

ConfiForms-Lösung für Jira Epic und Issues Erstellung

Übersicht der Lösung

Diese Lösung erstellt ein ConfiForms-Formular in Confluence, das:

1. Ein Epic in Jira mit den angegebenen Informationen erstellt
2. Die entsprechende Checkliste in das Epic einfügt
3. Optional bis zu vier Issues erstellt, die dem Epic zugeordnet sind
4. In jedes Issue eine unterschiedliche vorhandene Checkliste einfügt
5. Eine Bestätigungstabelle mit den erstellten Tickets anzeigt

Formularstruktur

1. ConfiForms Form Definition

Wir beginnen mit der Erstellung eines ConfiForms-Formulars mit dem Namen "EpicCreationForm".

2. Formularfelder

Folgende Felder werden benötigt:

Epic-Informationen

- **epicName** (Text) - Name des Epics
- **epicDescription** (Textarea) - Beschreibung des Epics
- **priority** (Select) - Priorität mit Optionen: "Sehr Hoch", "Hoch", "Mittel", "Niedrig", "Sehr Niedrig"
- **assignee** (User Picker) - Zugewiesen an
- **dueDate** (Date) - Fällig am
- **issueType** (Select) - Issue-Typ mit Option "Epic" (standardmäßig ausgewählt)
- **jiraProject** (Select) - Jira-Projekt mit Wert "JIRAPROJ24"
- **jiraKey** (Text, optional) - Jira-Key (falls bereits vergeben)

Issue-Checkboxen

- **createIssue1** (Checkbox) - Issue 1 erstellen?
- **createIssue2** (Checkbox) - Issue 2 erstellen?
- **createIssue3** (Checkbox) - Issue 3 erstellen?
- **createIssue4** (Checkbox) - Issue 4 erstellen?

Versteckte Felder für Ergebnisse

- **epicJiraKey** (Text) - Speichert den erstellten Epic-Key
- **issue1JiraKey** (Text) - Speichert den erstellten Issue 1-Key
- **issue2JiraKey** (Text) - Speichert den erstellten Issue 2-Key
- **issue3JiraKey** (Text) - Speichert den erstellten Issue 3-Key
- **issue4JiraKey** (Text) - Speichert den erstellten Issue 4-Key

3. ConfiForms Registration Control

Zur Anzeige des Formulars auf der Seite.

4. IFTTT Integration Rules

Mehrere IFTTT-Regeln werden benötigt:

1. Epic erstellen
2. Issue 1 erstellen (wenn Checkbox aktiviert)
3. Issue 2 erstellen (wenn Checkbox aktiviert)
4. Issue 3 erstellen (wenn Checkbox aktiviert)
5. Issue 4 erstellen (wenn Checkbox aktiviert)
6. Bestätigungstabelle erstellen

Implementierungsdetails

Schritt 1: Formular definieren

Fügen Sie das "ConfiForms Form Definition" Makro ein und geben Sie einen Namen für das Formular an, z.B. "EpicCreationForm".

Schritt 2: Felder definieren

Fügen Sie für jedes Feld ein "ConfiForms Field Definition" Makro ein:

Epic Name

Feld-ID: epicName
Feldtyp: Text
Label: Epic Name
Erforderlich: Ja

Epic Beschreibung

Feld-ID: epicDescription
Feldtyp: Textarea
Label: Epic Beschreibung
Erforderlich: Ja

Priorität

Feld-ID: priority
Feldtyp: Select
Label: Priorität
Optionen: Sehr Hoch | Hoch | Mittel | Niedrig | Sehr Niedrig
Erforderlich: Ja

Zugewiesen an

Feld-ID: assignee
Feldtyp: User Picker
Label: Zugewiesen an
Erforderlich: Ja

Fällig am

Feld-ID: dueDate
Feldtyp: Date
Label: Fällig am
Erforderlich: Ja

Issue-Typ

Feld-ID: issueType
Feldtyp: Select
Label: Issue-Typ
Optionen: Epic

Default: Epic
Erforderlich: Ja

Jira-Projekt

Feld-ID: jiraProject
Feldtyp: Select
Label: Jira-Projekt
Optionen: JIRAPROJ24
Default: JIRAPROJ24
Erforderlich: Ja

Jira-Key

Feld-ID: jiraKey
Feldtyp: Text
Label: Jira-Key (falls bereits vergeben)
Erforderlich: Nein

Issue 1 erstellen

Feld-ID: createIssue1
Feldtyp: Checkbox
Label: Issue 1 erstellen?

