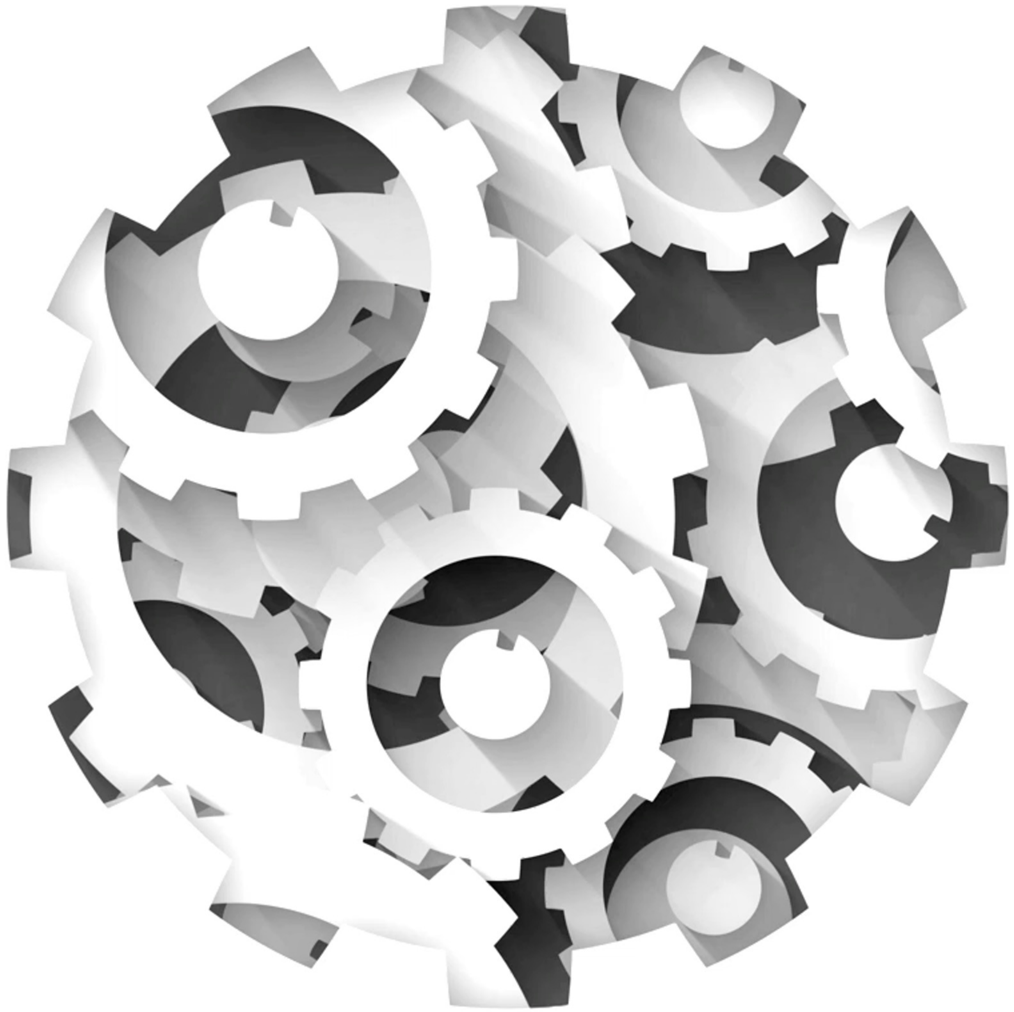
