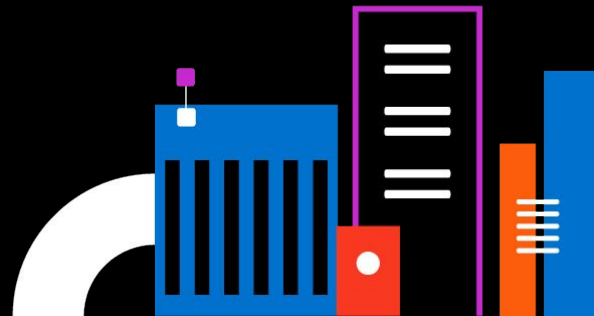


# Taking Templates to the Limit

**Create Bulletproof Camunda  
Element Templates with JSON  
Schema**

Niklas Kiefer  
Camunda



Camunda Modeler

diagram\_1.bpmn

POST PIZZA ORDER  
order pizza

General

Template Applied

Request payload

Pizza  
= pizza

Amount  
1

Payment method  
Visa

Confirmation email

Invalid email address  
[How to configure an email address](#)

XML Camunda Platform 8.0 0 errors Log 5.0.0

# About me



Software Engineer at Camunda, bpmn.io

Part of the team who delivers the web-based modeling BPMN, DMN, and Forms

GitHub: [@pinussilvestrus](https://github.com/pinussilvestrus)

Twitter: [lutki95](https://twitter.com/lutki95)



# What's in today



Element Templates

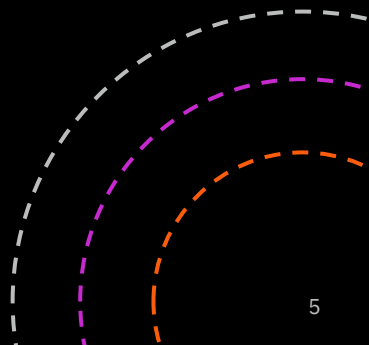
JSON Schema

Element Templates features you may not know

Coding Element Templates in your favorite IDE

How to model a pizza delivery with Element Templates

# Element Templates



# High Level Overview

reusable JSON files

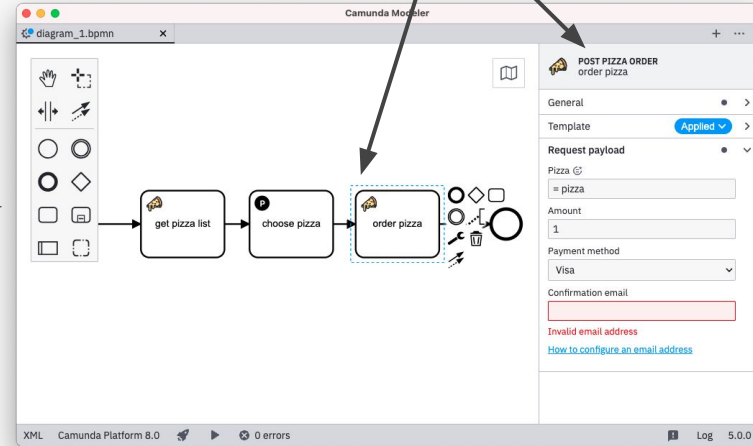
```
{
  "name": "GET Pizza list",
  "id": "com.example.orderPizza",
  "description": "Connect to ...",
  "appliesTo": [
    "bpmn:ServiceTask",
    "bpmn:Task"
  ],
  "properties": [
    ...
  ]
}
```

defines and  
validates

JSON Schema

load on start

domain-specific, user  
friendly interface and  
appearance



supported in Camunda  
Platform 7 and 8

# Check out the docs



Camunda Platform 8 Docs latest Guides Components APIs & Clients Self-Managed Reference

Overview Components

Concepts

Cloud Console

**Modeler**

About Modeler

Web Modeler

**Desktop Modeler**

Install the Modeler

Version: latest

## Defining templates

Templates are defined in template descriptor files as a JSON array:

```
[  
  {  
    "$schema": "https://unpkg.com/camunda-platform-8-templates/schema/1.0.0/schema.json",  
    "name": "Template 1",  
    "id": "some-descriptor-id"  
  }  
]
```

Copy

JSON Schema Compatibility

Supported BPMN Types

Defining Template Properties

Camunda Platform 8 Docs latest Guides Components APIs & Clients Self-Managed Reference

Overview Components

Concepts

Cloud Console

**Modeler**

About Modeler

Web Modeler

**Desktop Modeler**

Install the Modeler

Model your first diagram

Connect to Camunda Platform 8

Start a new process instance

**Element templates**

About element templates

Configuring templates

**Using templates**

Defining templates

Additional resources

## Updating Templates

If a template is applied and a new version of the template is found you can update the template.

