



FEEL Beyond DMN

Philipp Ossler
11th May 2023

**CAMUNDA
COMMUNITY
SUMMIT 2023**

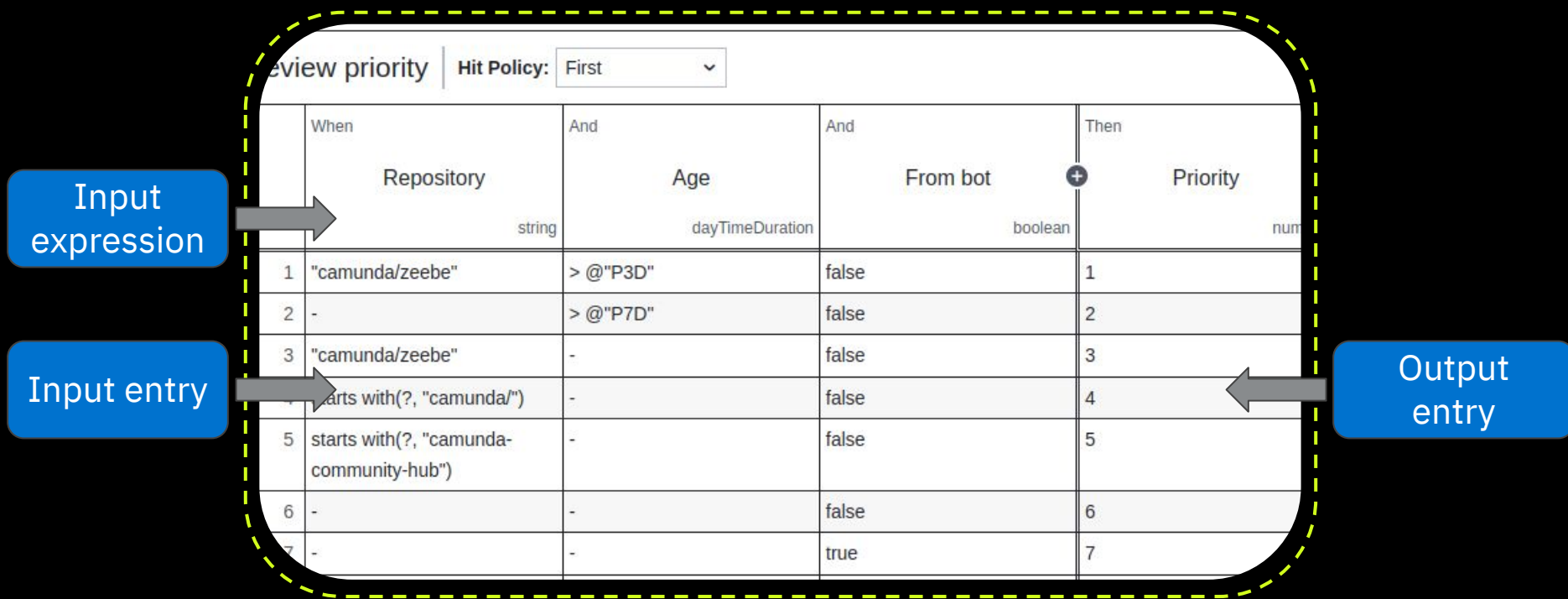


What is FEEL?



Friendly
Enough
Expression
Language

FEEL for DMN



FEEL for BPMN?



Agenda



Basics



FEEL for BPMN



Tips & Tricks

About myself



Philipp Ossler

Friendly Software Engineer

Camunda

- 8 years at Camunda
- Creator of Camunda's FEEL and DMN engine
- Maintainer of some community projects:
 - Zeebe Simple Monitor/Tasklist
 - ZeeQS (GraphQL API)
 - EZE (Embedded Zeebe Engine)
 - Zeebe-Play

Basics

Expression vs. script

Expression
Language



Script
Language

```
sublist(  
  response.body.items[  
    not(draft)],  
  1,  
  3)
```

```
items = response.body.items  
filtered = items[not(draft)]  
sublist(filtered, 1, 3)
```


Data types

FEEL

=

JSON

```
{  
  "boolean": true,  
  "number": 5.0,  
  "string": "foo",  
  "list": [1, 2, 3],  
  "context": {  
    "nested": "yes"  
  },  
  "nullable": null  
}
```

```
{  
  "boolean": true,  
  "number": 5.0,  
  "string": "foo",  
  "list": [1, 2, 3],  
  "context": {  
    "nested": "yes"  
  },  
  "nullable": null  
}
```

Temporal types

```
date("2023-05-11")
```

```
@"2023-05-11"
```

```
time("15:10:00")
```

```
@"15:10:00"
```

```
@"15:10:00+02:00"
```

```
@"15:10:00@Europe/Berlin"
```

```
date and time("2023-05-11T15:10:00")
```

```
@"2023-05-11T15:10:00"
```

```
@"2023-05-11T15:10:00+02:00"
```

```
@"2023-05-11T15:10:00@Europe/Berlin"
```

```
duration("PT45M")
```

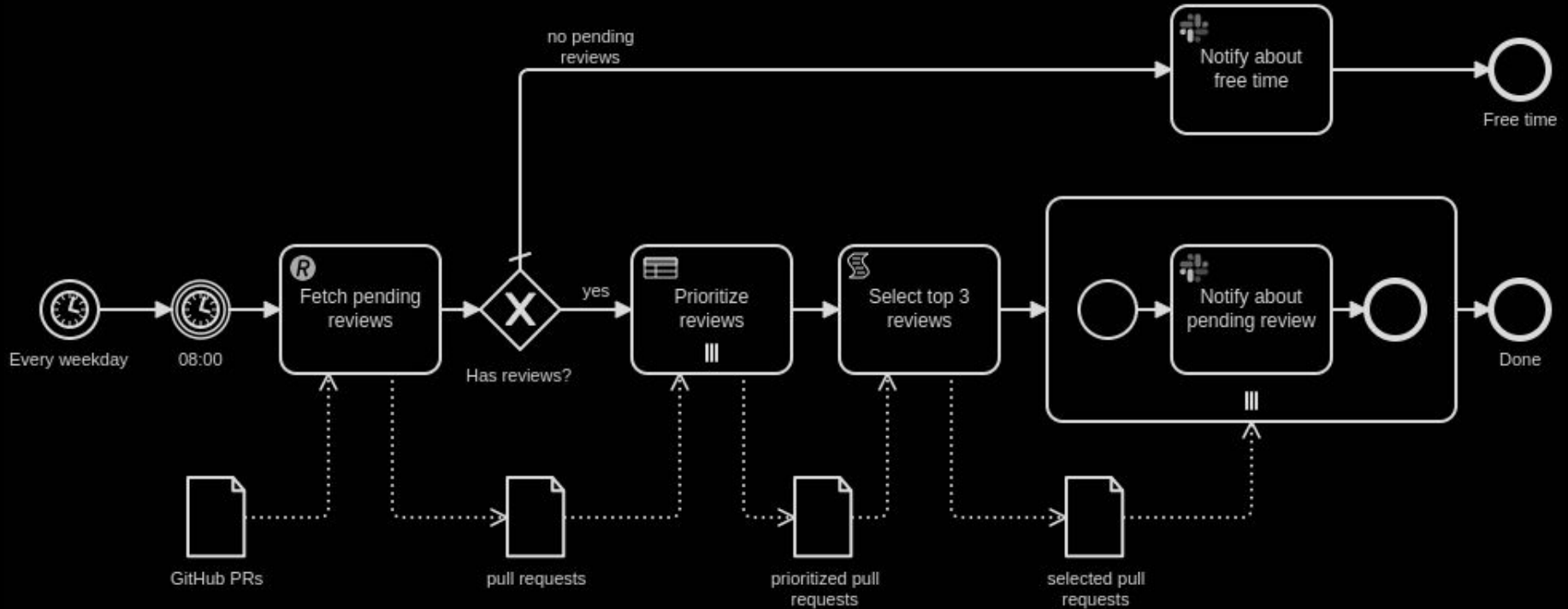
```
@"PT45M"
```

```
duration("P1Y")
```