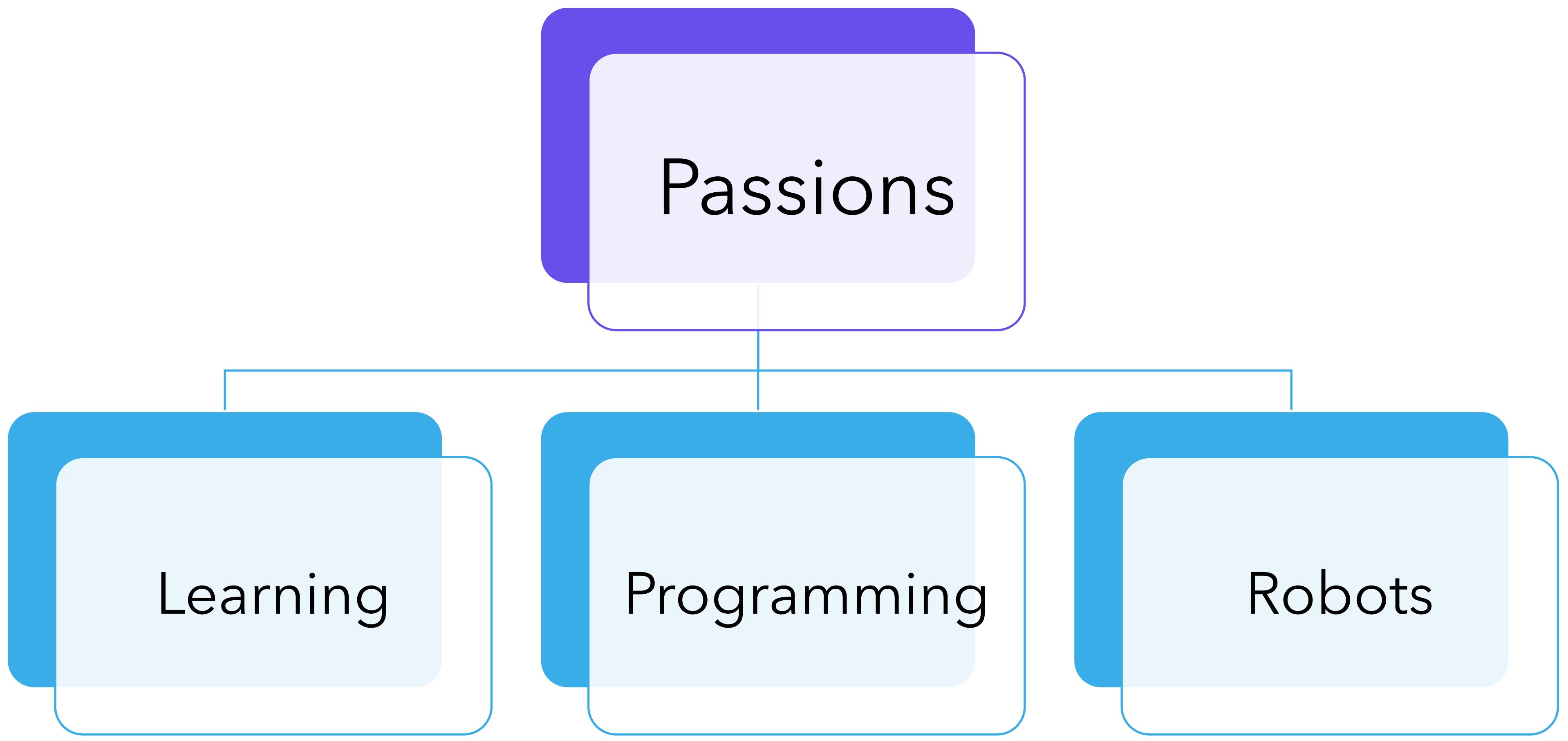
Workforce