Issue 2 erstellen

Feld-ID: createIssue2
Feldtyp: Checkbox
Label: Issue 2 erstellen?

Issue 3 erstellen

Feld-ID: createIssue3
Feldtyp: Checkbox
Label: Issue 3 erstellen?

Issue 4 erstellen

Feld-ID: createIssue4
Feldtyp: Checkbox
Label: Issue 4 erstellen?

Versteckte Felder für Ergebnisse

Feld-ID: epicJiraKey
Feldtyp: Text
Label: Erstellter Epic-Key

Feld-ID: issue1JiraKey
Feldtyp: Text
Label: Erstellter Issue 1-Key

Feld-ID: issue2JiraKey
Feldtyp: Text
Label: Erstellter Issue 2-Key

Feld-ID: issue3JiraKey
Feldtyp: Text
Label: Erstellter Issue 3-Key

Feld-ID: issue4JiraKey
Feldtyp: Text
Label: Erstellter Issue 4-Key

Schritt 3: Formular anzeigen

Fügen Sie das "ConfiForms Registration Control" Makro ein, um das Formular anzuzeigen.

Schritt 4: IFTTT-Regel für Epic-Erstellung

Fügen Sie das "ConfiForms IFTTT Integration Rules" Makro ein:

Event: onCreated
Action: Create JIRA Issue

Im Makro-Body (mit No Format Makro):

```
{  
  "fields": {  
    "project": {  
      "key": "[entry.jiraProject]"  
    },  
    "summary": "[entry.epicName]",  
    "description": "[entry.epicDescription]\n\n!* Muss zuerst abgearbeitet  
werden\n[* Annahmen und Entscheidungen dokumentiert\n[* Risiken und  
Abhangigkeiten ermittelt\n[* Keine kritischen offenen Fragen/Punkte\n[*  
Einfluss nicht-funktionaler Anforderungen bewertet\n[* Testbarkeit bestigt\n[*  
Grobschatzung liegt vor",  
    "issuetype": {  
      "name": "Epic"  
    },  
    "priority": {  
      "name": "[entry.priority]"  
    },  
    "assignee": {  
      "name": "[entry.assignee]"  
    },  
    "duedate": "[entry.dueDate]"  
  }  
}
```

Schritt 5: IFTTT-Regel zum Speichern des Epic-Keys

Fugen Sie ein weiteres "ConfiForms IFTTT Integration Rules" Makro ein:

```
Event: onCreated  
Action: Update Entry
```

Im Makro-Body:

```
epicJiraKey=[iftttResult_0]
```

Schritt 6: IFTTT-Regel fur Issue 1

Fugen Sie ein "ConfiForms IFTTT Integration Rules" Makro mit Bedingung ein:

```
Event: onCreated  
Action: Create JIRA Issue  
Condition: [entry.createIssue1] == "true"
```