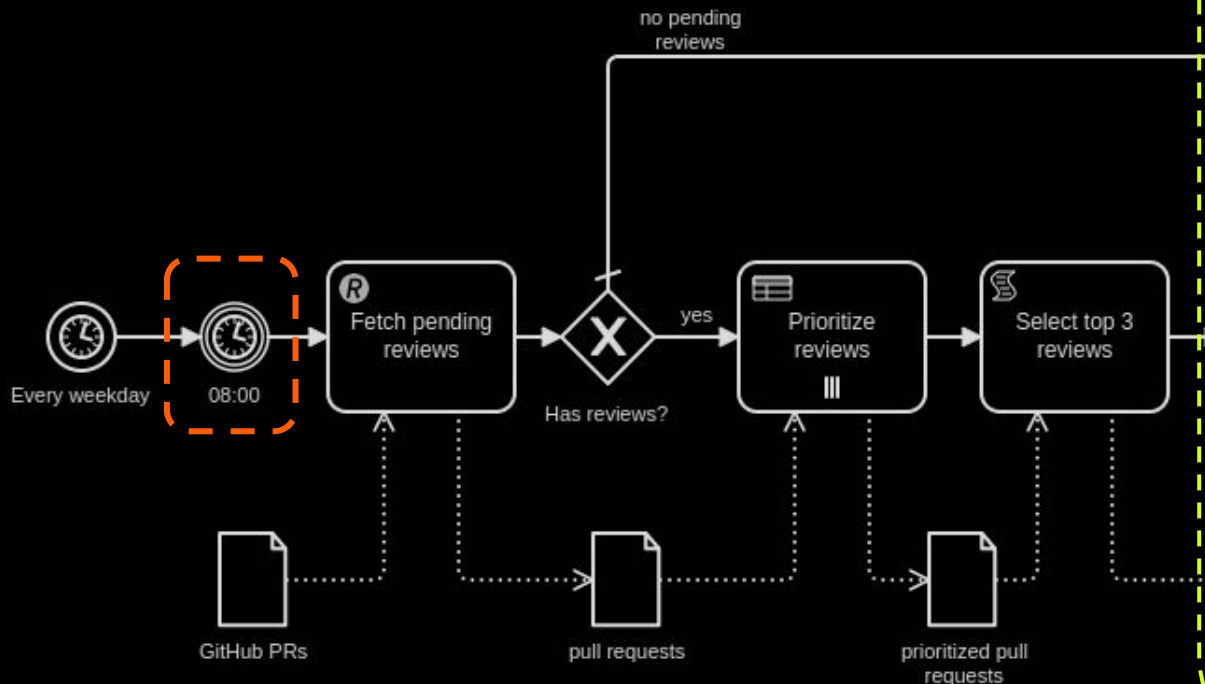
```
@"P1Y"
```

FEEL for BPMN

The process



Timer events



TIMER INTERMEDIATE CATCH EVENT
08:00

General >

Documentation >

Timer v

Duration ⊖

```
= date and time(  
  today(), @"08:00:00@Europe/Berlin"  
) - now()
```

A time duration defined as ISO 8601 durations format.

- PT15S - 15 seconds
- PT1H30M - 1 hour and 30 minutes
- P14D - 14 days

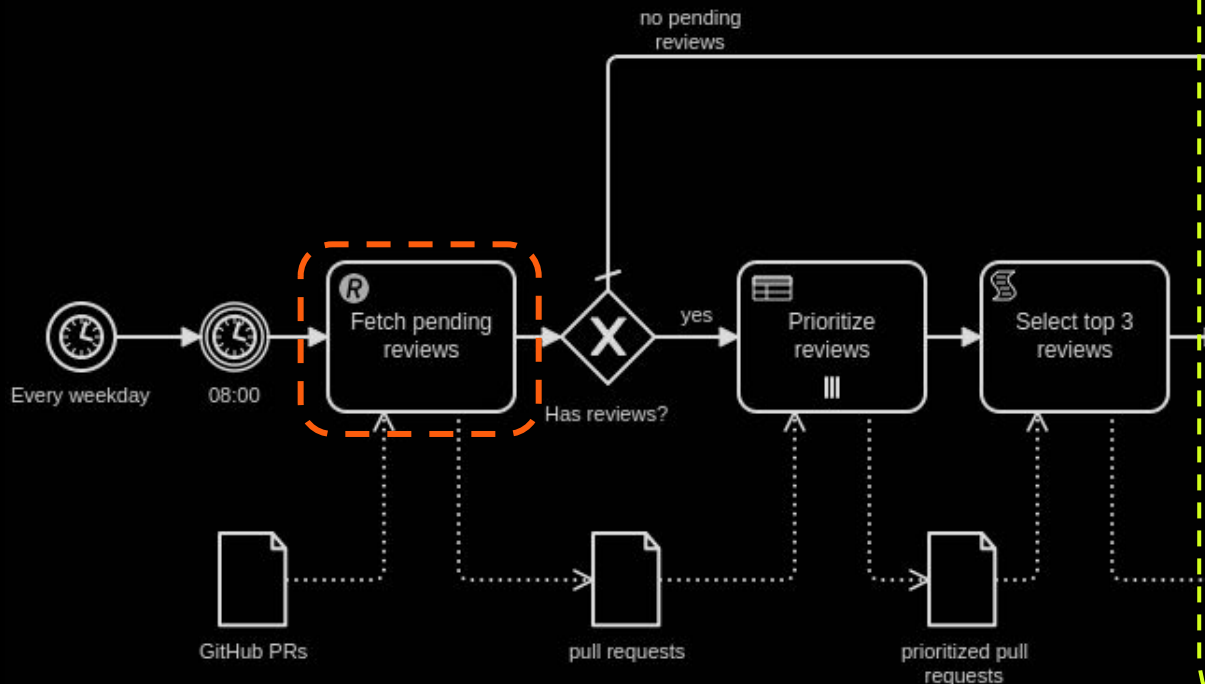
[How to configure a timer](#)

Outputs +

Extension properties +

Data >

Connectors



REST CONNECTOR
Fetch pending reviews

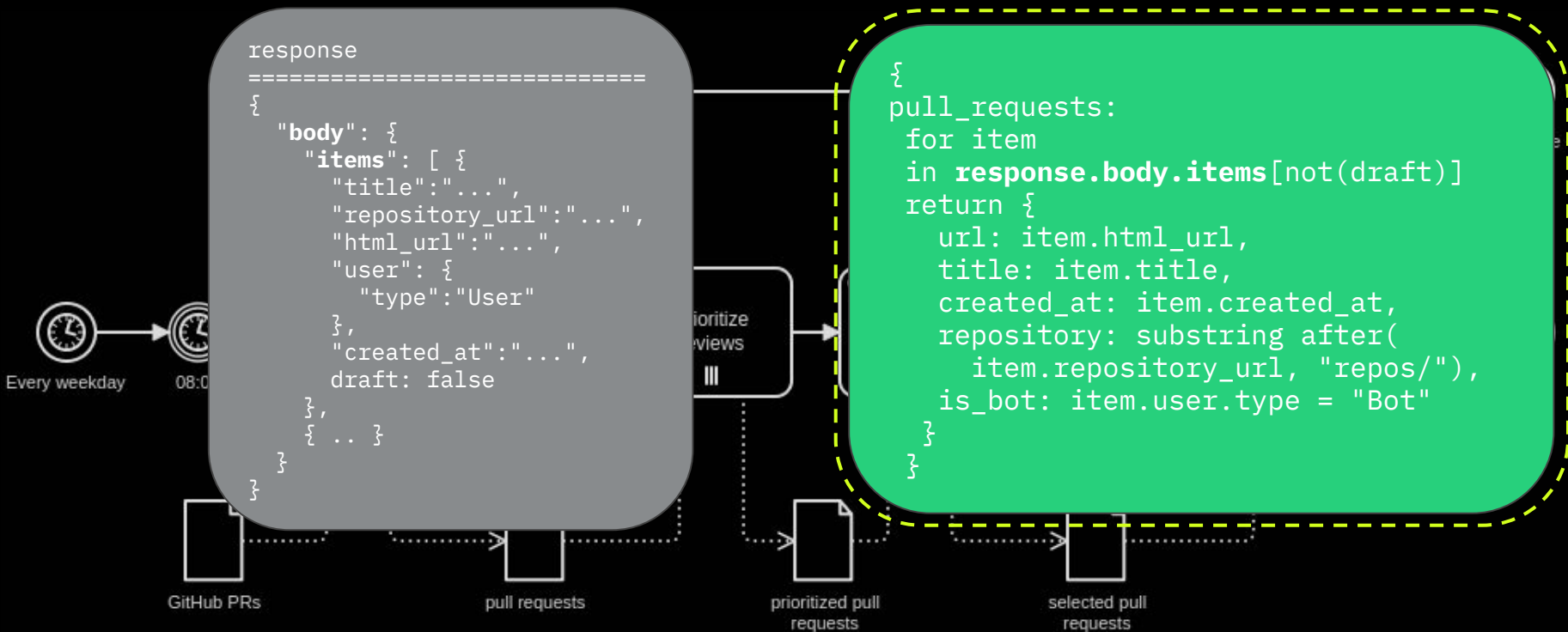
Connection
Sets the timeout in seconds to establish a connection or 0 for an infinite timeout