Continuous Intelligence



Jeff Joneson



Jeff Joneson

- Experience
  - Joined a hackathon 10 years ago
  - 10-year career in banking technology
  - Quit my job 2 weeks ago to pursue AI full time
  - Found another hackathon

Problem Statement

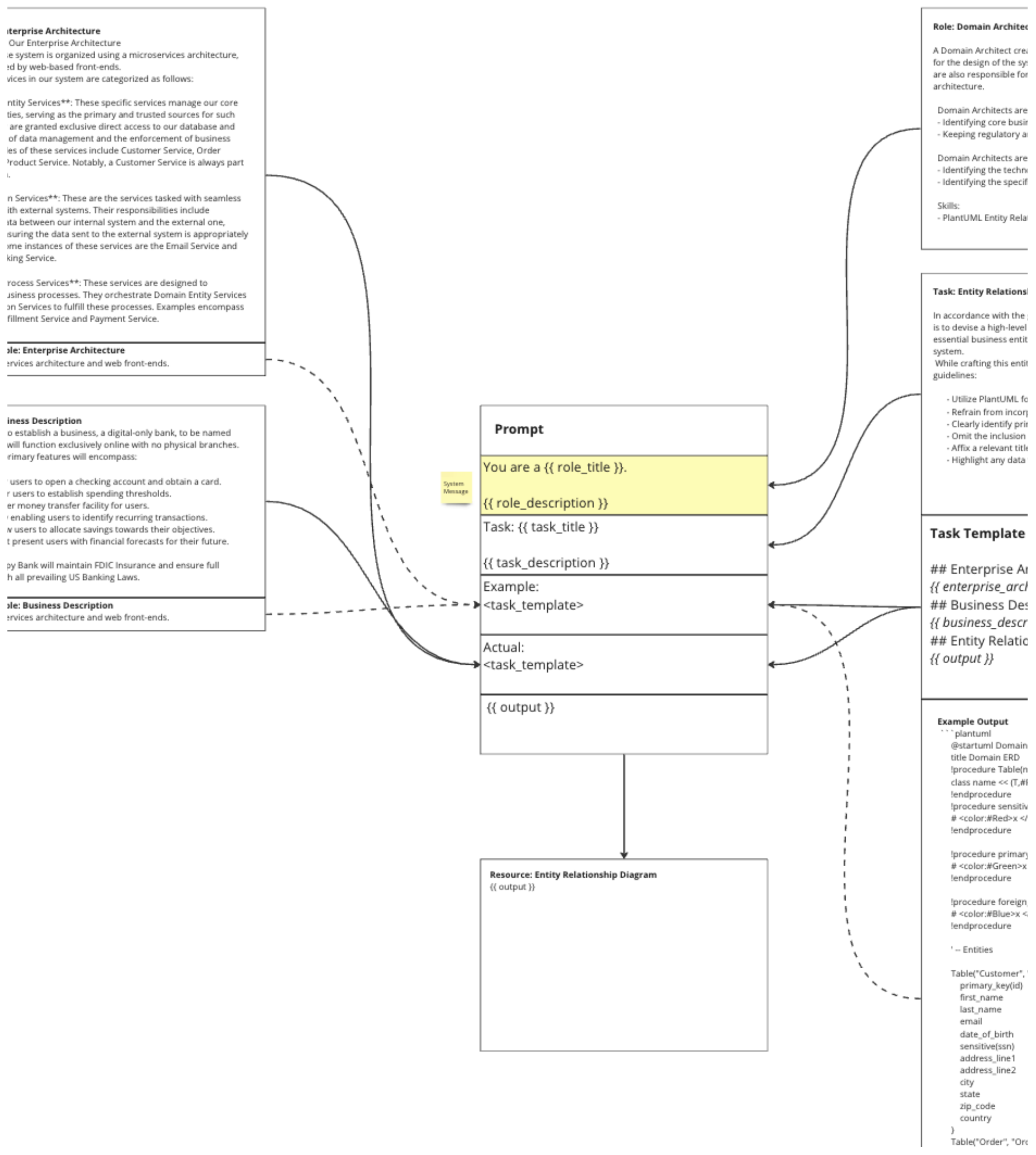
Language Models can plug into almost all natural language interfaces, given sufficient creativity.

However, the following requirements must be met:

- Access to the same information as humans (documentation, examples, etc.)
- They must interact with us using the same tools we use
- They must follow the same processes as humans

Key Challenges

- Programs are deterministic, language models are not
- Given too much autonomy, things can go off the rails quickly
- Coding is not the most accessible medium
- Hard to visualize dynamic prompts



iterprise Architecture
Our Enterprise Architecture
e system is organized using a microservices architecture,
ed by web-based front-ends.
vices in our system are categorized as follows:

ntity Services**: These specific services manage our core
ties, serving as the primary and trusted sources for such
are granted exclusive direct access to our database and
of data management and the enforcement of business
es of these services include Customer Service, Order
Product Service. Notably, a Customer Service is always part
.

n Services**: These are the services tasked with seamless
ith external systems. Their responsibilities include
ta between our internal system and the external one,
suring the data sent to the external system is appropriately
me instances of these services are the Email Service and
king Service.

rocess Services**: These services are designed to
usiness processes. They orchestrate Domain Entity Services
on Services to fulfill these processes. Examples encompass
fillment Service and Payment Service.

ole: Enterprise Architecture
ervices architecture and web front-ends.

iness Description
o establish a business, a digital-only bank, to be named
will function exclusively online with no physical branches.
rimary features will encompass:

users to open a checking account and obtain a card.
r users to establish spending thresholds.
er money transfer facility for users.
enabling users to identify recurring transactions.
v users to allocate savings towards their objectives.
t present users with financial forecasts for their future.

py Bank will maintain FDIC Insurance and ensure full
h all prevailing US Banking Laws.

ole: Business Description
ervices architecture and web front-ends.

Prompt

You are a {{ role_title }}.

{{ role_description }}

Task: {{ task_title }}

{{ task_description }}

Example:
<task_template>

Actual:
<task_template>

{{ output }}

Resource: Entity Relationship Diagram
{{ output }}

Role: Domain Architec

A Domain Architect cre
for the design of the sy
are also responsible for
architecture.