Im Makro-Body (mit No Format Makro):

```
{  
  "fields": {  
    "project": {  
      "key": "[entry.jiraProject]"  
    },  
    "summary": "Issue 1 für [entry.epicName]",  
    "description": "Unteraufgabe 1 für Epic [entry.epicJiraKey]\n\n#! nach  
Ticketerstellung, vor weiterer Beauftragung; durch Produktverantwortung\n[*]  
Patch im DevNet abgenommen\n[*] Patchname als Lösungsversion in allen  
zugehörigen Tickets eintragen; wenn nicht auswählbar dann ggf. über Stephan  
Vogt (Vertretung Michael Malburg) Lösungsversion anlegen lassen. Im Anschluss in  
den Tickets nachpflegen.\n[*] Confluence-Seite der Auslieferung anlegen und  
befüllen\n[*] neue Zeile in der Installationsübersicht mit Grunddaten  
befüllen\n[*] Installationsvoraussetzungen: Gibt es Abhängigkeiten /  
Voraussetzungen die der Installation dieses Produkts entgegenstehen? Bspw.  
abhängige Versionen die installiert sein müssen / offene Abnahmen von  
vorangegangenen Produkten auf dieser oder vorgelagerter Umgebung ... Dann  
Tickets entsprechend verknüpfen: bspw. \"wird blockiert durch\"\\n#! vor  
Beauftragung mündliche Absprache zur Installation; durch  
Produktverantwortung\n[*] Datenbankadministrator\n[*]  
Applikationsadministrator\n[*] Testkoordination\n#! Ticketabarbeitung nach  
erfolgter Absprache; primär durch Produktverantwortung\n[*] SG Mobile Apps  
über geplante Änderungen in Kenntnis setzen\n[*] Fälligkeitsdatum der Aufgabe /  
des Tickets auf das vereinbarte Installations-Datum setzen\n[*] Aufgabe / Ticket in den Status \"geplant\" setzen\n[*] Aufgabe / Ticket dem abgesprochenen Admin  
zuweisen\n#! Ticketabarbeitung; durch Administratoren\n[*] Zabbix-  
Wartungsfenster vor Beenden der Applikation eintragen\n[*] Schnittstellen bei  
Windows schließen lassen\n[*] Aufgabe / Ticket durch Admin ggf. in den Status  
\"in Arbeit\" setzen\n[*] durchgeführte Arbeiten im Ticket kommentieren, ggf. an  
folgende Administratoren zuweisen\n[*] Schnittstellen bei Windows öffnen  
lassen\n[*] nach Abschluss der Installationen, Aufgabe / Ticket in den Status \"in  
Review\" setzen und der beauftragenden Produktverantwortung zuweisen\n#!  
nach Installation; durch Produktverantwortung\n[*] Testkoordination über  
Abschluss der Arbeiten und Verfügbarkeit der Umgebung informieren\n[*]  
Installationsübersicht mit Installationsdatum ergänzen\n[*] Produkt  
abnehmen\n[*] Prüfung ob alle Beauftragungen / Unteraufgaben umgesetzt sind,  
ggf. nachhaken\n[*] nach erfolgreicher Abnahme Aufgabe / Ticket in den Status  
\"abgeschlossen\" / \"fertig\" setzen",  
    "issuetype": {  
      "name": "Task"  
    },  
    "priority": {  
      "name": "[entry.priority]"  
    },  
    "assignee": {  
      "name": "[entry.assignee]"  
    },  
    "epiclink": "[entry.epicJiraKey]"  
  }  
}
```

```
}
```

Schritt 7: IFTTT-Regel zum Speichern des Issue 1-Keys

Fügen Sie ein weiteres "ConfiForms IFTTT Integration Rules" Makro ein:

```
Event: onCreated  
Action: Update Entry  
Condition: [entry.createIssue1] == "true"
```

Im Makro-Body:

```
issue1JiraKey=[iftttResult_2]
```

Schritt 8-13: Wiederholen für Issues 2-4

Wiederholen Sie die Schritte 6-7 für die Issues 2, 3 und 4, mit entsprechend angepassten Bedingungen und Feldwerten.

Für Issue 2 (Schulung):

```
{
  "fields": {
    "project": {
      "key": "[entry.jiraProject]"
    },
    "summary": "Issue 2 für [entry.epicName]",
    "description": "Unteraufgabe 2 für Epic [entry.epicJiraKey]\n\n#! vor  
Beauftragung mündliche Absprache zur Installation; durch  
Produktverantwortung\n[*] Installationsvoraussetzungen: Gibt es  
Abhängigkeiten / Voraussetzungen die der Installation dieses Produkts  
entgegenstehen? Bspw. abhängige Versionen die installiert sein müssen / offene  
Abnahmen von vorangegangenen Produkten auf dieser oder vorgelagerter  
Umgebung ... Dann Tickets entsprechend verknüpfen: bspw. \"wird blockiert  
durch\"[*] Datenbankadministrator verfügbar? Im Feld \"Beschreibung\" des  
Sammeltickets oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung  
durchführen wird\n[*] Applikationsadministrator verfügbar? Im Feld  
\"Beschreibung\" des Sammeltickets oder der einzelnen Aufgabe erfassen wer  
geplant die Umsetzung durchführen wird\n[*] Ansprechpartner Schnittstellen  
schließen / öffnen, meist Windows, verfügbar? Im Feld \"Beschreibung\" des  
Sammeltickets oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung  
durchführen wird\n[*] Absprache / Information des Termins mit der Hochschule  
und Fortbildungseinrichtung der Polizei\n[*] Changemanagement\n#! Change; im  
Regelfall durch Produktverantwortung\n[*] Changeantrag erstellen und  
beantragen\n#! Ticketabarbeitung nach Change-Genehmigung; primär durch
```