Response Mapping
Result Variable: github_response
Name of variable to store the response in
Result Expression

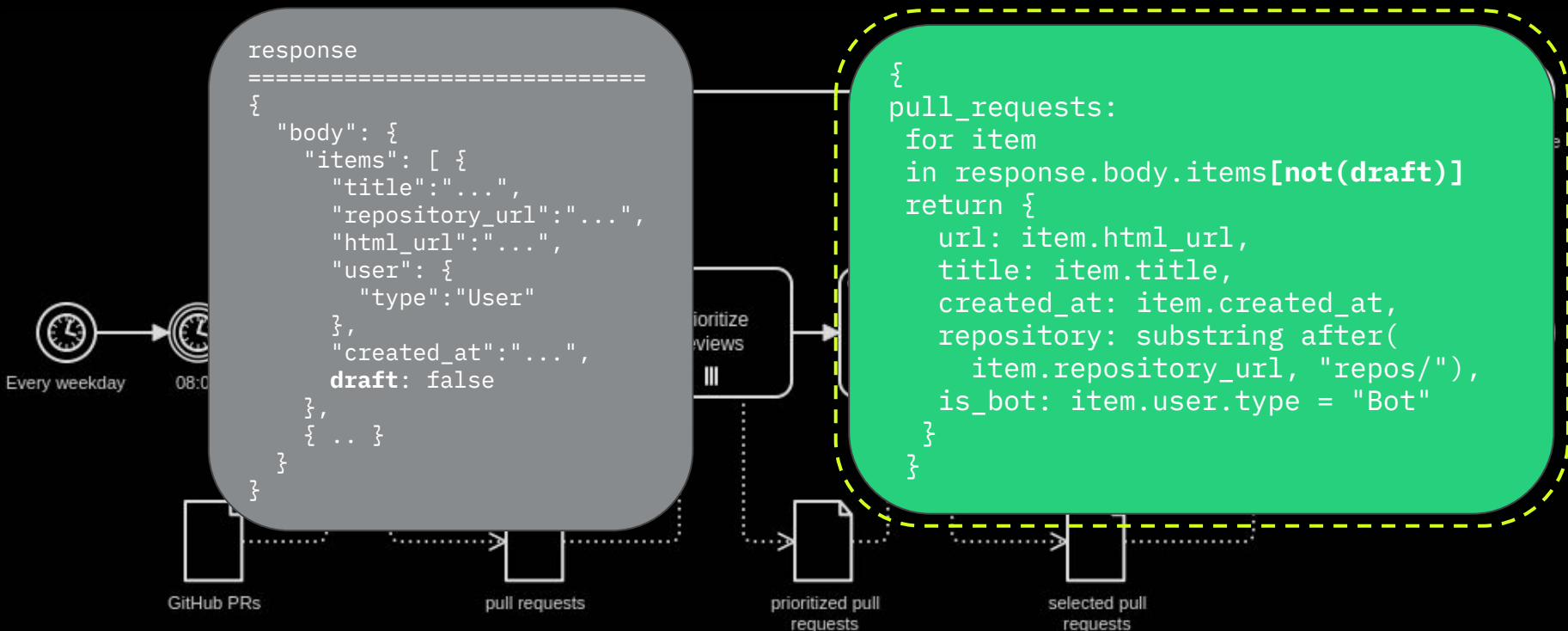
```
= {pull_requests:  
  for item  
  in response.body.items[not(draft)]  
  return {  
    url: item.html_url,  
    title: item.title,  
    created_at: item.created_at,  
    repository: substring after(  
      item.repository_url, "repos/"),  
    is_bot: item.user.type = "Bot"  
  }  
}
```