Domain Architects are
- Identifying core busi
- Keeping regulatory a

Domain Architects are
- Identifying the techn
- Identifying the specif

Skills:
- PlantUML Entity Rela

Task: Entity Relations

In accordance with the
is to devise a high-level
essential business entit
system.
While crafting this enti
guidelines:

- Utilize PlantUML fo
- Refrain from incor
- Clearly identify pri
- Omit the inclusion
- Affix a relevant titl
- Highlight any data

Task Template

```
## Enterprise Ar
{{ enterprise_arcl
## Business Des
{{ business_descr
## Entity Relatic
{{ output }}
```

Example Output

~~~
```plantuml
  @startuml Domain
  title Domain ERD
  !procedure Table(n
  class name << (T,#
  !endprocedure
  !procedure sensitiv
  # <color:#Red>x </
  !endprocedure

  !procedure primary
  # <color:#Green>x
  !endprocedure

  !procedure foreign
  # <color:#Blue>x <
  !endprocedure

  ' -- Entities

  Table("Customer", '
    primary_key(id)
    first_name
    last_name
    email
    date_of_birth
    sensitive(ssn)
    address_line1
    address_line2
    city
    state
    zip_code
    country
  }
  Table("Order", "Orc
~~~

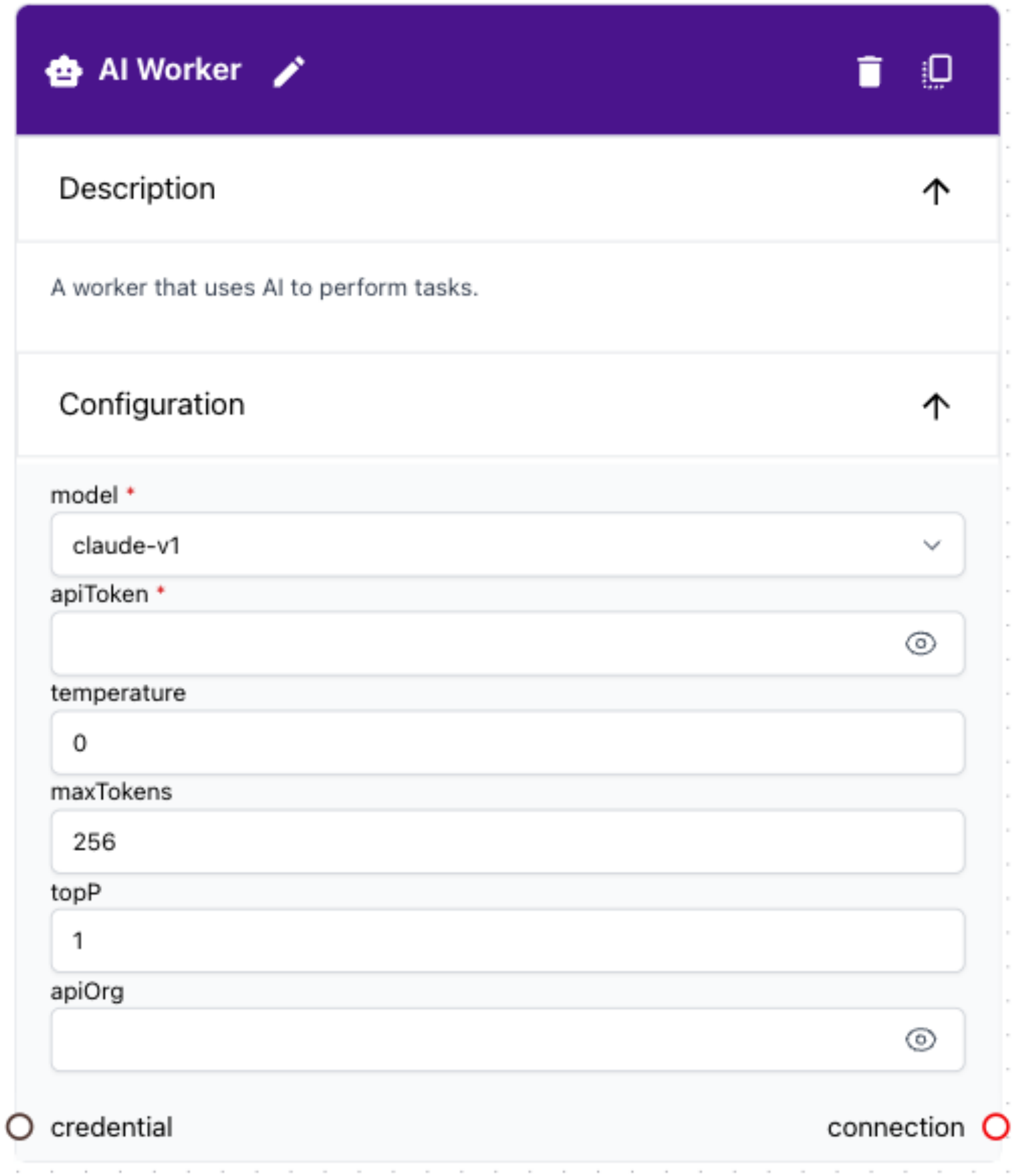
Solution: Workforce

- No-Code UI
- Real-Time, Continuous Integration
- Event-Driven
- Integrate With Existing Tools

Key Features

Workers

Connect to external models


AI Worker

Description

A worker that uses AI to perform tasks.

Configuration

model *
claude-v1

apiToken *

temperature
0

maxTokens
256

topP
1

apiOrg

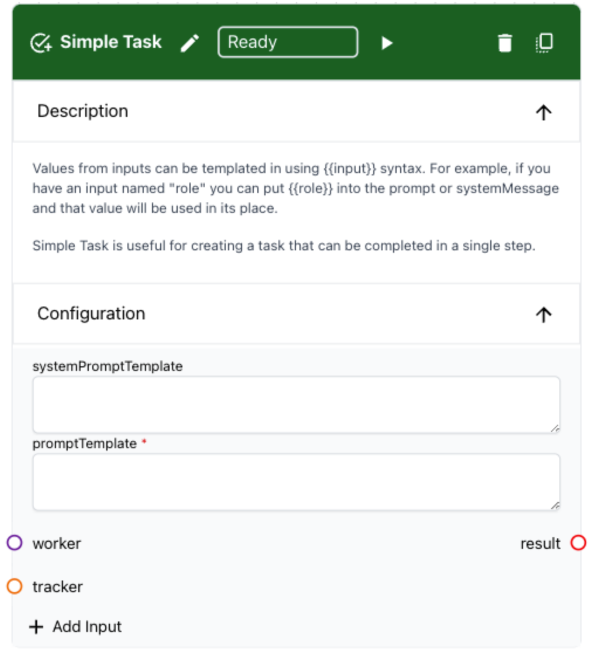
credential

connection

Key Features

Tasks

Create prompt templates



Simple Task Ready

Description

Values from inputs can be templated in using {{input}} syntax. For example, if you have an input named "role" you can put {{role}} into the prompt or systemMessage and that value will be used in its place.

Simple Task is useful for creating a task that can be completed in a single step.

Configuration

systemPromptTemplate

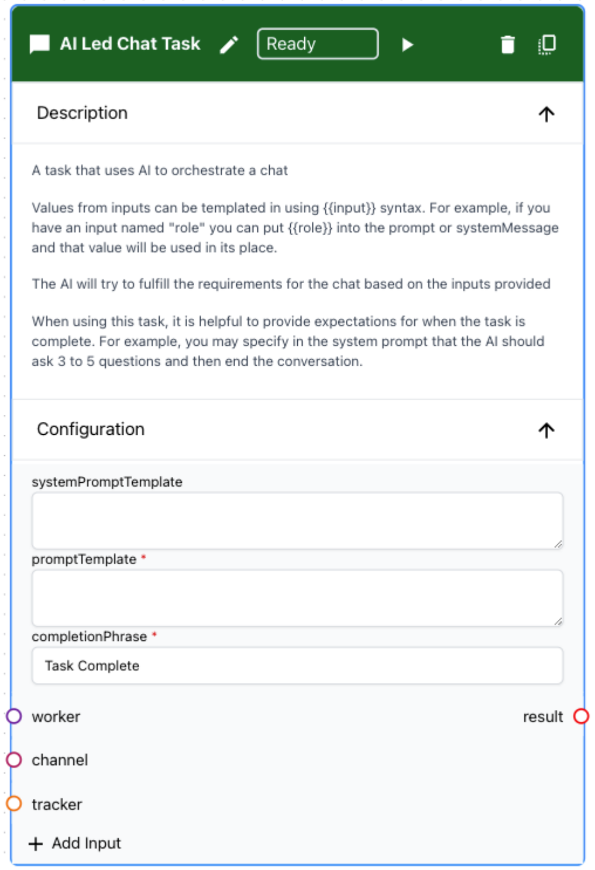
promptTemplate *

worker

result

tracker

+ Add Input



AI Led Chat Task Ready

Description

A task that uses AI to orchestrate a chat

