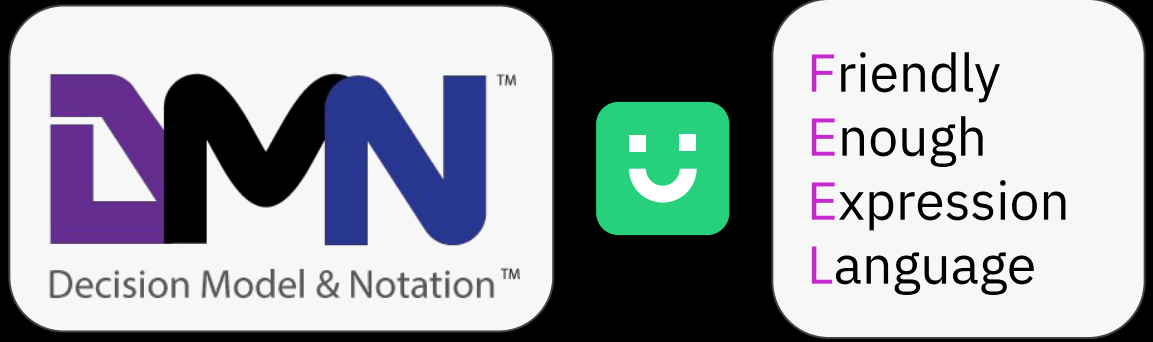
FEEL Beyond DMN

Philipp Ossler
11th May 2023

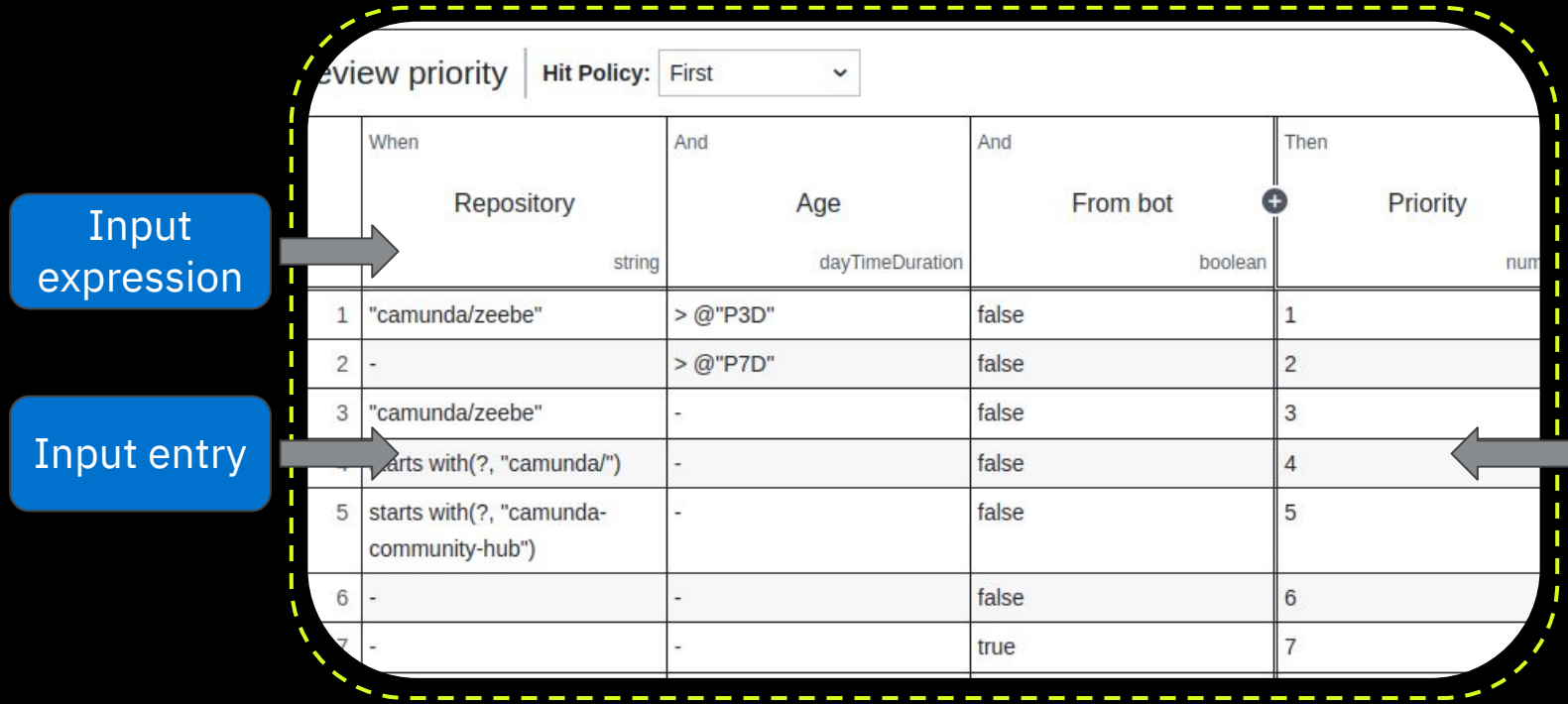
CAMUNDA COMMUNITY SUMMIT 2023

What is FEEL?


DMN™
Decision Model & Notation™
Friendly
Enough
Expression
Language

FEEL for DMN

CCS 2023



FEEL for BPMN?