Produktverantwortung\n[]* Change-Nummer/-Datum/-Uhrzeit/-Dauer im Feld
 \"Beschreibung\" des Sammeltickets; ist kein Sammelticket vorhanden die Angaben
 in der Aufgabe erfassen. Ggf. die Change-Beteiligten weitergehend
 informieren\n[]* Fälligkeitsdatum der Aufgabe / des Tickets auf das vereinbarte
 Change-Datum setzen\n[]* Aufgabe / Ticket in den Status \"geplant\" setzen\n[]*
 Aufgabe / Ticket dem abgesprochenen Admin / Ansprechpartner zuweisen\n[]*
 Anwenderinformation / Ausfallankündigung in POLAS-Umgebung für Ausfall
 einstellen\n[]* SG Mobile Apps über geplante Änderungen in Kenntnis setzen\n#!
 Change Durchführung; durch Administration\n[]* Aufgabe / Ticket durch Admin
 ggf. in den Status \"in Arbeit\" setzen\n[]* durchgeführte Arbeiten im Ticket
 kommentieren, ggf. an folgende Administratoren zuweisen\n[]* nach Abschluss
 der Installationen, Aufgabe / Ticket in den Status \"in Review\" setzen und der
 beauftragenden Produktverantwortung zuweisen\n[]* USU-Aktivität
 abschließen\n#! nach Installation; durch Produktverantwortung\n[]*
 Installationsübersicht mit Installationsdatum ergänzen\n[]* Produkt
 abnehmen\n[]* Prüfung ob alle Beauftragungen / Unteraufgaben umgesetzt sind,
 ggf. nachhaken\n[]* nach erfolgreicher Abnahme Aufgabe / Ticket in den Status
 \"abgeschlossen\" / \"fertig\" setzen",

```

  "issuetype": {
    "name": "Task"
  },
  "priority": {
    "name": "[entry.priority]"
  },
  "assignee": {
    "name": "[entry.assignee]"
  },
  "epiclink": "[entry.epicJiraKey]"
}
}
  
```

Für Issue 3 (QS):

```
{
  "fields": {
    "project": {
      "key": "[entry.jiraProject]"
    },
    "summary": "Issue 3 für [entry.epicName]",
    "description": "Unteraufgabe 3 für Epic [entry.epicJiraKey]\n\n#! vor  

Beauftragung mündliche Absprache zur Installation; durch  

Produktverantwortung\n[ ]* Installationsvoraussetzungen: Gibt es  

Abhängigkeiten / Voraussetzungen die der Installation dieses Produkts  

entgegenstehen? Bspw. abhängige Versionen die installiert sein müssen / offene  

Abnahmen von vorangegangenen Produkten auf dieser oder vorgelagerter  

Umgebung ... Dann Tickets entsprechend verknüpfen: bspw. \"wird blockiert  

durch\"\n[ ]* gibt des von anderen Produktverantwortlichen auf der Umgebung  

noch Produkte die im gleichen Zug mit installiert werden können / sollen? Wenn ja,  

dann prüfen es bereits ein Sammelticket gibt und mit diesem verknüpfen ODER  

andernfalls ein EPIC als Sammelticket erstellen und die gemeinsam"
}
```