Values from inputs can be templated in using {{input}} syntax. For example, if you have an input named "role" you can put {{role}} into the prompt or systemMessage and that value will be used in its place.

The AI will try to fulfill the requirements for the chat based on the inputs provided

When using this task, it is helpful to provide expectations for when the task is complete. For example, you may specify in the system prompt that the AI should ask 3 to 5 questions and then end the conversation.

Configuration

systemPromptTemplate

promptTemplate *

completionPhrase *

Task Complete

worker

result

channel

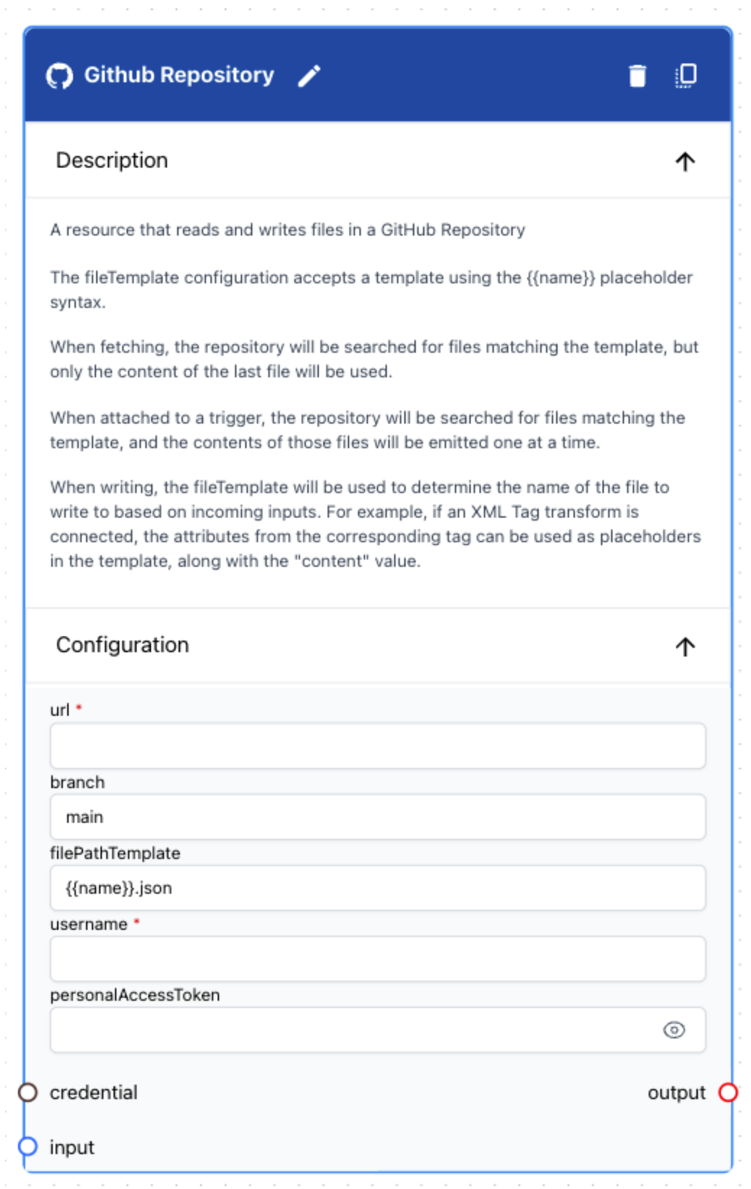
tracker

+ Add Input

Key Features

External Resources

Push and pull data



Description

A resource that reads and writes files in a GitHub Repository

The fileTemplate configuration accepts a template using the {{name}} placeholder syntax.

When fetching, the repository will be searched for files matching the template, but only the content of the last file will be used.

