllowance.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["samplesCharges.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["otherCharges.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["InvoiceTotal.Value"]?.ToString()) &&
  !(it.Where(Key.Contains("InvoiceLineItem.Items") && Key.Contains("].Amount.Value")).Sum(decimal.Parse(Value
.ToString())) +
  it["otherCharges.Value"] +
  it["samplesCharges.Value"] +
  it["pickupAllowance.Value"] +
  it["miscCharges.Value"] +
  it["TotalTax.Value"]
  <= it["InvoiceTotal.Value"] + 1) &&
  (it.Where(Key.Contains("InvoiceLineItem.Items") && Key.Contains("].Amount.Value")).Sum(decimal.Parse(Value.
ToString())) +
  it["otherCharges.Value"] +
  it["samplesCharges.Value"] +
  it["pickupAllowance.Value"] +
  it["miscCharges.Value"] +
  it["TotalTax.Value"]
  >= it["InvoiceTotal.Value"] - 1)
then:
  name: InvoiceTotal
  message: Gross Amount mismatch
  severity: Severity
  source: Source
  tags: Tags
```

Parameter	Description
if then	The condition based on which the rules will be applied.
Name	The name of the field.
Message	The validation message to be displayed below the field in case of validation error.
Severity	The severity of the exception.
Source	The source of the exception. For example, Invoice Number.
Tags	Any tags associated with the exception.

For example, in the above template:

```
- if: 'IsNullOrEmpty(it["InvoiceId.Value"])'
  then:
    name: InvoiceId
    message: Required field
    severity: Critical
    source: InvoiceID
    tags: InvoiceIDisNull
```

If the InvoiceId value is missing then the name, message, severity, source and tags field values will be filled from the other sources of data.

lineitem-validation.yaml

```
kind: ruleSet
metadata:
  name: extraction/v1/documents/lineitem-validation
spec:
  nodes:
  - if: 'string.IsNullOrEmpty(it["Amount.Value"]?.ToString())'
    then:
      name: InvoiceLineItem.Amount
      message: Required
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: 'string.IsNullOrEmpty(it["UnitPrice.Value"]?.ToString())'
    then:
      name: InvoiceLineItem.UnitPrice
      message: Required
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: |
    string.IsNullOrEmpty(it["UnitPrice.Value"]?.ToString()) &&
    string.IsNullOrEmpty(it["Quantity.Value"]?.ToString()) &&
    string.IsNullOrEmpty(it["CatchWeight.Value"]?.ToString())
    then:
      name: InvoiceLineItem.Amount
      message: Amount mismatch
      value: Value
      severity: Severity
      source: Source
      tags: Tags

  - if: |
    !string.IsNullOrEmpty(it["UnitPrice.Value"]?.ToString()) &&
    !string.IsNullOrEmpty(it["Quantity.Value"]?.ToString()) &&
    !string.IsNullOrEmpty(it["CatchWeight.Value"]?.ToString()) &&
    !(decimal.Parse(it["UnitPrice.Value"].ToString()) * decimal.Parse(it["CatchWeight.Value"].ToString()) <= it["Amount.Value"] + 1 && decimal.Parse(it["UnitPrice.Value"].ToString()) * decimal.Parse(it["CatchWeight.Value"].ToString()) >= it["Amount.Value"] - 1)
    then:
      name: InvoiceLineItem.Amount
      message: Amount mismatch
      value: Value
      severity: Severity
      source: Source
      tags: Tags
```

```

- if: |
  !string.IsNullOrEmpty(it["UnitPrice.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["CatchWeight.Value"]?.ToString()) &&
  string.IsNullOrEmpty(it["Quantity.Value"]?.ToString()) && !(decimal.Parse(it["UnitPrice.Value"].ToString()) *
decimal.Parse(it["CatchWeight.Value"].ToString()) <= it["Amount.Value"] + 1 && decimal.Parse(it["UnitPrice.Value"].
ToString()) * decimal.Parse(it["CatchWeight.Value"].ToString()) >= it["Amount.Value"] - 1)
  then:
    name: InvoiceLineItem.Amount
    message: Amount mismatch
    value: Value
    severity: Severity
    source: Source
    tags: Tags

- if: |
  !string.IsNullOrEmpty(it["UnitPrice.Value"]?.ToString()) &&
  !string.IsNullOrEmpty(it["Quantity.Value"]?.ToString()) &&
  string.IsNullOrEmpty(it["CatchWeight.Value"]?.ToString()) &&
  !(decimal.Parse(it["UnitPrice.Value"].ToString()) * decimal.Parse(it["Quantity.Value"].ToString()) <= it["Amount.V
alue"] + 1 && decimal.Parse(it["UnitPrice.Value"].ToString()) * decimal.Parse(it["Quantity.Value"].ToString()) >= it["
Amount.Value"] - 1)
  then:
    name: InvoiceLineItem.Amount
    message: Amount mismatch
    value: Value
    severity: Severity
    source: Source
    tags: Tags

- if: |
  (string.IsNullOrEmpty(it["UnitPrice.Value"]?.ToString()) &&
  string.IsNullOrEmpty(it["Quantity.Value"]?.ToString())) ||
  (string.IsNullOrEmpty(it["CatchWeight.Value"]?.ToString()) && string.IsNullOrEmpty(it["UnitPrice.Value
"]?.ToString()))
  then:
    name: InvoiceLineItem.Amount
    message: Amount mismatch
    value: Value
    severity: Severity
    source: Source
    tags: Tags

```

In the above config parameters, the system validates InvoiceAmount for each line and for the missing value the name, message, severity, source and tags field values will be filled from the other sources of data.