Applying Templates

Removing Templates

**Updating Templates**

Missing Templates

Camunda Modeler

diagram\_1.bpmn

SERVICE TASK

General

Template

Name: A new version of the template is available (22-01-2020)

Version: 01.01.2020

Description: A new version of the template is available (22-01-2020)

Custom properties

Camunda Platform 8 Docs latest Guides Components APIs & Clients Self-Managed Reference

Overview Components

Concepts

Cloud Console

**Modeler**

About Modeler

Web Modeler

**Desktop Modeler**

Install the Modeler

Model your first diagram

Connect to Camunda Platform 8

Start a new process instance

**Element templates**

About element templates

Configuring templates

Using templates

**Defining templates**

Additional resources

Notice that adherence to the following configuration options is enforced by design. If not adhering, it logs a validation error and ignores the respective element template.

Bindings for Camunda Platform 7 and 8

Bindings for Camunda Platform 7

**Bindings for Camunda Platform 8**

JSON Schema Compatibility

Supported BPMN Types

Defining Template Properties

zeebe:input

Binding type	zeebe:input
Valid property type's	String Text Hidden Dropdown
Binding parameters	name: the name of the input parameter
Mapping result	<zeebe:input target="[name]" source="[userInput]" />

zeebe:output

Binding type	zeebe:output
--------------	--------------

# Examples



Send an Email

Call a UiPath bot

Get data from an Excel sheet

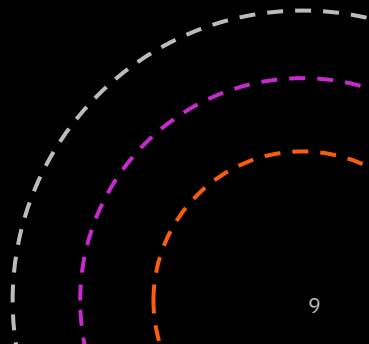
Connect to a REST API

....

Find some samples [on GitHub!](#)



# JSON Schema



# JSON Schema



Vocabulary that allows you to annotate and validate JSON documents

Defining the structure of JSON data

It makes JSON documents robust on the application level

Check it out: <https://json-schema.org/>

# High Level Overview

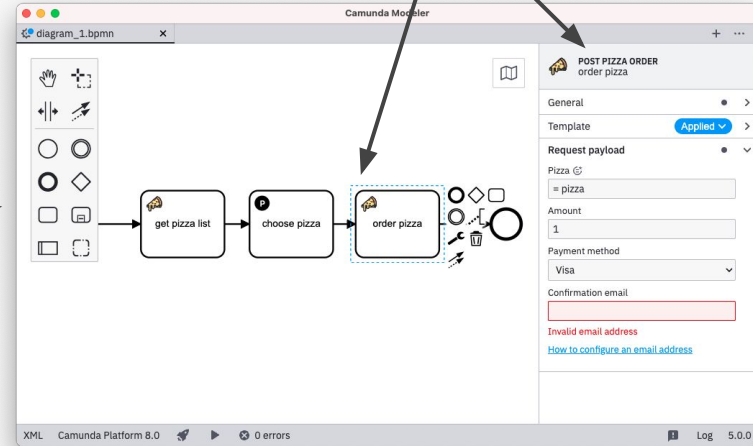
reusable JSON files

```
{
  "name": "GET Pizza list",
  "id": "com.example.orderPizza",
  "description": "Connect to ...",
  "appliesTo": [
    "bpmn:ServiceTask",
    "bpmn:Task"
  ],
  "properties": [
    ...
  ]
}
```

defines and  
validates

JSON Schema

load on start



domain-specific, user  
friendly interface and  
appearance

supported in Camunda  
Platform 7 and 8

# Element Templates JSON Schema



Check it out: <https://github.com/camunda/element-templates-json-schema>

The screenshot shows the GitHub repository page for `camunda/element-templates-json-schema`. The repository is public and has 4 watchers, 1 fork, and 6 stars. The main content area displays a list of files and folders, including `.github`, `example`, `packages`, `tasks`, `.eslintrc`, `README.md`, `lerna.json`, `package-lock.json`, and `package.json`. The `README.md` file is selected, showing the title `element-templates-json-schema` and a status bar indicating `build passing`. The README text states: "JSON Schema for (Camunda) Element Templates. The schema is built on top of and validated by `json-schema@draft-07`."