Expression to map the response into process variables

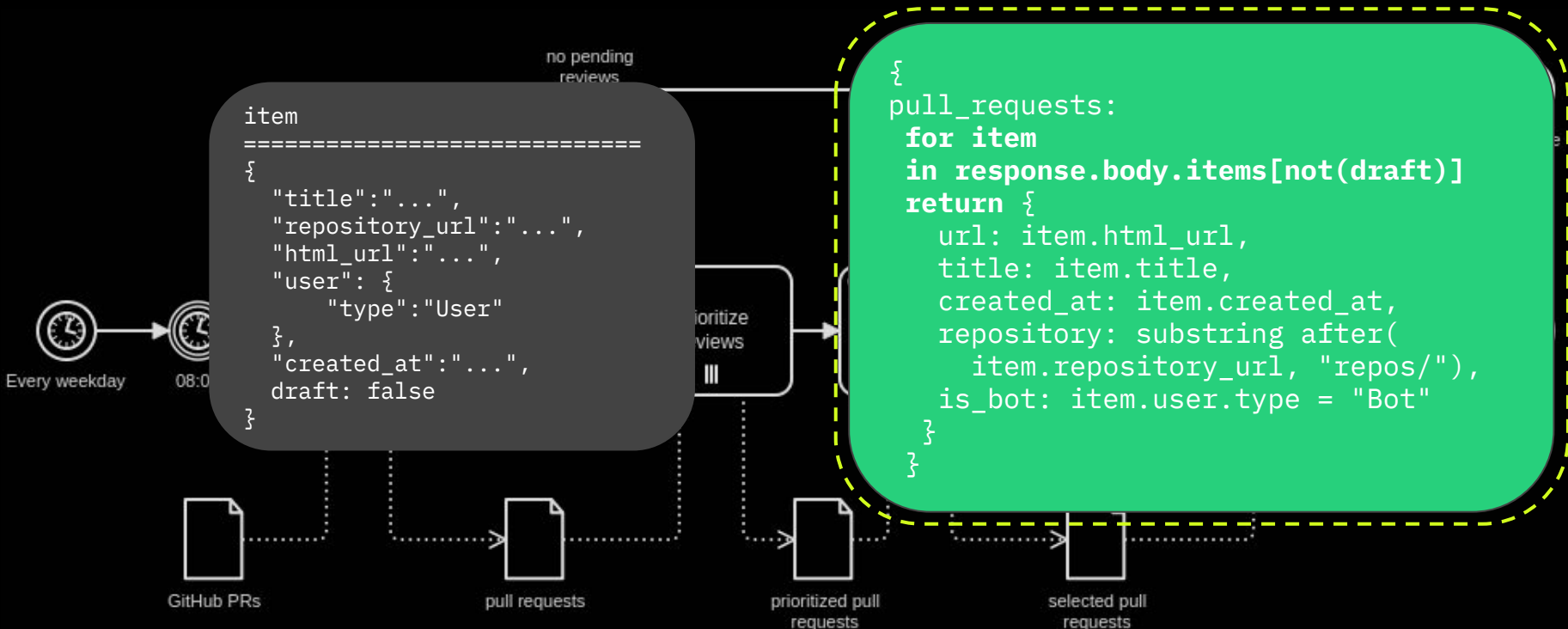
Connectors



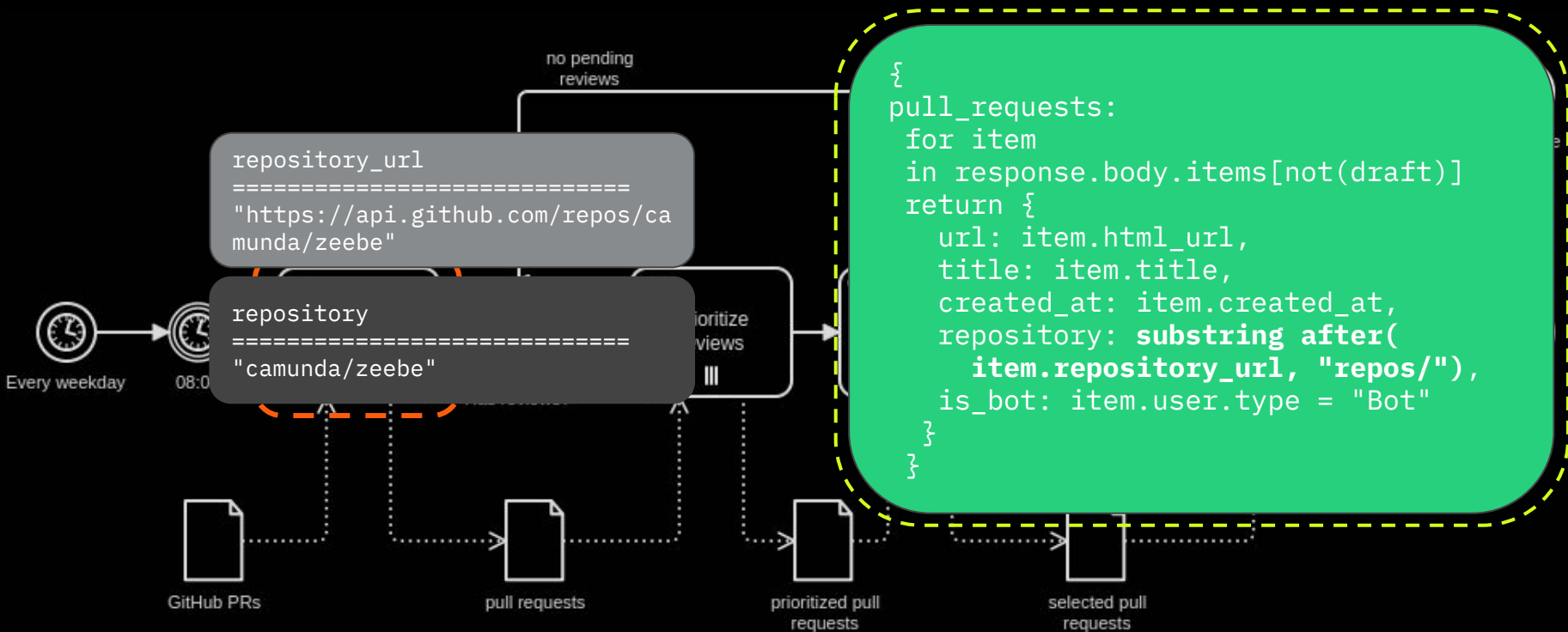
Connectors



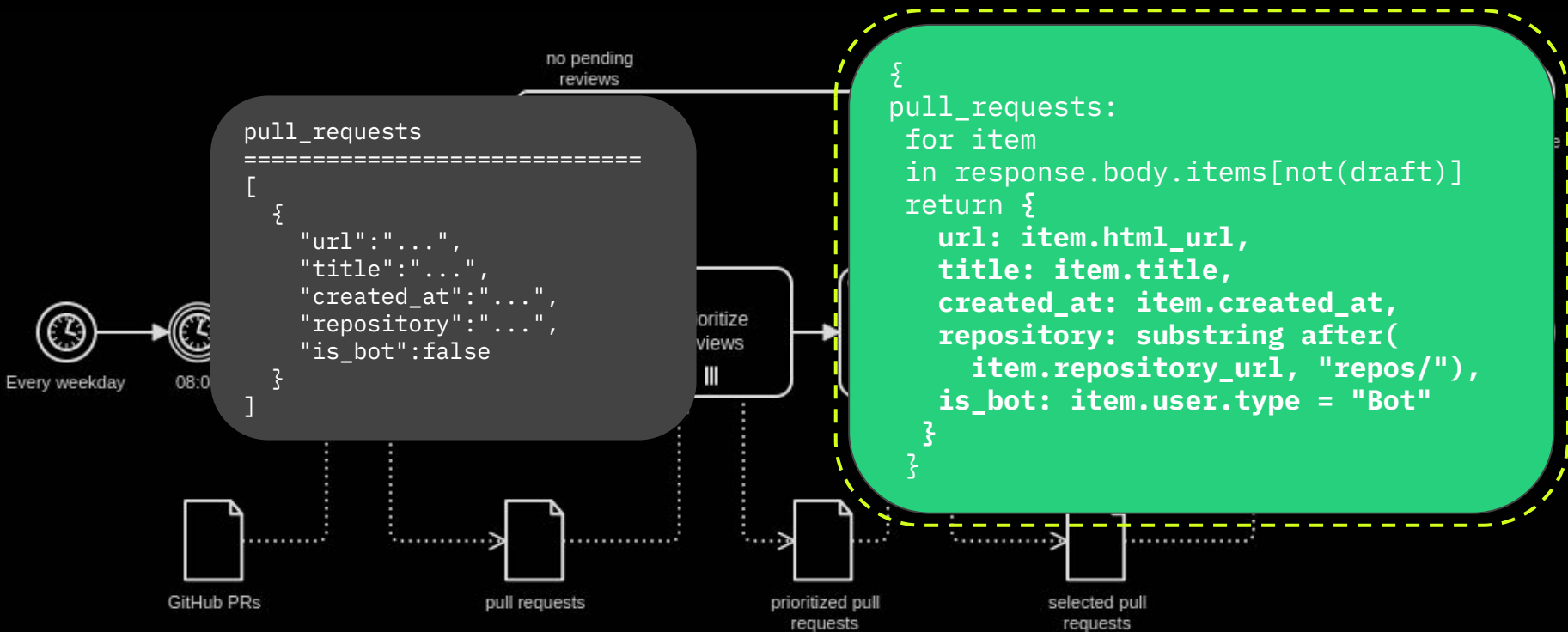
Connectors



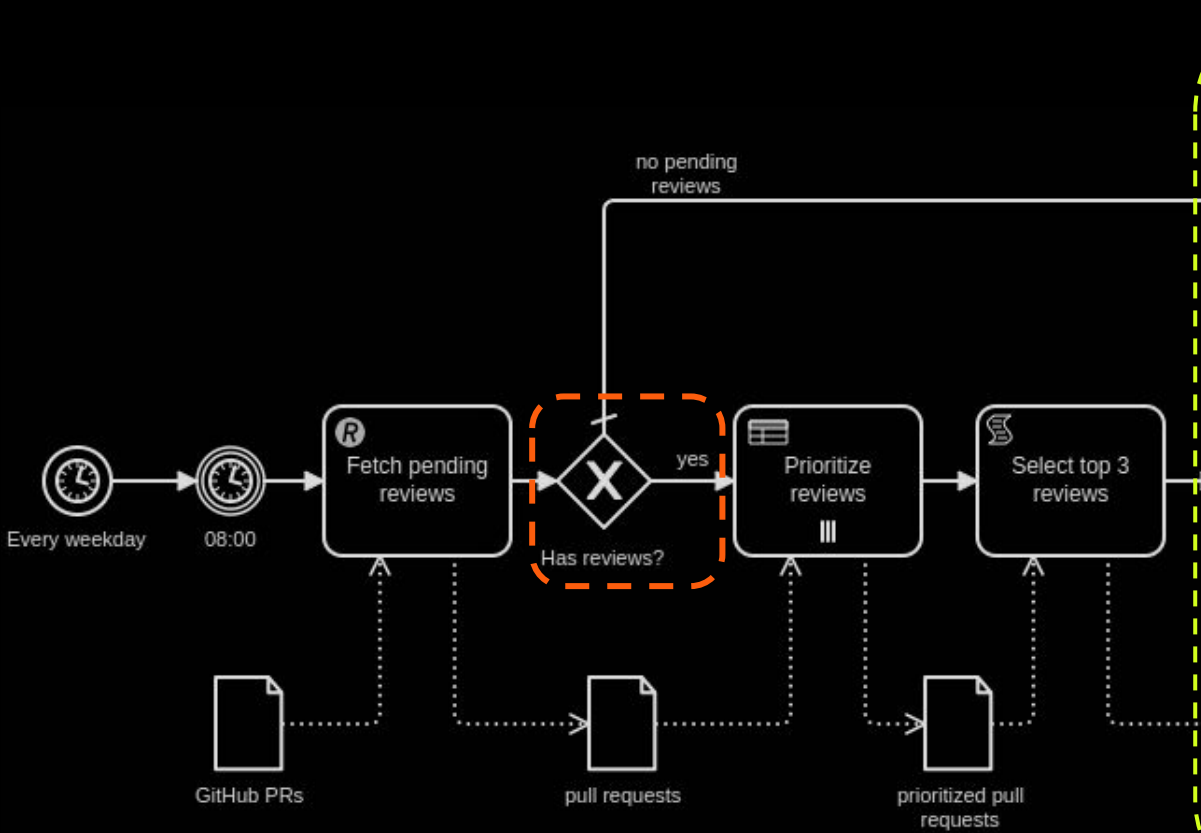
Connectors



Connectors



Gateways



SEQUENCE FLOW
yes

General • >

Documentation >

Condition • ∨

Condition expression

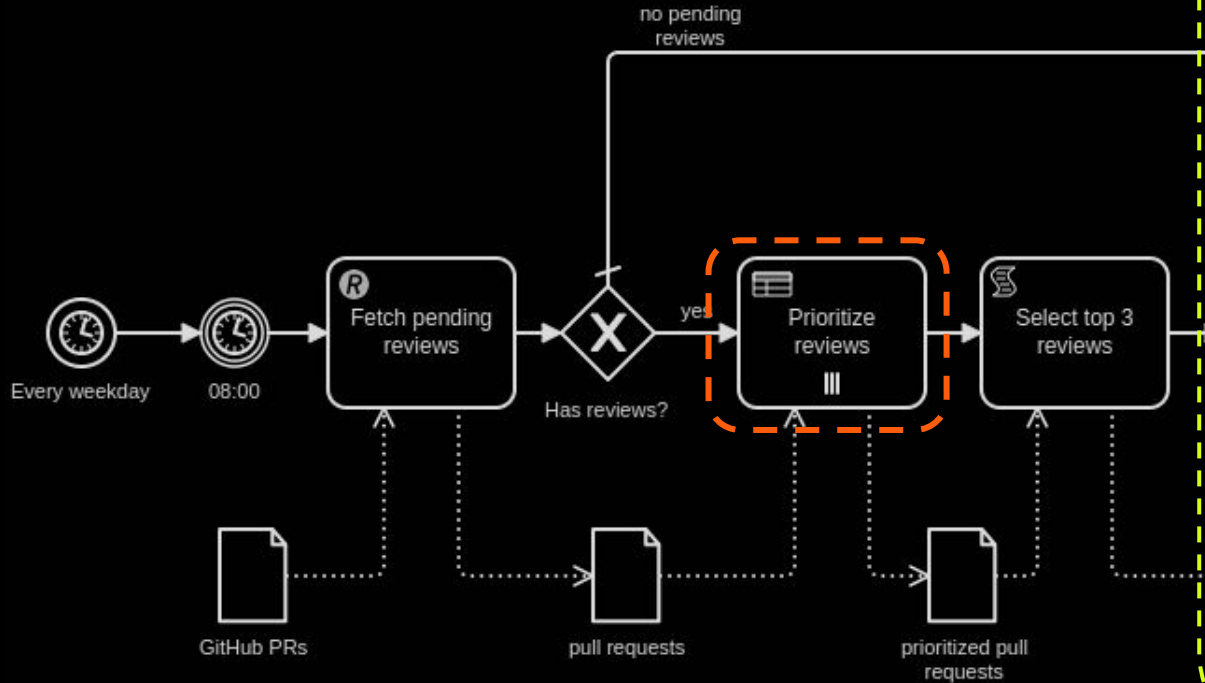
```
= count(pull_requests) >= 1
```

[How to define conditions](#)

Extension properties +

Data >

Business rule tasks



BUSINESS RULE TASK
Prioritize reviews

General

Documentation

Template + Select

Multi-instance • ▼