CCS 2023

CCS 2023

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

CCS 2023

Basics

Expression vs. script

CCS 2023

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

CCS 2023

```
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"
```

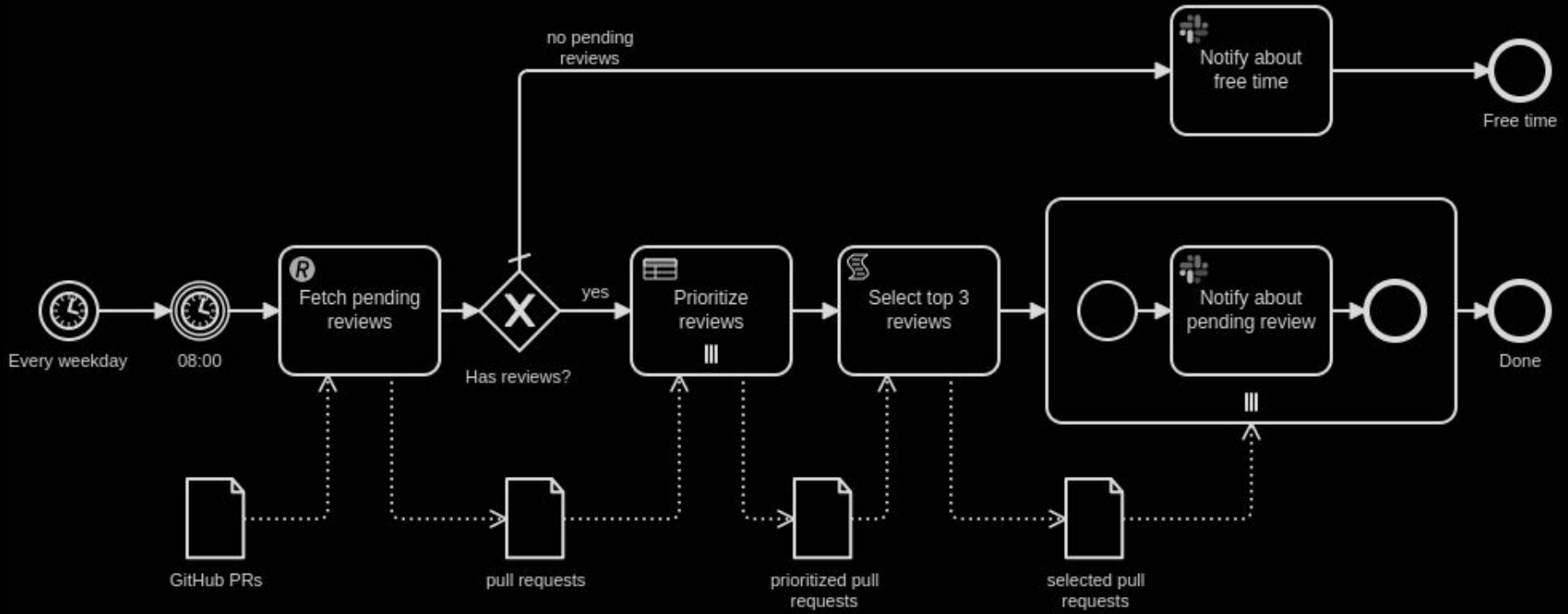
```
duration("P1Y")

@"P1Y"
```

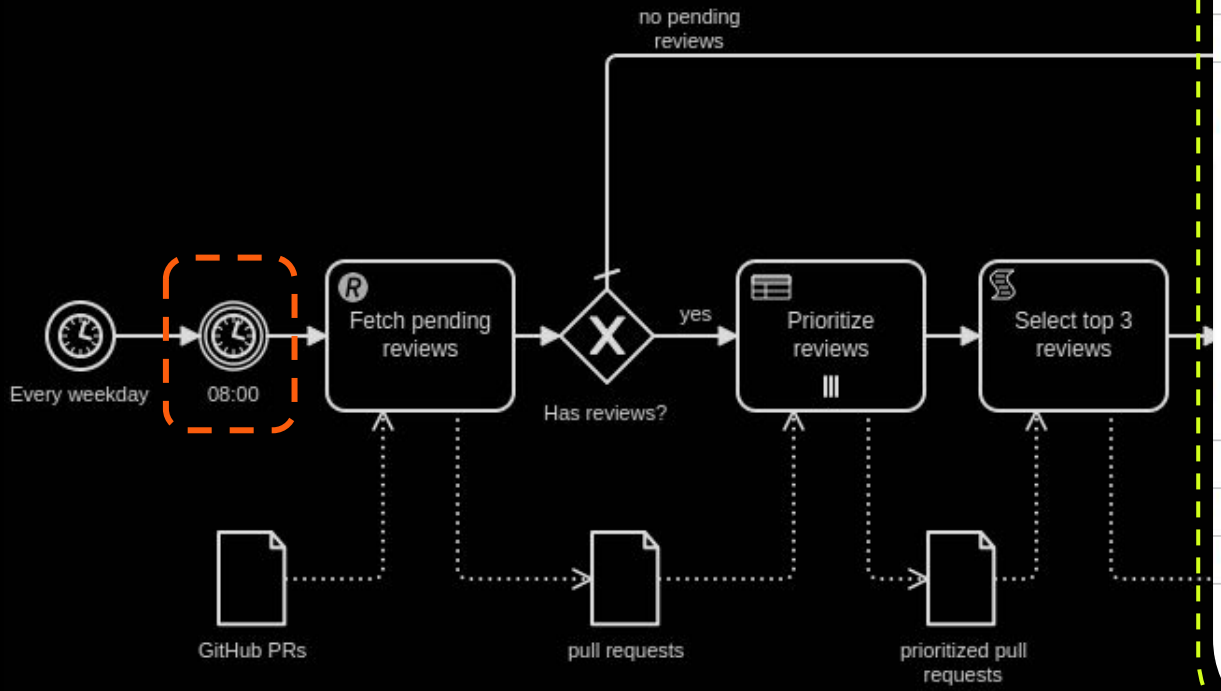
FEEL for BPMN

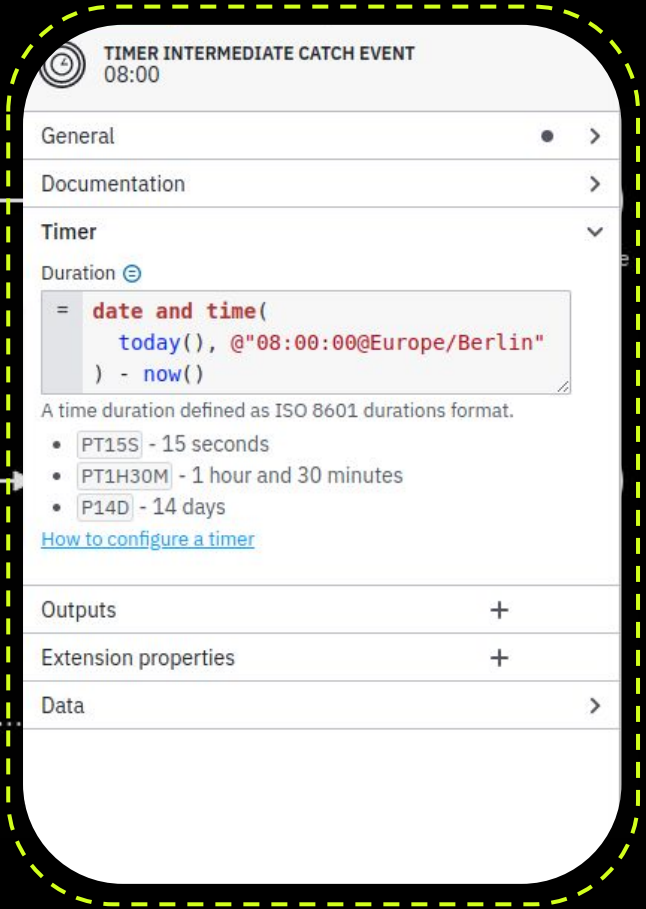
The process

CCS 2023

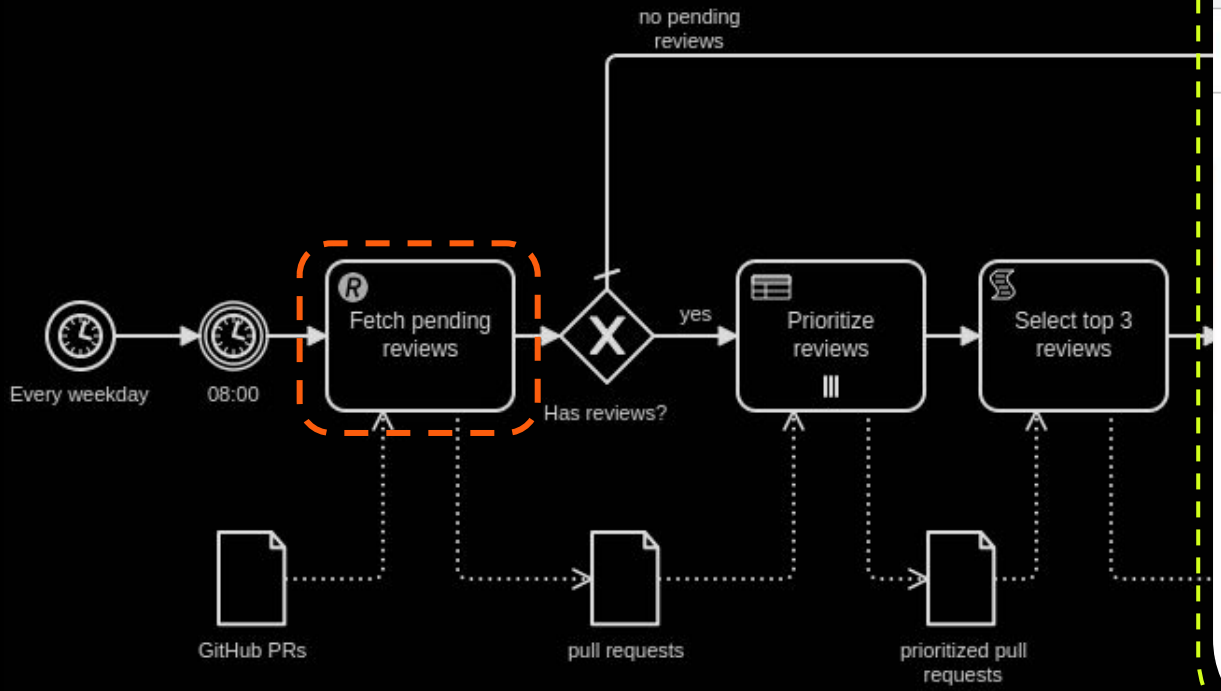


Timer events





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

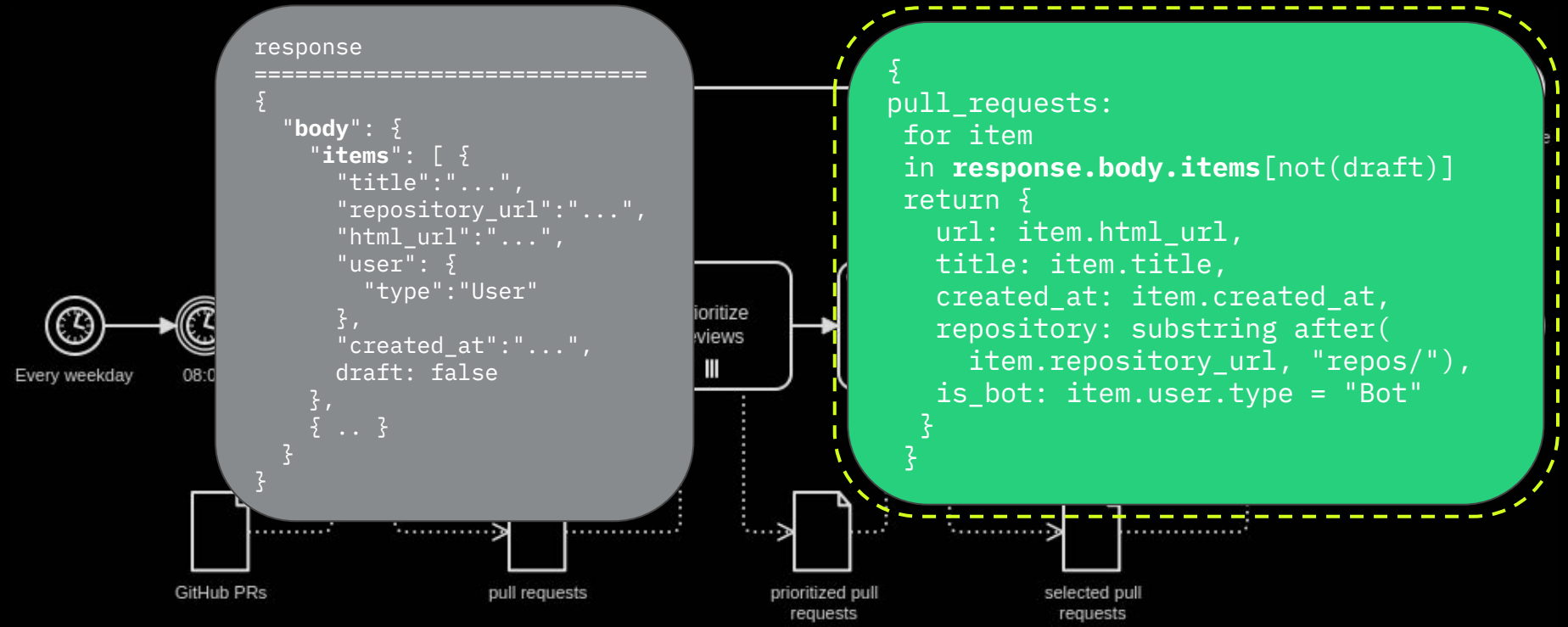
Name of variable to store the response in

Result Expression

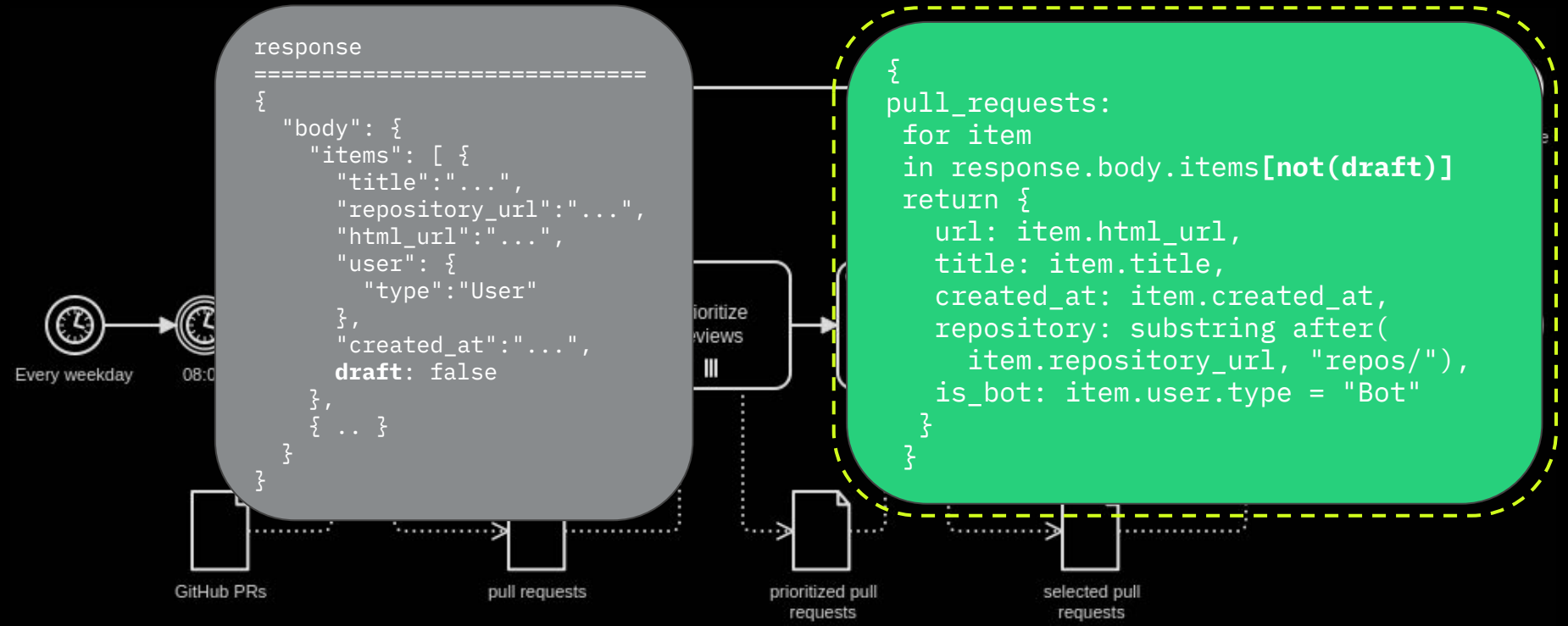
```