Repository: `camunda/element-templates-json-schema` (Public)

Actions: `<> Code`, `Issues 5`, `Pull requests`, `Actions`, `Security`, `Insights`

Branches: `master` (3 branches), Tags: 13 tags

Files and Folders:

File/Folder	Description	Last Commit
<code>.github</code>	chore: add issue templates	12 months ago
<code>example</code>	chore(example): update example	last month
<code>packages</code>	feat(zeebe): add documentationRef	2 days ago
<code>tasks</code>	chore: split up schema bundling	last month
<code>.eslintrc</code>	chore: do not lint dist folders	17 months ago
<code>.eslintrc</code>	feat(project): initial implementation	17 months ago
<code>.gitignore</code>	feat: remove validator parts	17 months ago
<code>README.md</code>	feat(project): introduce monorepo	last month
<code>lerna.json</code>	chore(lerna): use independent versioning	last month
<code>package-lock.json</code>	feat(project): introduce monorepo	last month
<code>package.json</code>	chore(package): add publish scripts	last month

About: JSON Schema for (Camunda) Element Templates

Contributors: 5

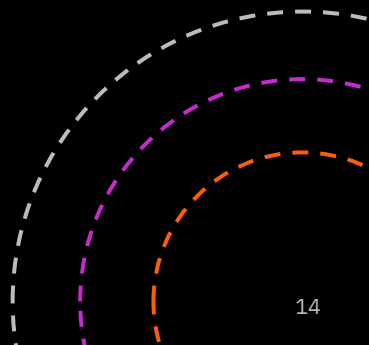
Languages: JavaScript 100.0%

# Element Templates JSON Schema



```
{  
  // use latest (Camunda Platform 7)  
  "$schema": "https://unpkg.com/@camunda/element-templates-json-schema/resources/schema.json",  
  
  // use specific version  
  "$schema": "https://unpkg.com/@camunda/element-templates-json-schema@0.8.0/resources/schema.json",  
  
  // use latest (Camunda Platform 8)  
  "$schema": "https://unpkg.com/@camunda/zeebe-element-templates-json-schema/resources/schema.json",  
  
  // use specific version  
  "$schema": "https://unpkg.com/@camunda/zeebe-element-templates-json-schema@0.3.0/resources/schema.json",  
  
  "id": "com.camunda.example.MailTask"  
  ...  
}
```

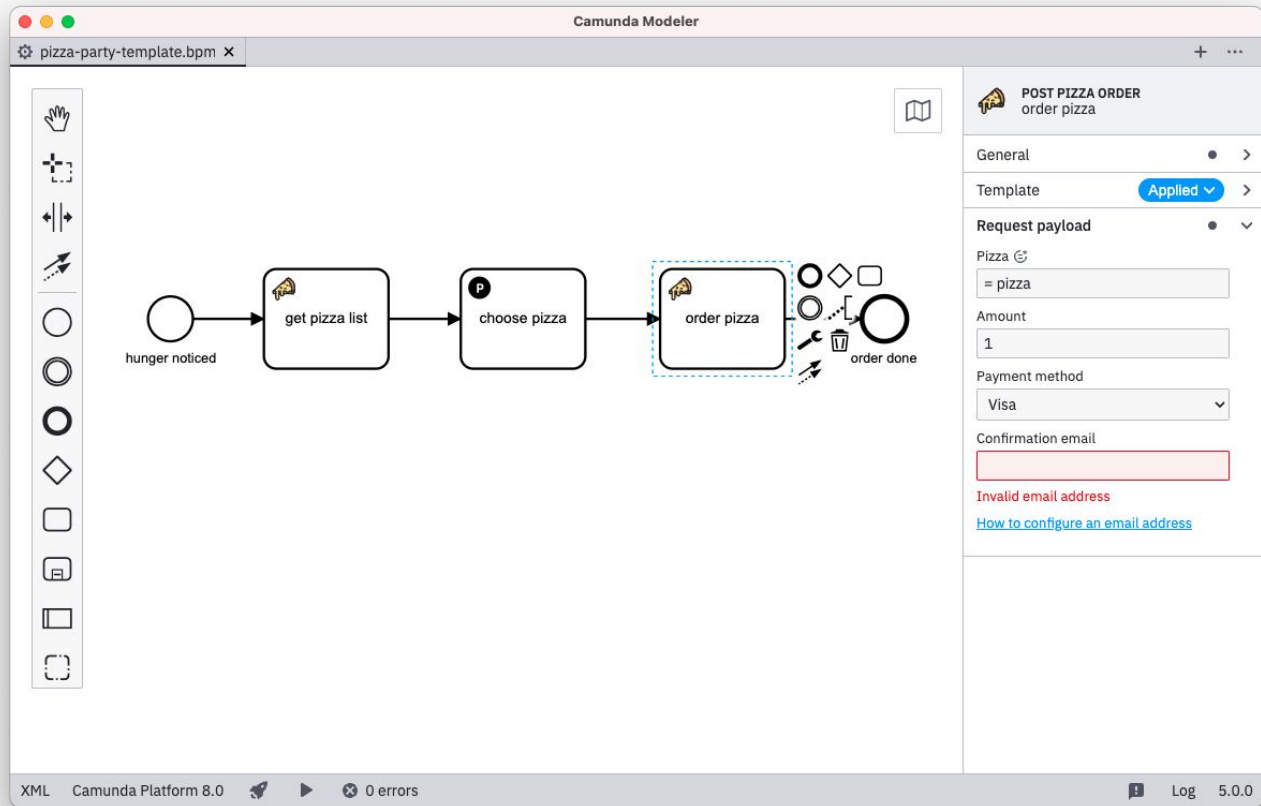
# Let's Code Some Element Templates!