Input collection 🔗
= pull_requests
[How to configure a multi instance activity](#)

Input element
pull_request

Output collection
prioritized_pull_requests

Output element 🔗
= {
 pull_request: pull_request,
 priority: review_priority
}

Completion condition 🔗
=

Script tasks

SCRIPT TASK
Select top 3 reviews

General • >

Documentation • >

Template + Select

Implementation • >

Script • v

Result variable
selected_pull_requests

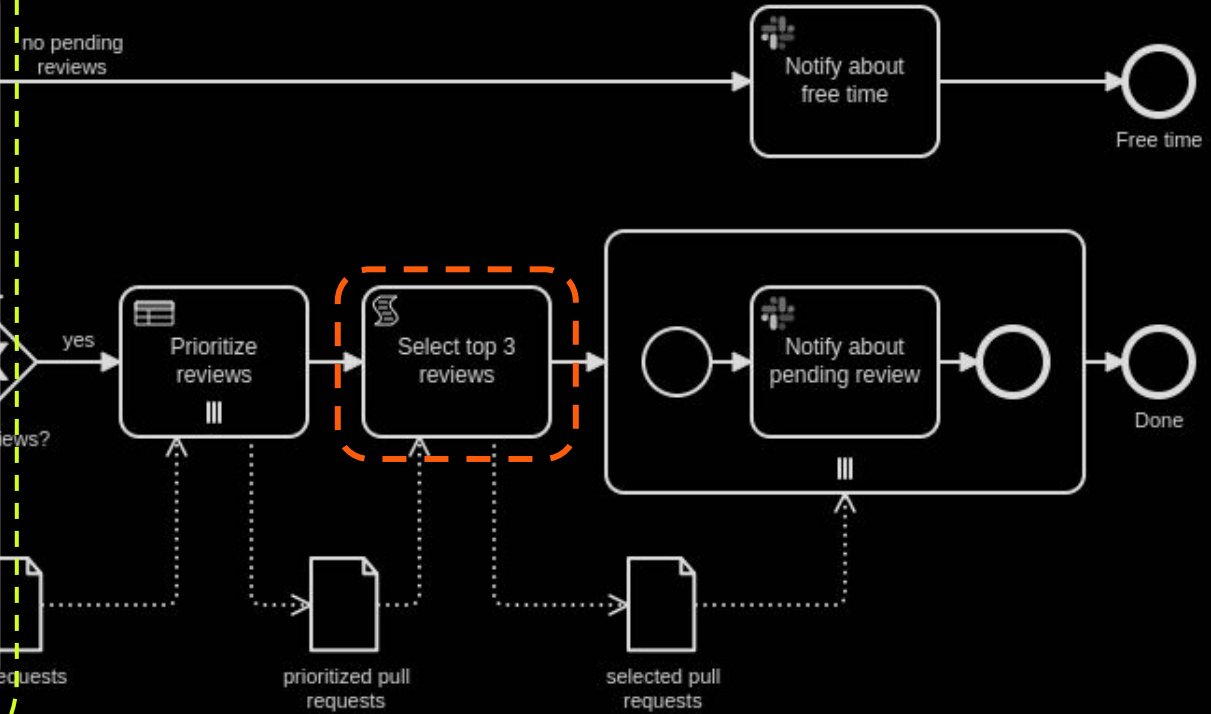
FEEL expression

```
= {  
  sorted: sort(  
    prioritized_pull_requests,  
    function (x,y)  
      x.priority < y.priority),  
  top_3: sublist(sorted, 1, 3)  
}.top_3.pull_request
```