= {pull_requests:
   for item
   in response.body.items[not(draft)]
   return {
     url: item.html_url,
     title: item.title,
     created_at: item.created_at,
     repository: substring after(
       item.repository_url, "repos/"),
     is_bot: item.user.type = "Bot"
   }
  }
```

xpression to map the response into process variables

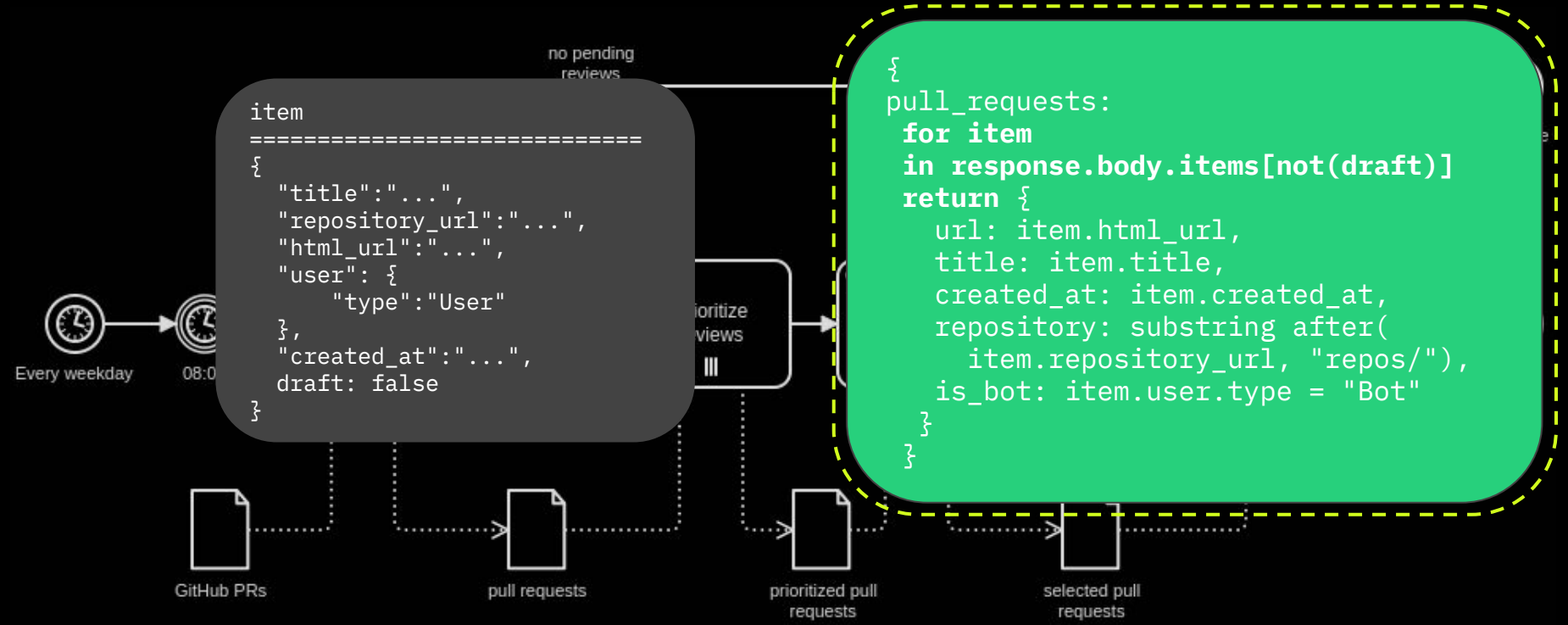
Connectors


response
==========
{
"body": {
"items": [{
"title":"...",
"repository_url":"...",
"html_url":"...",
"user": {
"type":"User"
},
"created_at":"...",
draft: false
},
{ .. }
}
}
{
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"
}
}
Every weekday
GitHub PRs
pull requests
prioritized pull requests
selected pull requests

