

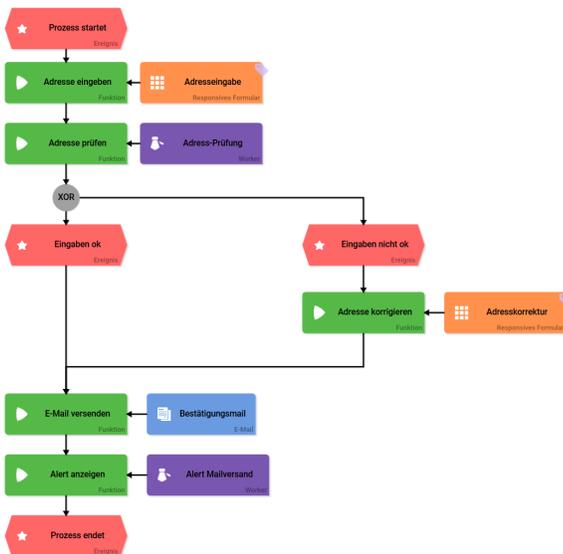
# Komplexe Bedingungspruefung

Bei der Verwendung einer XOR-Verzweigung müssen in den Ereignissen, die auf den Konnektor folgen, Bedingungen abgeprüft werden. Je nach Ergebnis wird ein anderer Strang des Prozesses durchlaufen. Bedingungsprüfungen können so komplex werden, dass sie nicht mehr im Ereignis selbst abgebildet werden können. Bei komplexen Bedingungsprüfungen bietet sich daher der Einsatz eines Workers an.

Gliedern Sie in diesen Fällen die Bedingungsprüfung in einen Worker aus und setzen Sie am Ende der Prüfung eine Variable mit entsprechenden Werten, zum Beispiel **success/failure**. In den Ereignissen können Sie anschließend auf den Inhalt dieser Variable prüfen.

## Anwendungsbeispiel

Im Formular **Adresseingabe** soll ein Anwender seine Adresse eintragen. Anschließend werden die Eingaben geprüft: Sind die Eingaben in Ordnung, wird eine **Bestätigungsmail** an den Anwender versendet. Entsprechen die Eingaben nicht den zugrunde liegenden Bedingungen, wird ein Korrekturversuch unternommen. Die korrigierten Daten werden dem Benutzer im Formular **Adresskorrektur** angezeigt. Der Anwender erhält im Anschluss eine **Bestätigungsmail** mit den korrigierten Daten. Über den Versand der Mail wird der Anwender durch einen abschließenden **Alert** informiert:



### On this Page:

- [Anwendungsbeispiel](#)

### Related Pages:

- [Einen Alert einbauen](#)

### Related Documentation:

- [BPaaS](#)

### Persönliche Daten

Vorname

Nachname

Geburtsdatum

### Anschrift

Straße  Nr.

Adresszusatz

PLZ  Wohnort

Land

Im Formular **Adresseingabe** gibt es separate Formulare für Straße und Hausnummer (Nr). Das Hausnummernfeld wird jedoch gerne übersehen. Daher sollen die Eingaben nach dem Ausfüllen des Formulars geprüft werden.

Im Editor des Workers Adress-Prüfung werden die Prüfungen definiert, die nach der Eingabe vorgenommen werden sollen.

Folgende Operationen sind vorgesehen:

- Es wird geprüft, ob im Feld Nr Inhalt vorhanden ist. Wurde das Feld leer gelassen, wird geprüft, ob am Ende der Eingabe des Feldes Straße Zahlen bzw. Zahlen mit einem Kleinbuchstaben eingegeben wurden (um Hausnummern wie zum Beispiel "14a" nicht auszuschließen). Ist dies der Fall, wird die eingegebene Zahl als angenommene Hausnummer ermittelt.
- Wenn aus dem Feld Straße eine Hausnummer ermittelt werden konnte, soll diese in das Feld Nr geschrieben werden. Gleichzeitig wird die Hausnummer im Feld Straße entfernt.
- Wurde eine Korrektur vorgenommen, erhält die Variable **numberWasGussed** den Eintrag **yes**. Wurde keine Korrektur vorgenommen, enthält die Variable **numberWasGussed** den Eintrag **no**.

```

Eigenschaften des Workers Adress-Prüfung
1 var streetCandidate = container.get('Straße');
2 var numberCandidate = container.get('Nr');
3
4 var numberLists = (numberCandidate);
5 var guessedNumberList = null;
6
7 if (numberExists)
8   guessedNumberList = streetCandidate.match(/^(?!(0-9){1,5})$/);
9
10
11
12 if (guessedNumberList)
13   numberCandidate = guessedNumberList[0];
14   streetCandidate = streetCandidate.replace(guessedNumberList[0], '');
15   container.put('Straße', streetCandidate);
16   container.put('Nr', numberCandidate);
17   container.put('numberWasGussed', 'yes');
18 } else {
19   container.put('numberWasGussed', 'no');
20 }

```

SPEICHERN ABBRECHEN

```

graph TD
    A[Eingaben nicht ok - Ereignis] --> B[Adresse korrigieren - Funktion]
    B --> A

```

**Bearbeiten: Ereignis**

Name: Eingaben nicht ok

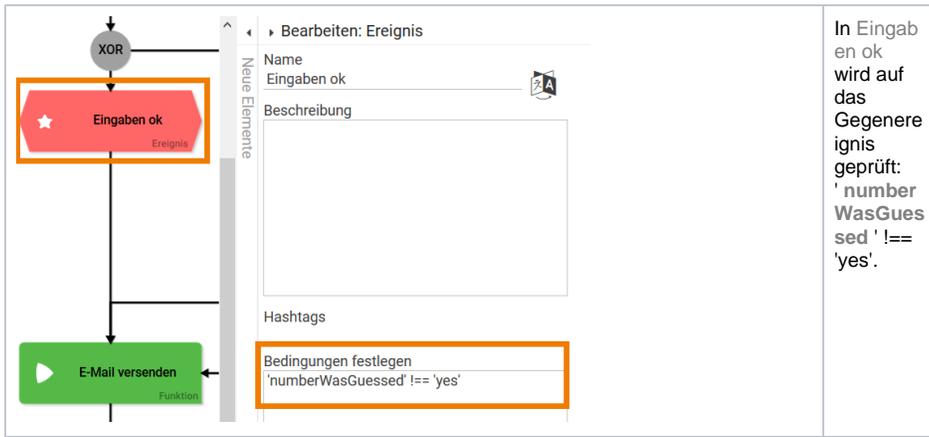
Beschreibung:

Hashtags:

Bedingungen festlegen: 'numberWasGussed' === 'yes'

Die Bedingun gsprüfung in den Ereignissen bezieht sich nun ausschließlich auf die Variable **numberWasGussed**.

Im Ereignis E ingaben nicht ok wird geprüft, ob Korrekturen vorgenommen wurden: ' number WasGussed ' === 'yes'.



In Eingaben ok wird auf das Gegenereignis geprüft: 'numberWasGessed' != 'yes'.