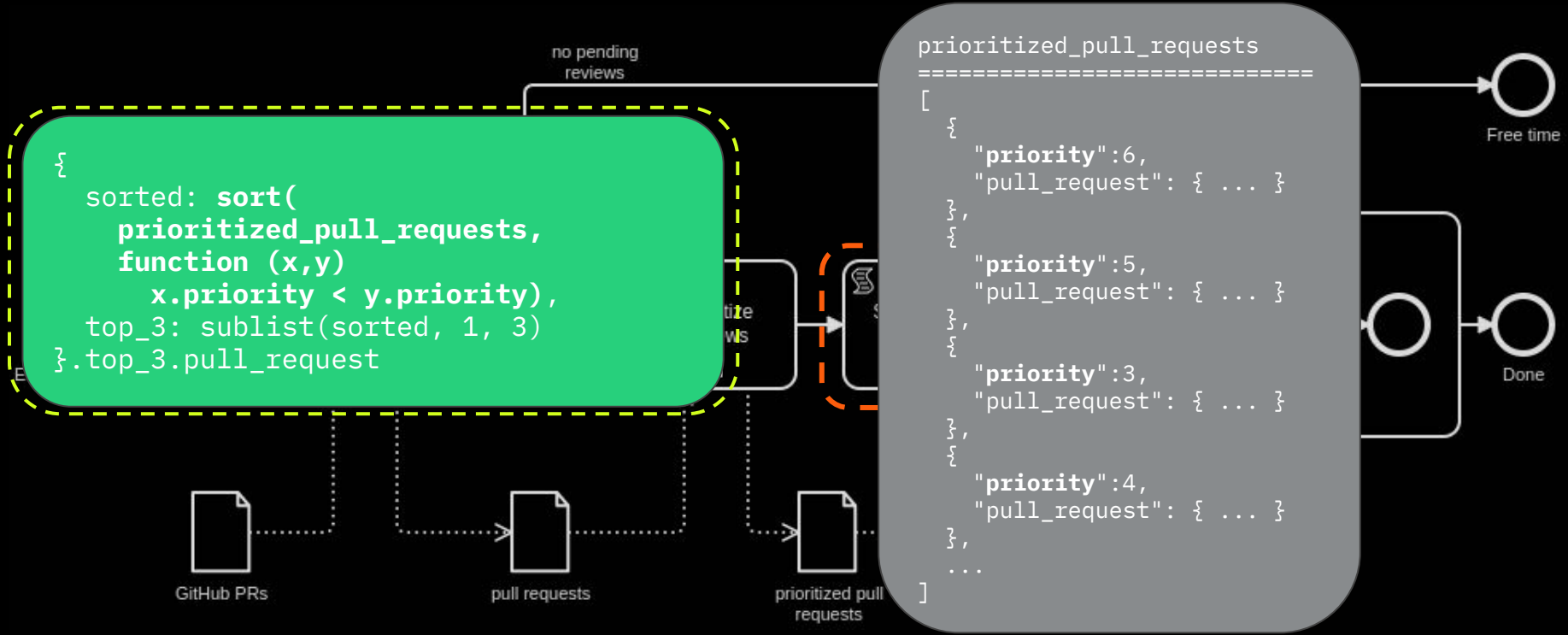
Inputs +

Outputs +

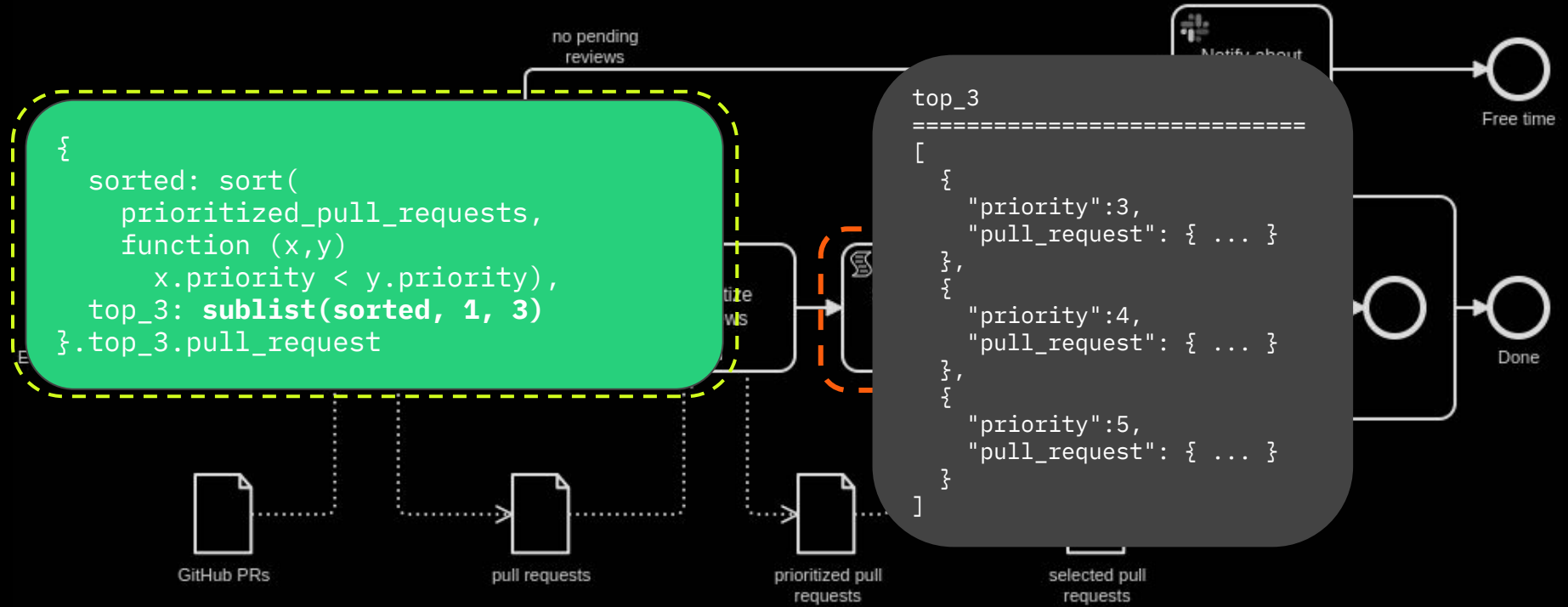
Extension properties +



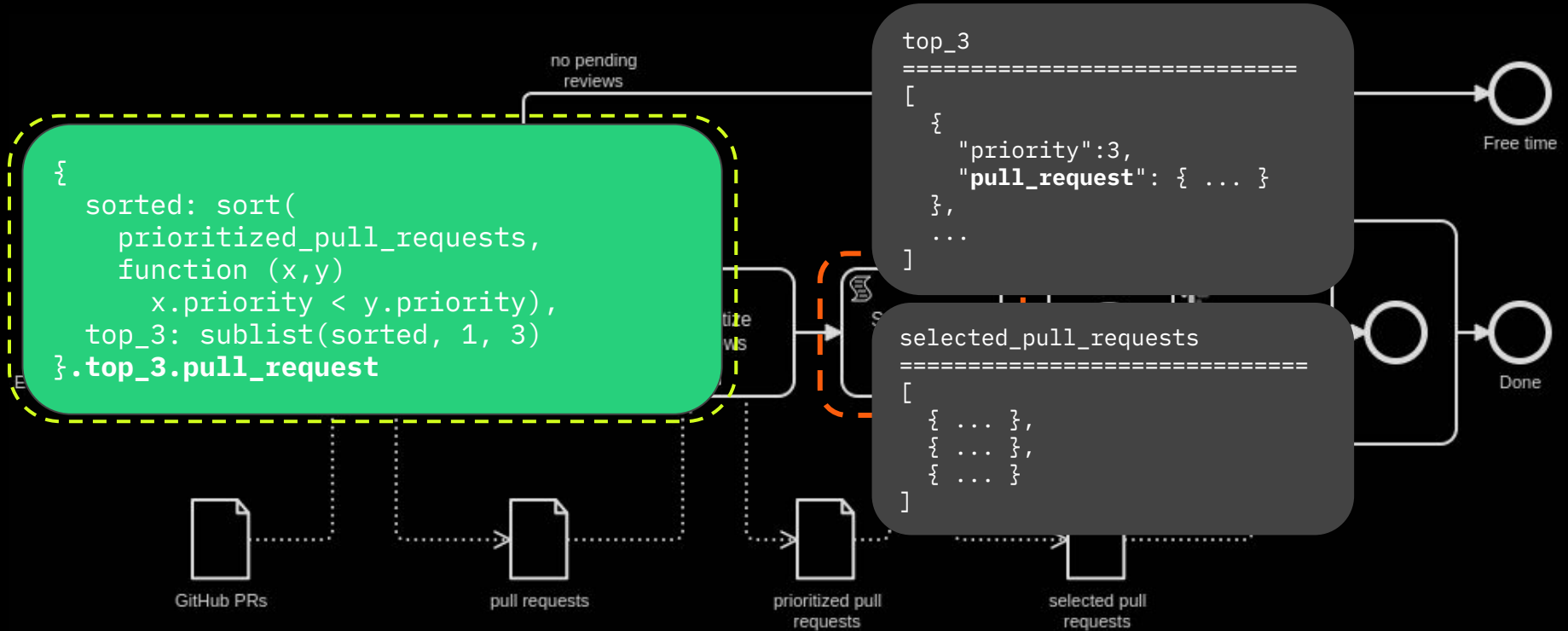
Script tasks



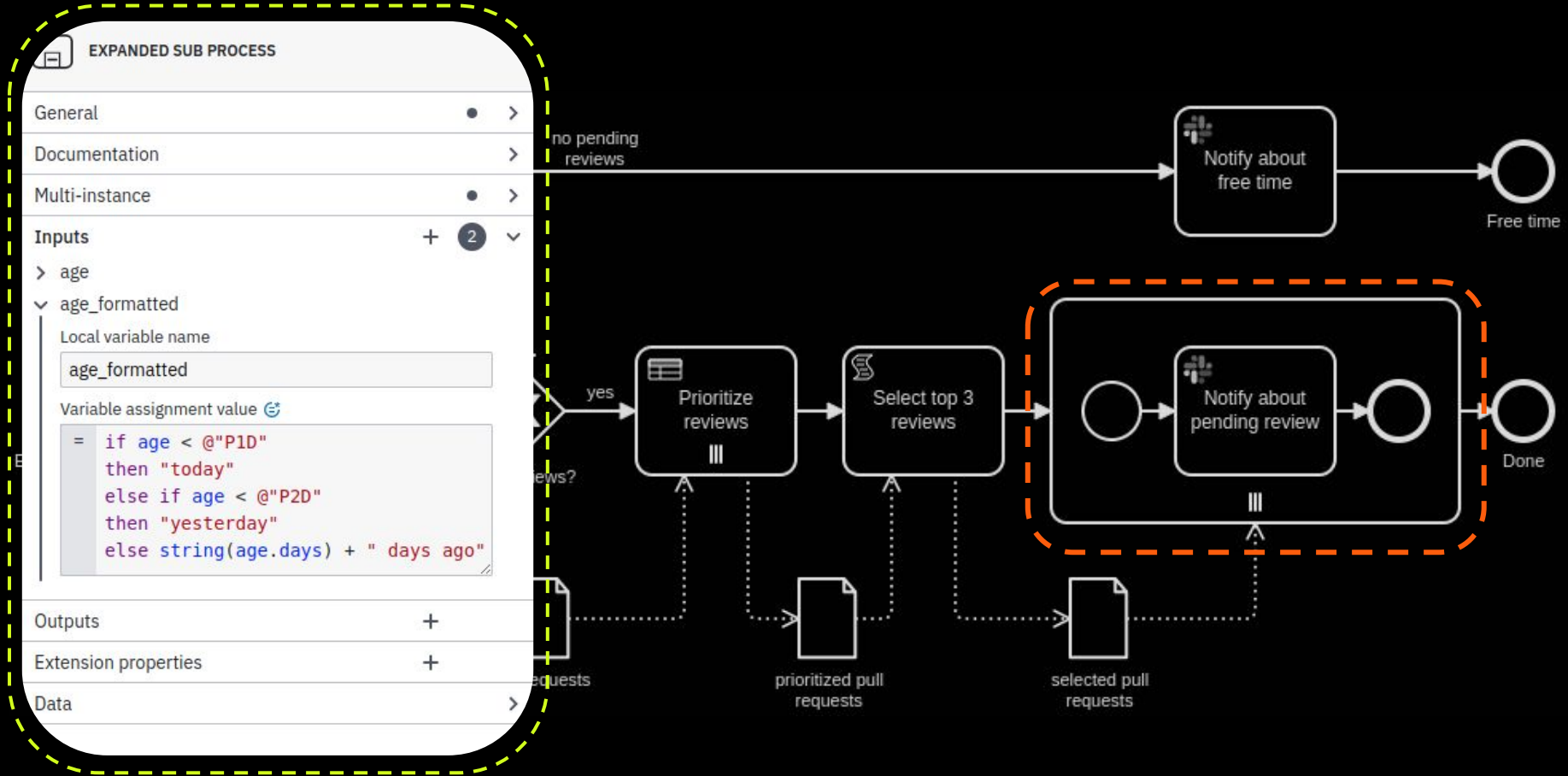
Script tasks



Script tasks

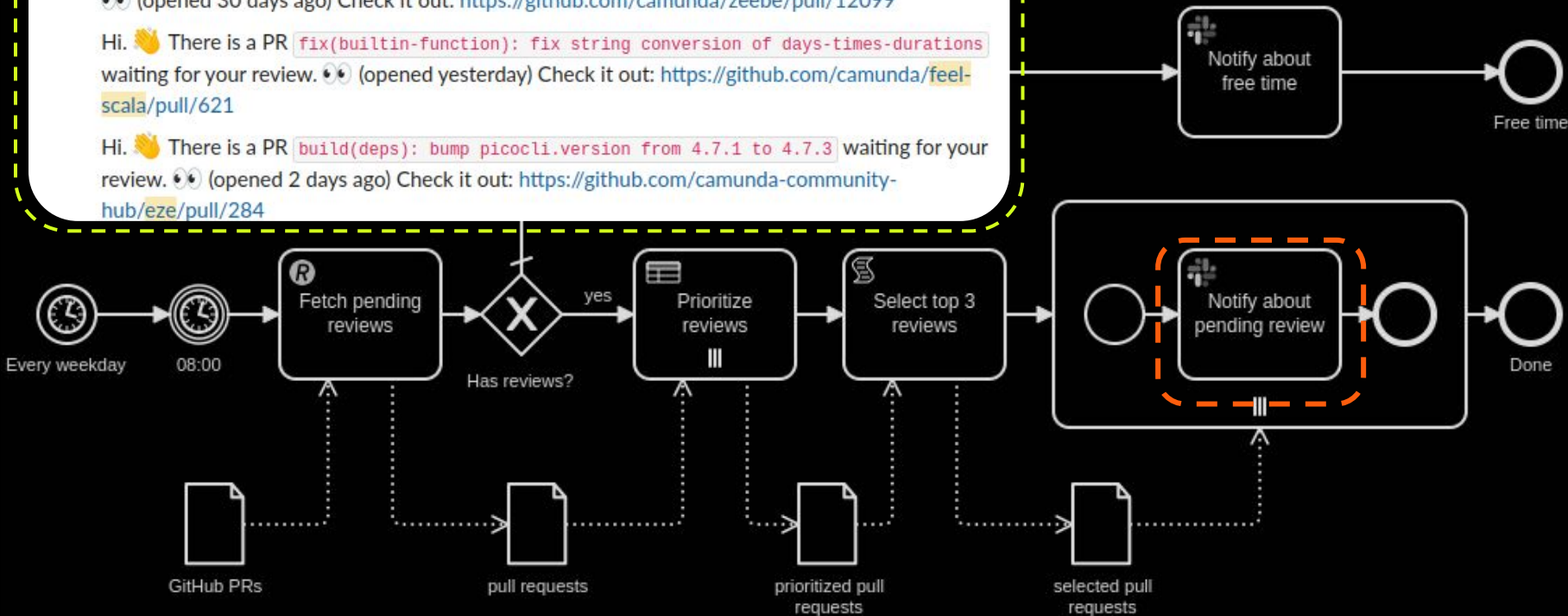


Variable mapping



The final result

Zee PA APP 10:34 AM
Hi. 🙌 There is a PR `feat(clients/go): exponential backoff supplier` waiting for your review. 🕒 (opened 30 days ago) Check it out: <https://github.com/camunda/zeebe/pull/12099>
Hi. 🙌 There is a PR `fix(builtin-function): fix string conversion of days-times-durations` waiting for your review. 🕒 (opened yesterday) Check it out: <https://github.com/camunda/feel-scala/pull/621>
Hi. 🙌 There is a PR `build(deps): bump picocli.version from 4.7.1 to 4.7.3` waiting for your review. 🕒 (opened 2 days ago) Check it out: <https://github.com/camunda-community-hub/eze/pull/284>