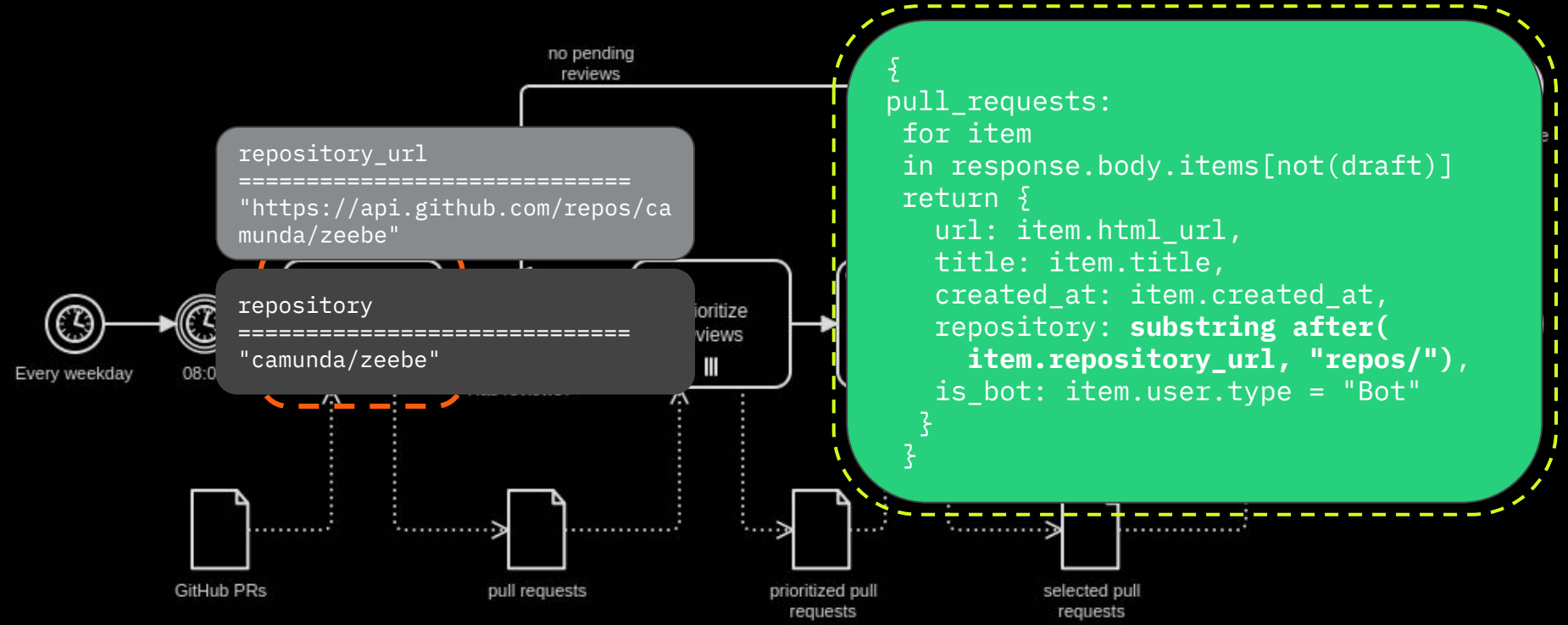
Connectors


response
==========
{
"body": {
"items": [{
"title":"...",
"repository_url":"...",
"html_url":"...",
"user": {
"type":"User"
},
"created_at":"...",
draft: false
},
{ .. }
}
}
{
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"
}
}
Every weekday
GitHub PRs
pull requests
prioritized pull requests
selected pull requests

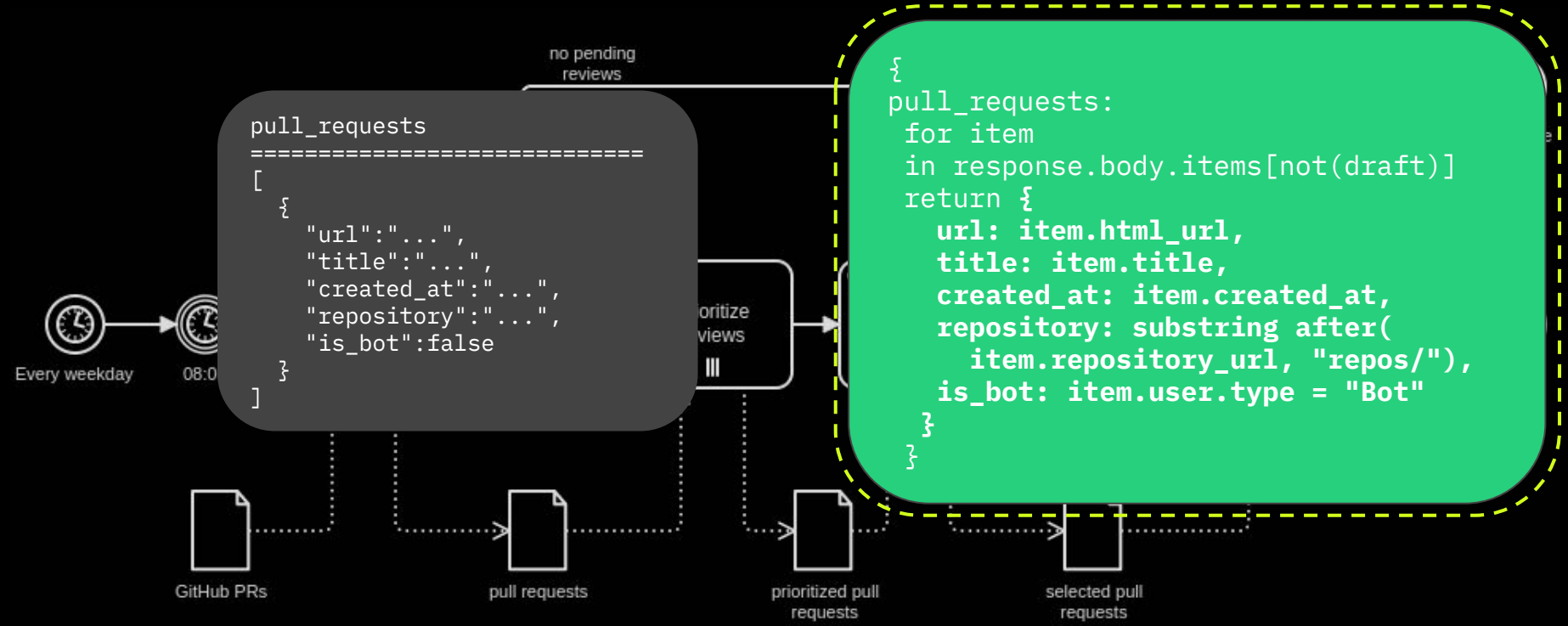
Connectors


no pending reviews
item
==========
{
"title":"...",
"repository_url":"...",
"html_url":"...",
"user": {
"type":"User"
},
"created_at":"...",
draft: false
}
{
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"
}
}
Every weekday
08:0
oritize
views
GitHub PRs
pull requests
prioritized pull requests
selected pull requests