umzusetzenden Tickets mit dem Sammelticket verknüpfen.\n[]* Datenbankadministrator verfügbar? Im Feld \"Beschreibung\" des Sammeltickets oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung durchführen wird\n[]* Applikationsadministrator verfügbar? Im Feld \"Beschreibung\" des Sammeltickets oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung durchführen wird\n[]* Ansprechpartner Schnittstellen schließen / öffnen, meist Windows, verfügbar? Im Feld \"Beschreibung\" des Sammeltickets oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung durchführen wird\n[]* Changemanagement\n#! Change; im Regelfall durch Produktverantwortung\n[]* Changeantrag erstellen und Change beantragen\n#! Ticketabarbeitung nach Change-Genehmigung; primär durch Produktverantwortung\n[]* Change-Nummer/-Datum/-Uhrzeit/-Dauer im Feld \"Beschreibung\" des Sammeltickets; ist kein Sammelticket vorhanden die Angaben in der Aufgabe erfassen. Ggf. die Change-Beteiligten weitergehend informieren\n[]* Fälligkeitsdatum der Aufgabe / des Tickets auf das vereinbarte Change-Datum setzen\n[]* Aufgabe / Ticket in den Status \"geplant\" setzen\n[]* Aufgabe / Ticket dem abgesprochenen Admin / Ansprechpartner zuweisen\n[]* Anwenderinformation / Ausfallankündigung in POLAS-Umgebung für Ausfall einstellen\n[]* ggf. Dokumentation in P-Online anpassen und Anwenderinformation zu Änderungen in P-Online einstellen\n[]* SG Mobile Apps über geplante Änderungen in Kenntnis setzen\n#! Change Durchführung; durch Administration\n[]* Aufgabe / Ticket durch Admin ggf. in den Status \"in Arbeit\" setzen\n[]* durchgeführte Arbeiten im Ticket kommentieren, ggf. an folgende Administratoren zuweisen\n[]* nach Abschluss der Installationen, Aufgabe / Ticket in den Status \"in Review\" setzen und der beauftragenden Produktverantwortung zuweisen\n[]* USU-Aktivität abschließen\n#! nach Installation; durch Produktverantwortung\n[]* Windows Schnittstellen ggf. öffnen lassen\n[]* Installationsübersicht mit Installationsdatum ergänzen\n[]* ITK - KM Software-Versionen ergänzen:

```

"issuetype": {
    "name": "Task"
},
"priority": {
    "name": "[entry.priority]"
},
"assignee": {
    "name": "[entry.assignee]"
},
"epiclink": "[entry.epicJiraKey]"
}
}

```

Für Issue 4 (Prod):

```

{
"fields": {
    "project": {
        "key": "[entry.jiraProject]"
    },
    "summary": "Issue 4 für [entry.epicName]",
    "description": "Unteraufgabe 4 für Epic [entry.epicJiraKey]\n\n#! vor  
Beauftragung mündliche Absprache zur Installation; durch  
Produktverantwortung\n[* Installationsvoraussetzungen: Gibt es  
Abhängigkeiten / Voraussetzungen die der Installation dieses Produkts  
entgegenstehen? Bspw. abhängige Versionen die installiert sein müssen / offene  
Abnahmen von vorangegangenen Produkten auf dieser oder vorgelagerter  
Umgebung ... Dann Tickets entsprechend verknüpfen: bspw. \"wird blockiert  
durch\"[* gibt des von anderen Produktverantwortlichen auf der Umgebung  
noch Produkte die im gleichen Zug mit installiert werden können / sollen? Wenn ja,  
dann prüfen es bereits ein Sammelticket gibt und mit diesem verknüpfen ODER  
andernfalls ein EPIC als Sammelticket erstellen und die gemeinsam  
umzusetzenden Tickets mit dem Sammelticket verknüpfen.\n[*  
Datenbankadministrator verfügbar? Im Feld \"Beschreibung\" des Sammeltickets  
oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung durchführen  
wird\n[* Applikationsadministrator verfügbar? Im Feld \"Beschreibung\" des  
Sammeltickets oder der einzelnen Aufgabe erfassen wer geplant die Umsetzung  
durchführen wird\n[* Ansprechpartner Schnittstellen schließen / öffnen, meist  
Windows, verfügbar? Im Feld \"Beschreibung\" des Sammeltickets oder der  
einzelnen Aufgabe erfassen wer geplant die Umsetzung durchführen wird\n[*  
Changemanagement\n#! Change; im Regelfall durch Produktverantwortung\n[*  
Changeantrag erstellen und Change beantragen\n#! Ticketabarbeitung nach  
Change-Genehmigung; primär durch Produktverantwortung\n[* Change-  
Nummer/-Datum/-Uhrzeit/-Dauer im Feld \"Beschreibung\" des Sammeltickets; ist  
kein Sammelticket vorhanden die Angaben in der Aufgabe erfassen. Ggf. die  
Change-Beteiligten weitergehend informieren\n[* Fälligkeitsdatum der Aufgabe /  
des Tickets auf das vereinbarte Change-Datum setzen\n[* Aufgabe / Ticket in den  
Status \"geplant\" setzen\n[* Aufgabe / Ticket dem abgesprochenen Admin /  
Ansprechpartner zuweisen\n[* Anwenderinformation / Ausfallankündigung in  
POLAS-Umgebung für Ausfall einstellen\n[* ggf. Dokumentation in P-Online
}