Tips & Tricks

Comparison

```
pull_request.user = "Bot"
```

```
pull_request.user != "Bot"
```

```
count(pull_requests) >= 1
```

```
age between @"P1D" and @"P2D"
```

```
user in ("Chris", "Nico")
```

```
age in [@"P1D" .. @"P2D")
```

List access

```
pull_requests
=====
[
  { "title": "First" },
  { "title": "Second" },
  { "title": "Third" },
  { "title": "Fourth" },
  { "title": "Firth" }
]
```

`pull_requests[1]`

`{ "title": "First" }`

`pull_requests[-1]`

`{ "title": "Firth" }`

`pull_requests[6]`

`null`

Summary

Summary

Why FEEL?

- A common language for DMN and BPMN
- Aims to be simple
- Powerful



Empower BPMN

- Gateways
- Connectors
- Script tasks
- Variable mappings
- Time events



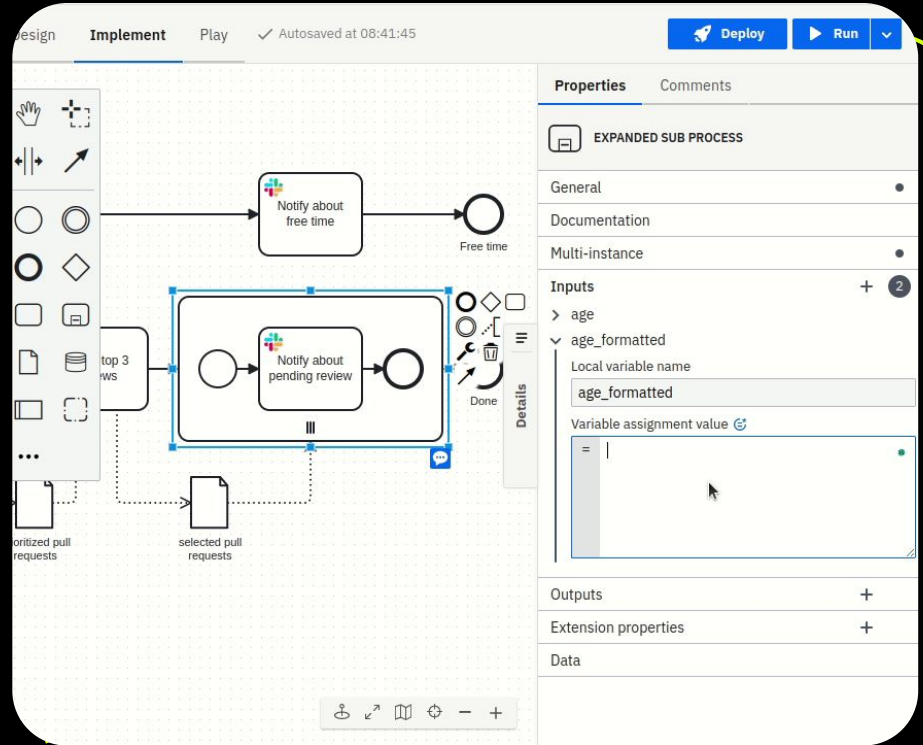
Pitfalls

- Comparison = vs. ==
- List access [1] vs. [0]
- String concatenation `string(days) + " days"`



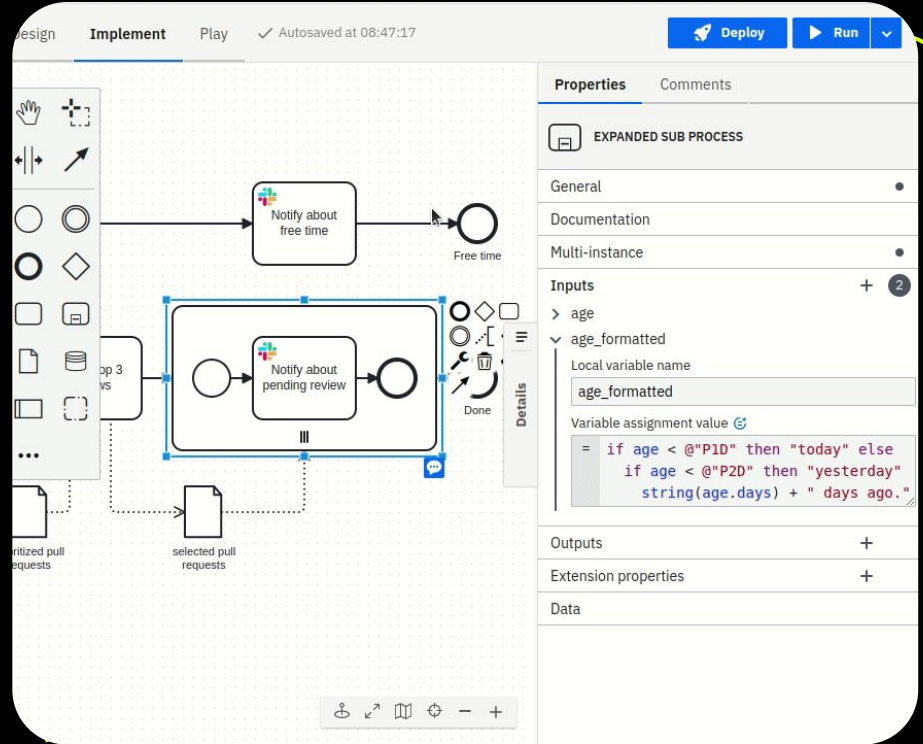
Tooling

- **Camunda Modeler**
- Play mode (Zeebe-Play)
- Playground:
 - FEEL-Scala
<https://camunda.github.io/feel-scala/docs/playground>
 - FEEL-JS
<https://nikku.github.io/feel-playground>



Tooling

- Camunda Modeler
- **Play mode (Zeebe-Play)**
- Playground:
 - FEEL-Scala
<https://camunda.github.io/feel-scala/docs/playground>
 - FEEL-JS
<https://nikku.github.io/feel-playground>



Learning

- Camunda documentation
<https://docs.camunda.io>
- Blog post: Using FEEL with Camunda 8
<https://camunda.com/blog/2022/09/using-feel-with-camunda-8/>
- FEEL samples + challenge
<https://camunda.github.io/feel-scala/docs/learn/>
- Talk BPMN + DMN
<https://github.com/saig0/ccs2023-feel-beyond-dmn>

Camunda Platform 8 Docs 8.2 Guides Components APIs & Tools Self-Managed Reference

Overview Components

Concepts >

Console >

Modeler >

About Modeler >

Web Modeler >

Desktop Modeler >

BPMN >

DMN >

FEEL expressions >

What is FEEL?

Data types

Unary-tests

Expressions >

Built-in Functions >

Introduction

Conversion functions

Boolean functions

String functions

Home > Modeler > FEEL expressions > Built-in Functions > String functions

Version: 8.2

String functions

substring(string, start position)

Returns a substring of the given value starting at `start position`.

Function signature

```
substring(string: string, start position: number): string
```

The `start position` starts at the index `1`. The last position is `-1`.

Examples

```
substring("foobar", 3)
// "obar"
```

substring(string, start position, length)

Returns a substring of the given value starting at `start position`.

Questions?



Thank You



<https://github.com/saig0>



<https://www.linkedin.com/in/philipp-ossler-b169872>



<https://camunda.com/platform/zeebe>