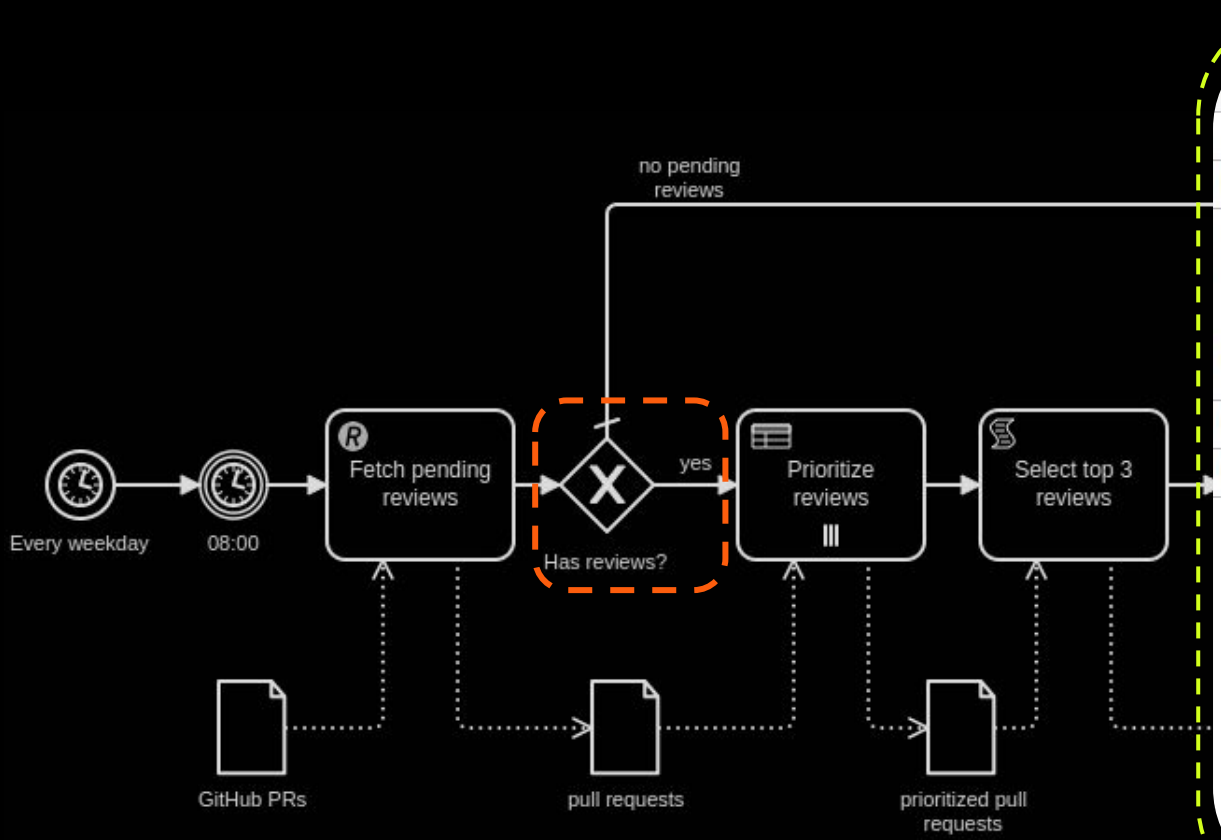
Connectors


no pending reviews
repository_url
==========
"https://api.github.com/repos/camunda/zeebe"
repository
==========
"camunda/zeebe"
{
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"
}
}
Every weekday
GitHub PRs
pull requests
prioritized pull requests
selected pull requests

Connectors


no pending reviews
pull_requests
==========
[
{
"url":"...",
"title":"...",
"created_at":"...",
"repository":"...",
"is_bot":false
}
]
{
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"
}
}
Every weekday
oritize
views
GitHub PRs
pull requests
prioritized pull requests
selected pull requests

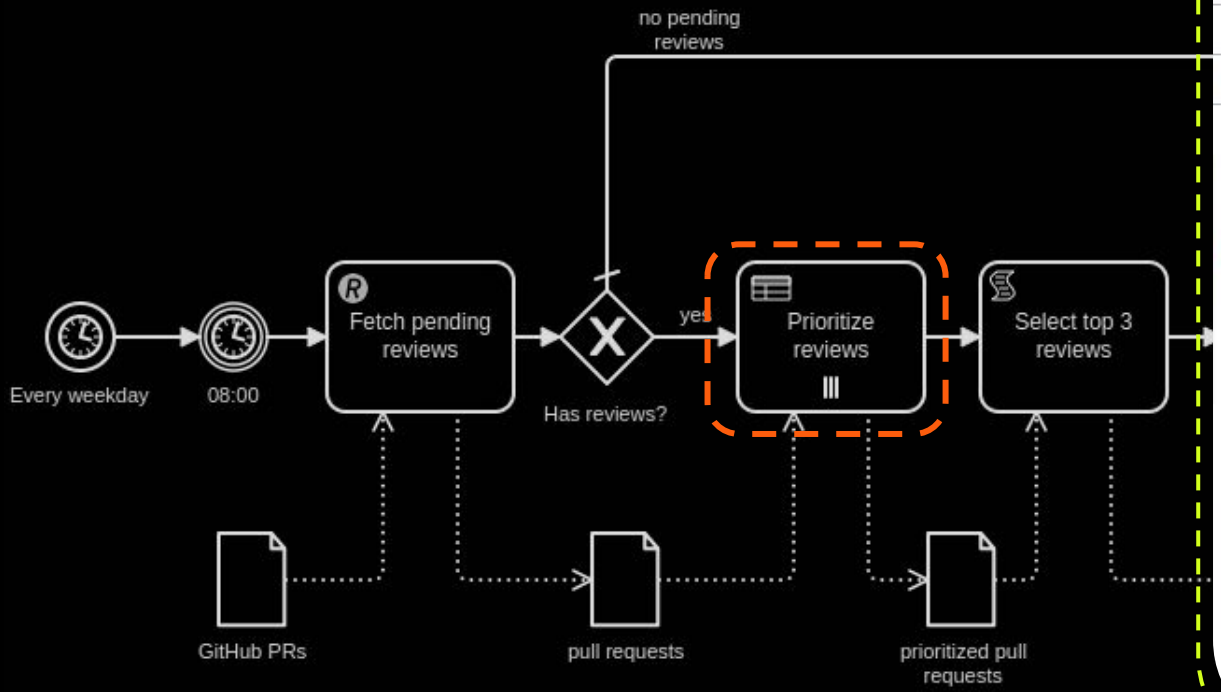
Gateways


no pending reviews
Every weekday
08:00
Fetch pending reviews
Has reviews?
yes
Prioritize reviews
Select top 3 reviews
GitHub PRs
pull requests
prioritized pull requests


SEQUENCE FLOW
yes
General
Documentation
Condition
Condition expression
= count(pull_requests) >= 1
How to define conditions
Extension properties
Data