# Hungry for pizza

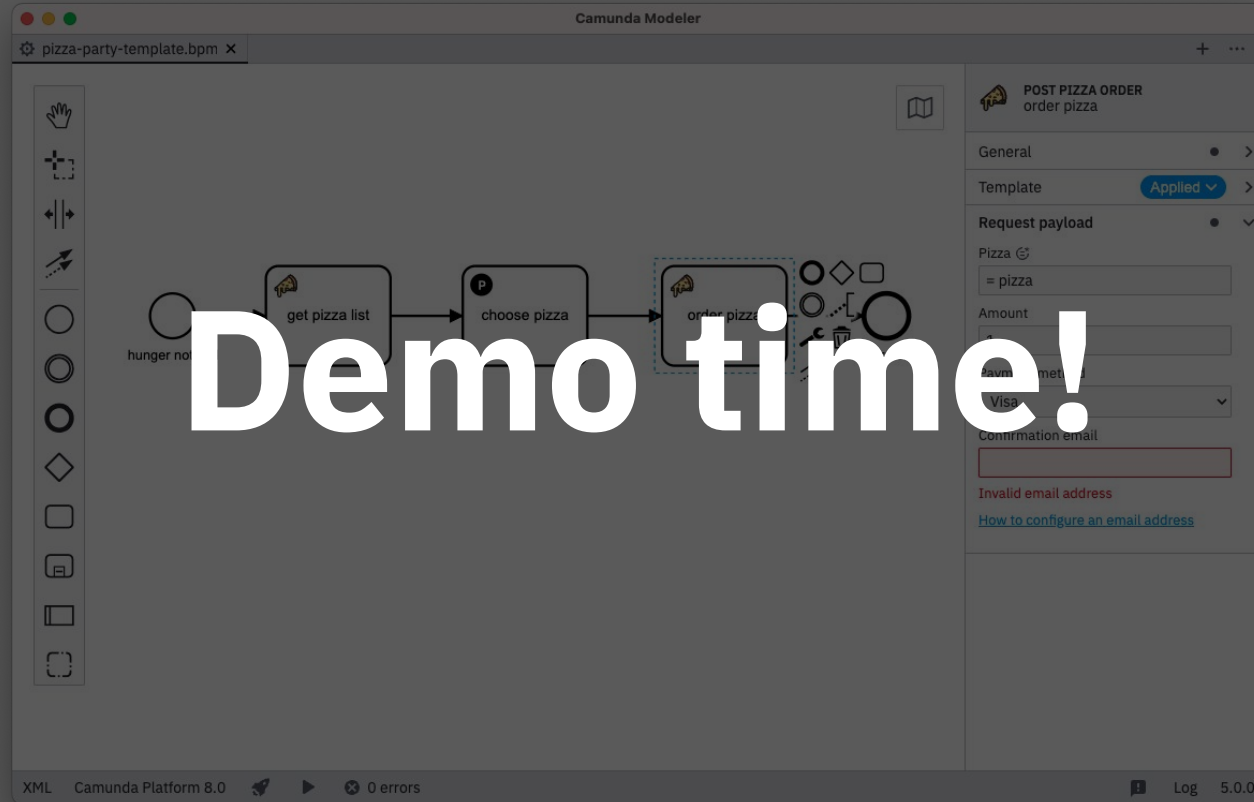


# Hungry for pizza (with templates)





# Hungry for pizza (with templates)



Demo time!

# Resources



[Element Templates Documentation](#)

[Element Templates JSON Schema](#)

[Element Templates Samples](#)

[Report Issue](#)

[Give Feedback](#)

# Resources



[Element Templates Documentation](#)

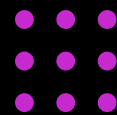
[Element Templates JSON Schema](#)

[Element Templates Samples](#)

[Report Issue](#)

[Give Feedback](#)

**You are welcome  
to contribute!**



**Thank you!**  
**Any questions?**