```

anpassen und Anwenderinformation zu Änderungen in P-Online einstellen\n[* SG Mobile Apps über geplante Änderungen in Kenntnis setzen\n#! Change Durchführung; durch Administration\n[* Aufgabe / Ticket durch Admin ggf. in den Status \"in Arbeit\" setzen\n[* durchgeführte Arbeiten im Ticket kommentieren, ggf. an folgende Administratoren zuweisen\n[* nach Abschluss der Installationen, Aufgabe / Ticket in den Status \"in Review\" setzen und der beauftragenden Produktverantwortung zuweisen\n[* USU-Aktivität abschließen\n#! nach Installation; durch Produktverantwortung\n[* Windows Schnittstellen ggf. öffnen lassen\n[* Installationsübersicht mit Installationsdatum ergänzen\n[* ITK - KM Software-Versionen ergänzen: <https://confluence.itk.extrapol.de/confluence/pages/viewpage.action?spaceKey=LB&title=Software+Versionen>\n[* Produkt abnehmen\n[* Prüfung ob alle Beauftragungen / Unteraufgaben umgesetzt sind, ggf. nachhaken\n[* nach erfolgreicher Abnahme Aufgabe / Ticket in den Status \"abgeschlossen\" / \"fertig\" setzen",

```

"issuetype": {
  "name": "Task"
},
"priority": {
  "name": "[entry.priority]"
},
"assignee": {
  "name": "[entry.assignee]"
},
"epiclink": "[entry.epicJiraKey]"
}
}
}
```

Schritt 14: IFTTT-Regel für Bestätigungstabelle

Fügen Sie ein letztes "ConfiForms IFTTT Integration Rules" Makro ein:

```

Event: onCreated
Action: Show Message
```

Im Makro-Body:

```

<h2>Erstellte Jira-Tickets</h2>
<table class="confluenceTable">
<tr>
  <th>Typ</th>
  <th>Jira-Key</th>
  <th>Name</th>
</tr>
<tr>
  <td>Epic</td>
  <td><a href="https://jira.example.com/browse/[entry.epicJiraKey]">[entry.epicJiraKey]</a></td>
  <td>[entry.epicName]</td>
```

```

</tr>
#if([entry.issue1JiraKey] != "")
|  |  |  |
| --- | --- | --- |
| Issue 1 | \[entry.issue1JiraKey\] | Issue 1 für [entry.epicName] |

#end
#if([entry.issue2JiraKey] != "")
| Issue 2 | \[entry.issue2JiraKey\] | Issue 2 für [entry.epicName] |

#end
#if([entry.issue3JiraKey] != "")
| Issue 3 | \[entry.issue3JiraKey\] | Issue 3 für [entry.epicName] |

#end
#if([entry.issue4JiraKey] != "")
| Issue 4 | \[entry.issue4JiraKey\] | Issue 4 für [entry.epicName] |

#end
</table>

```

Wichtige Hinweise zur Implementierung

1. Stellen Sie sicher, dass die Jira-Verbindung korrekt konfiguriert ist und unter dem Namen "JIRA" in ConfiForms verfügbar ist.
2. Das Feld "epiclink" muss dem tatsächlichen Feldnamen in Ihrem Jira-System entsprechen.
3. Die Checklisten-Texte wurden direkt in die Beschreibungsfelder integriert, wie in den bereitgestellten Dateien angegeben.
4. Die Bestätigungstabelle zeigt nur die tatsächlich erstellten Tickets an, basierend auf den ausgewählten Checkboxen.
5. Die URL "https://jira.example.com/browse/" sollte durch die tatsächliche URL Ihres Jira-Systems ersetzt werden.

Fehlerbehebung

Wenn Probleme auftreten:

1. Überprüfen Sie, ob alle Feldnamen korrekt sind und den Anforderungen von Jira entsprechen.
2. Stellen Sie sicher, dass die Jira-Verbindung funktioniert und die richtigen Berechtigungen hat.
3. Überprüfen Sie die Syntax der IFTTT-Regeln, insbesondere die JSON-Struktur.
4. Prüfen Sie, ob die Bedingungen für die Issue-Erstellung korrekt formuliert sind.
5. Stellen Sie sicher, dass die Epic-Link-Verknüpfung korrekt funktioniert.