Business rule tasks

CCS 2023



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

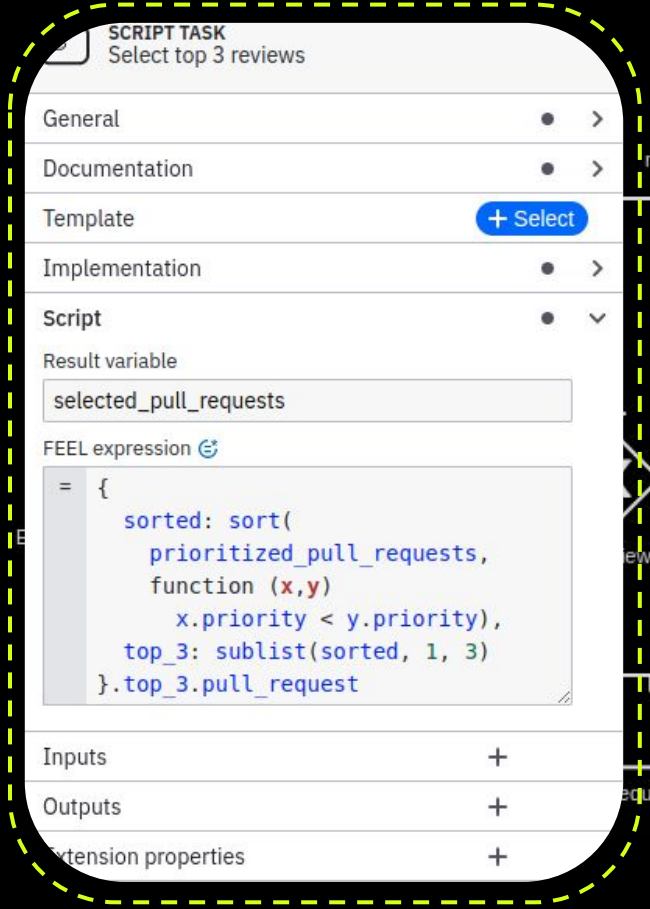
Output element

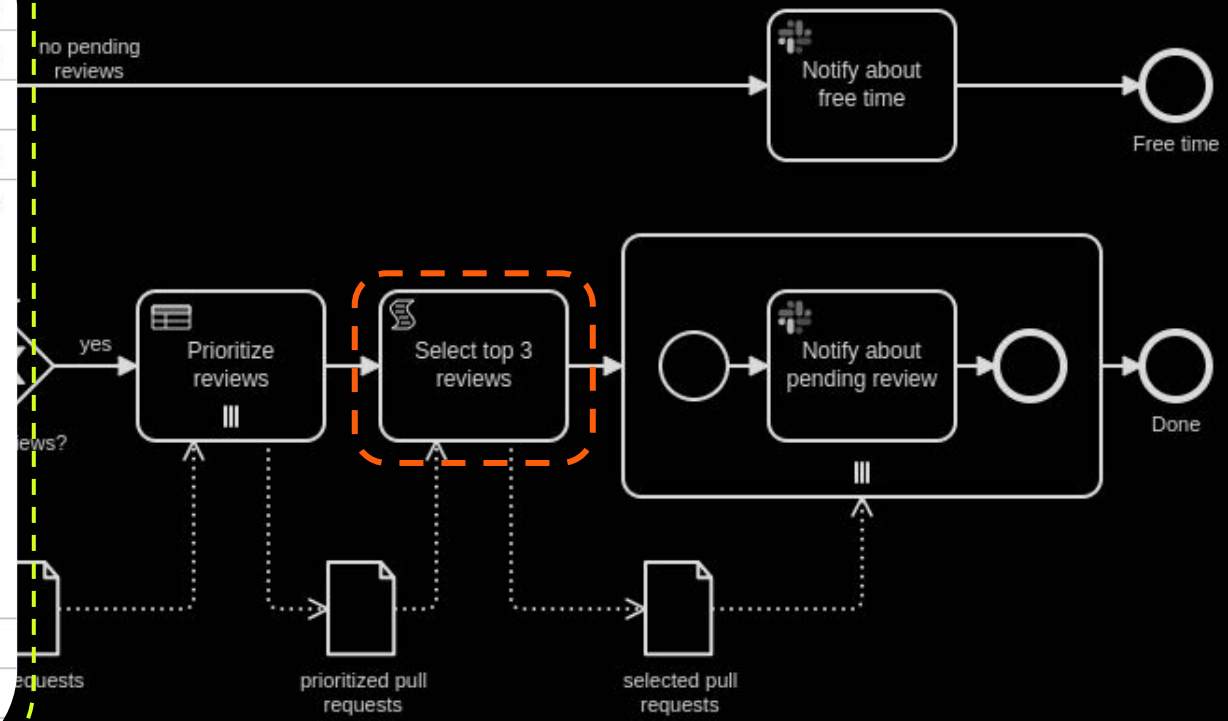
```
= {
    pull_request: pull_request,
    priority: review_priority
  }
```