When attached to a trigger, the repository will be searched for files matching the template, and the contents of those files will be emitted one at a time.

When writing, the fileTemplate will be used to determine the name of the file to write to based on incoming inputs. For example, if an XML Tag transform is connected, the attributes from the corresponding tag can be used as placeholders in the template, along with the "content" value.

Configuration

url *

branch: main

filePathTemplate: {{name}}.json

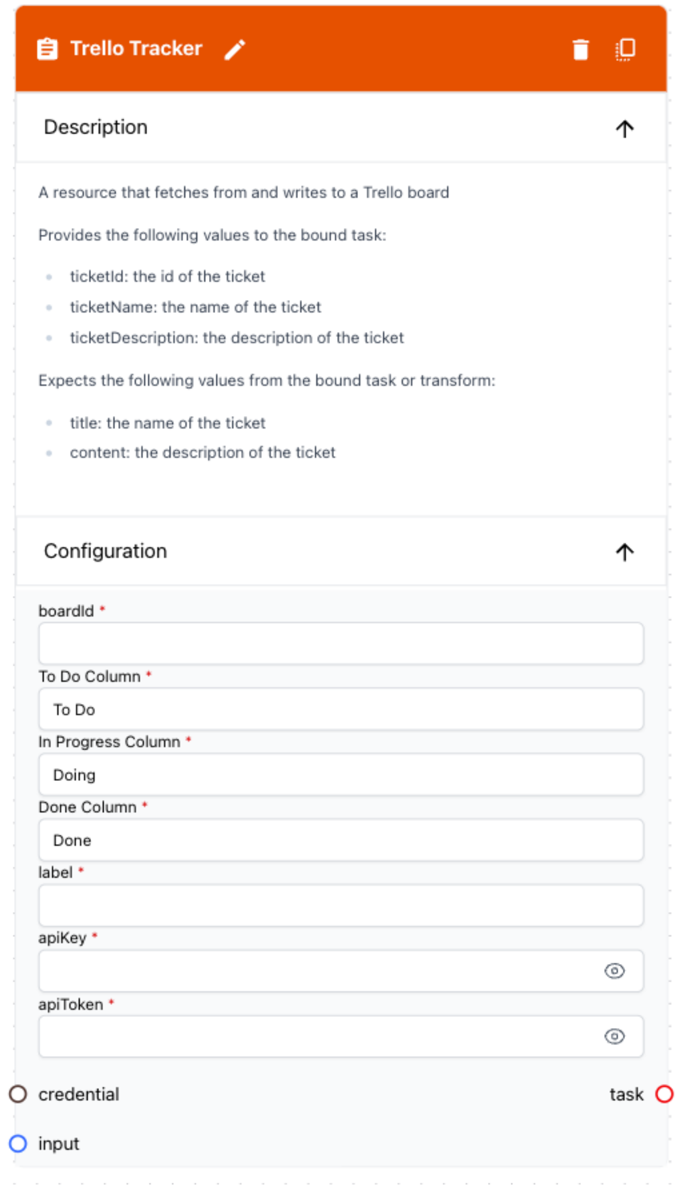
username *

personalAccessToken

Key Features

Trackers

Pull tickets and create new ones



Description

A resource that fetches from and writes to a Trello board

Provides the following values to the bound task:

- ticketId: the id of the ticket
- ticketName: the name of the ticket
- ticketDescription: the description of the ticket

Expects the following values from the bound task or transform:

- title: the name of the ticket
- content: the description of the ticket

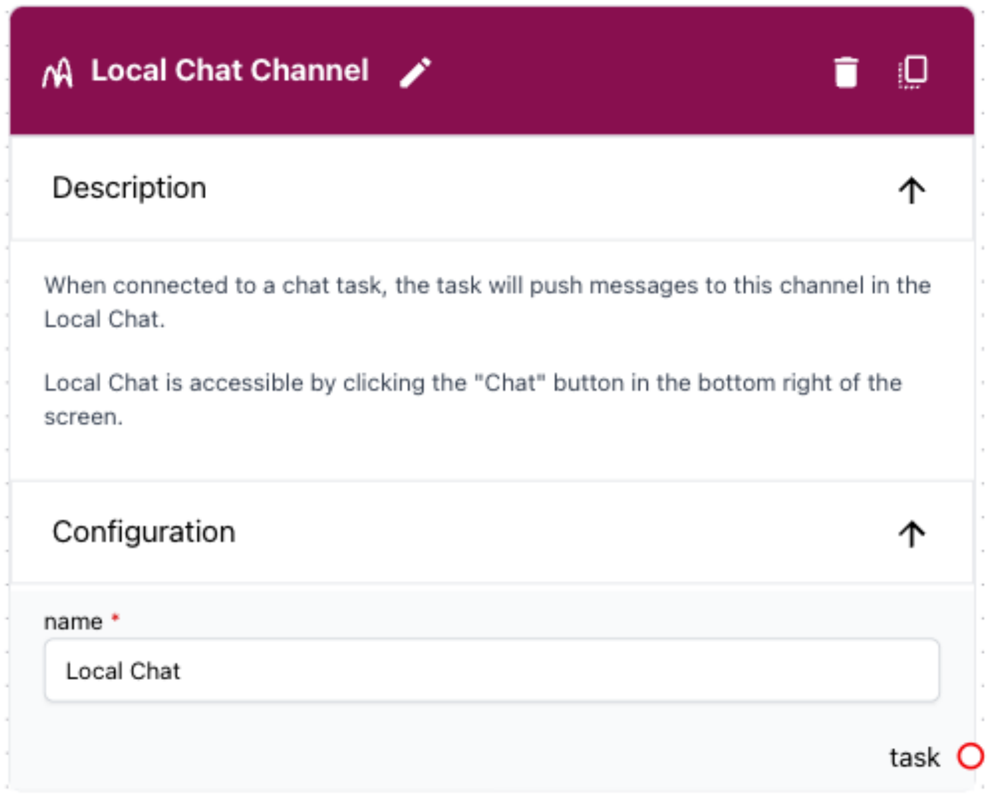
Configuration