Completion condition

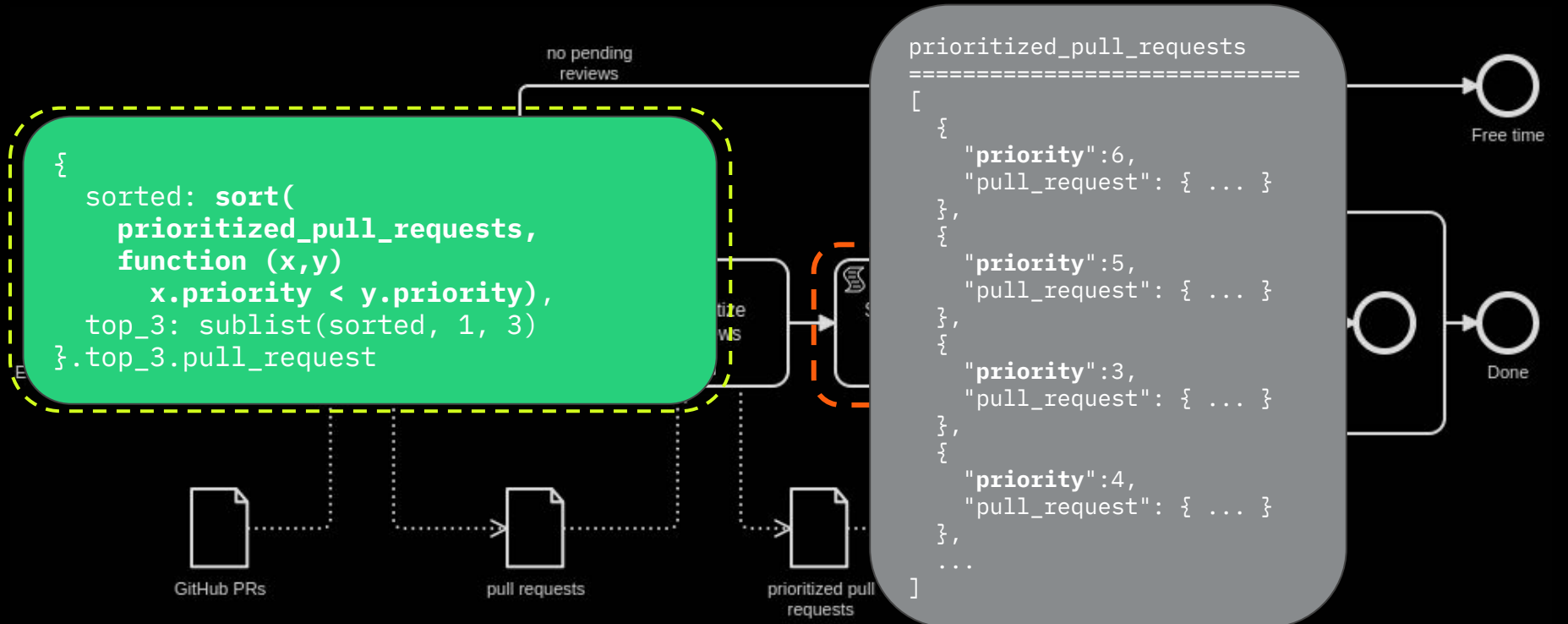
```
=
```

Script tasks

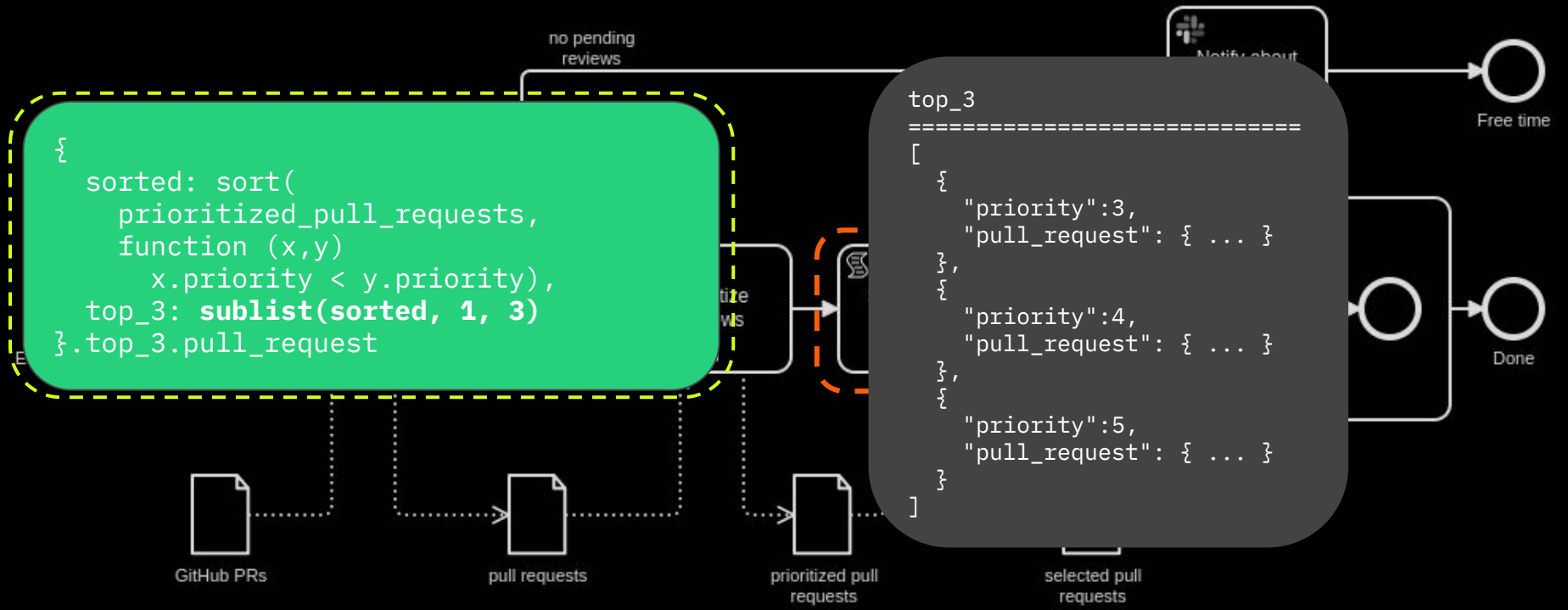




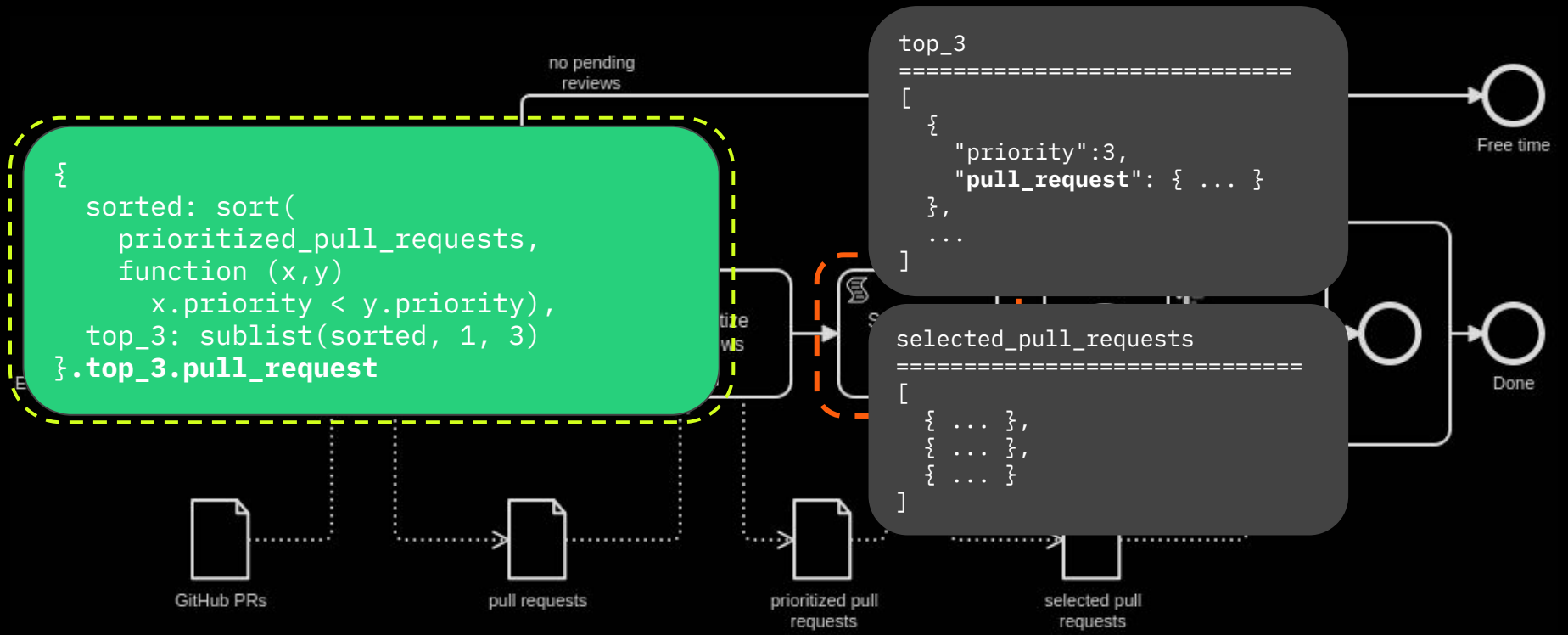
Script tasks



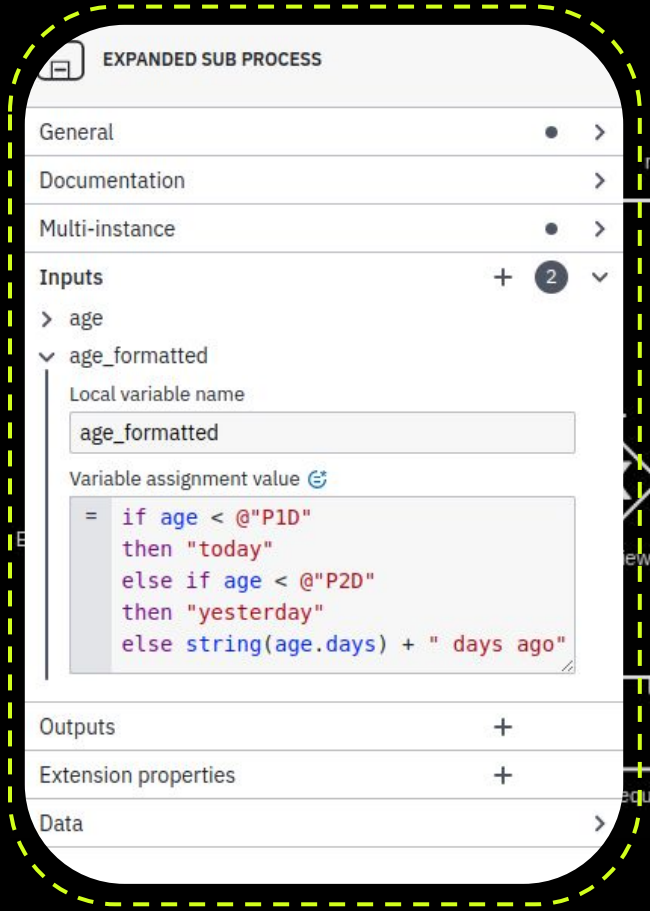
Script tasks

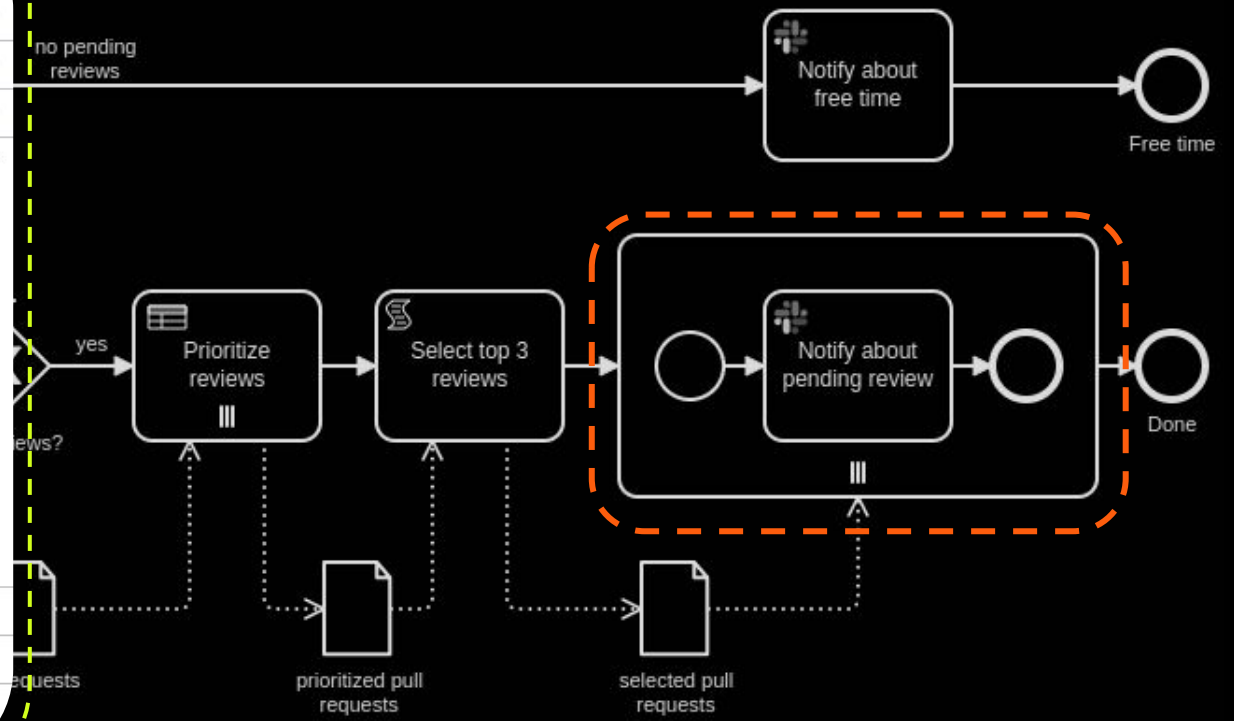


Script tasks



Variable mapping





The final result

CCS 2023

CCS 2023

Tips & Tricks

Comparison

CCS 2023

```
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

CCS 2023

```
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

CCS 2023

Why FEEL?

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

Empower BPMN

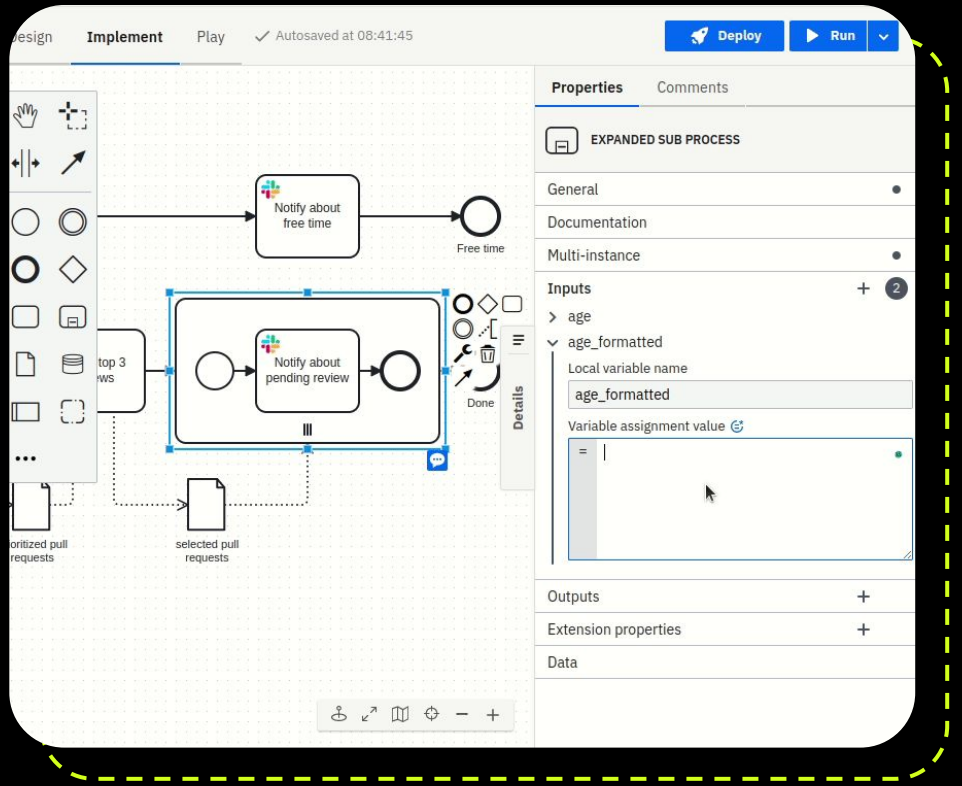
- Gateways
- Connectors
- Script tasks
- Variable mappings
- Time events

Pitfalls

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

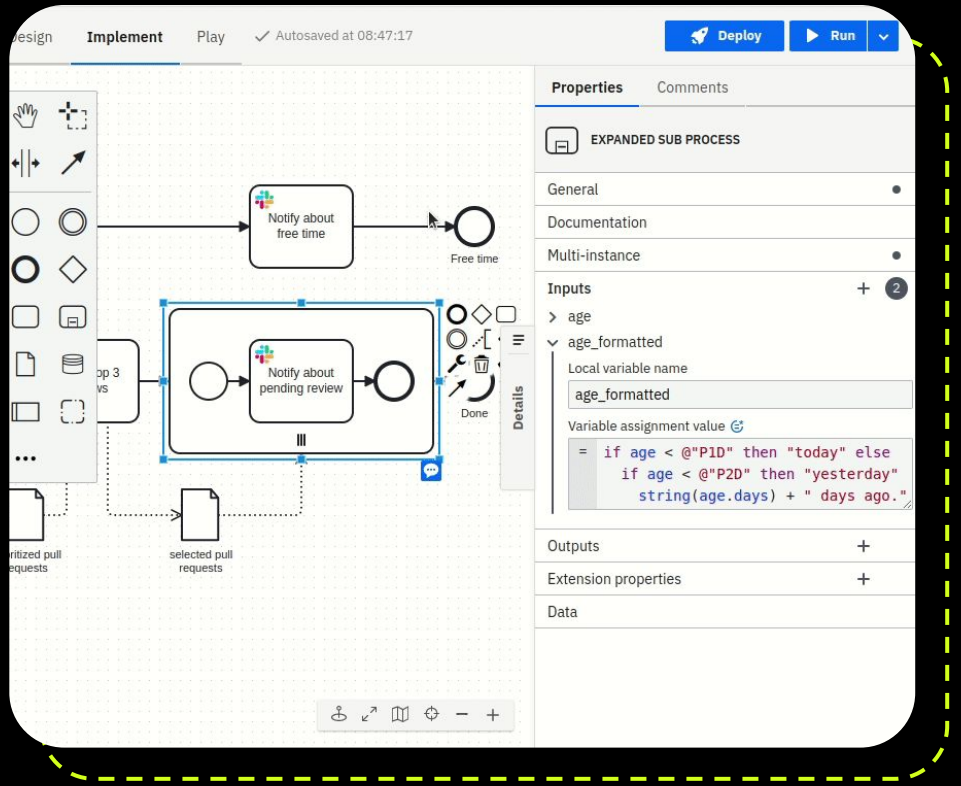
Tooling

- **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



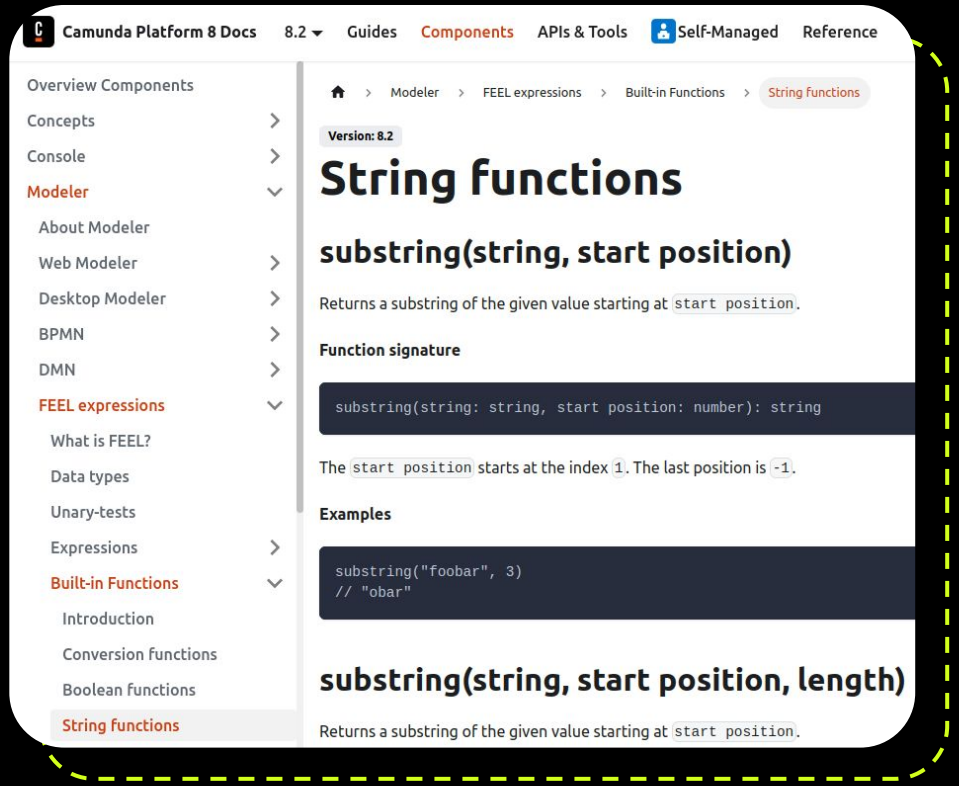
Tooling

- 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



CCS 2023

Questions?

Thank You

https://github.com/saig0

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

https://camunda.com/platform/zeebe