boardId *	
To Do Column *	To Do
In Progress Column *	Doing
Done Column *	Done
label *	
apiKey *	
apiToken *	

Key Features

Channels

Connect to humans for conversations


Local Chat Channel

Description

When connected to a chat task, the task will push messages to this channel in the Local Chat.

Local Chat is accessible by clicking the "Chat" button in the bottom right of the screen.

Configuration

name *

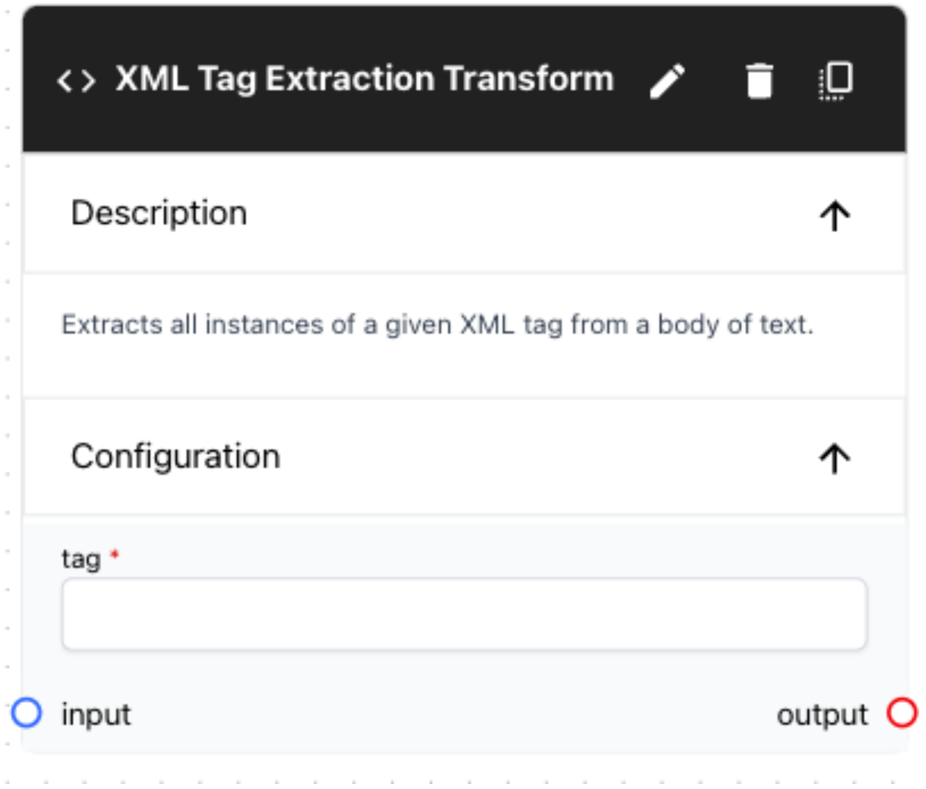
Local Chat

task

Key Features

Transforms

Extract data from outputs

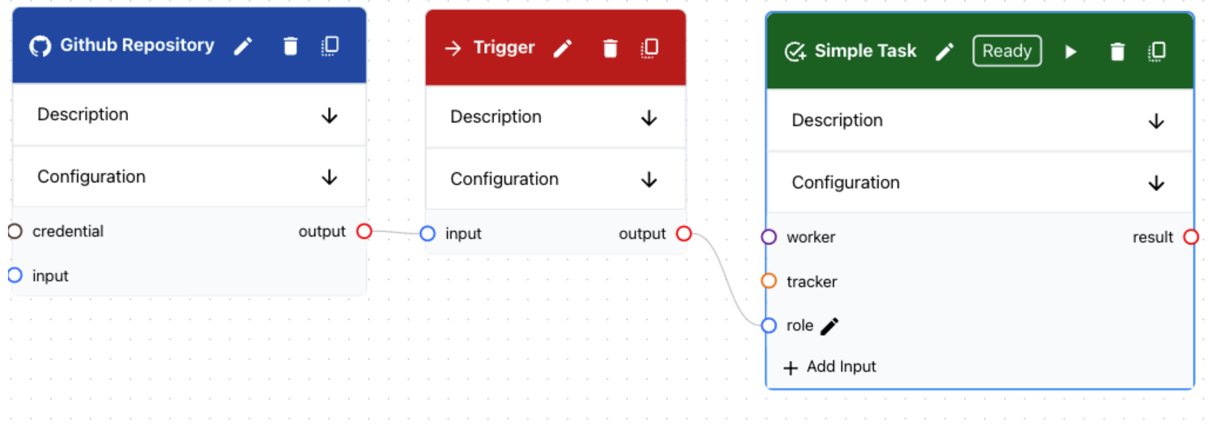


Extracts all instances of a given XML tag from a body of text.

Key Features

Triggers

Trigger tasks when external resources change



Technologies

- Github Copilot
- React
- React-Flow
- Rxjs
- Zustand State Management
- Express.js

Business Model

- Free, bring-your-own-key web-only version to drive excitement
- Monthly subscription for premium features and server-side scheduling
- Enterprise deals for self-hosted or managed-hosting
- Professional services to automate existing workflows

Demo