

Beeker for Teenagers

LA TEMPLATISATION PAR L'IA



SOMMAIRE

Introduction	3
Installation de Beeker	4
Les actions	5
Les règles	6
Exécution d'une action	7
Exemple Terraform	8
GUI Beeker	11
Les Bees	12
GUI - Les jobs	13
Exemple Page Web	14

Introduction



Beeker est un outil avancé de génération de code source, utilisant des fichiers YAML et des prompts pour automatiser et exécuter des tâches de programmation. Il permet aux développeurs de combiner des directives précises (prompts) avec des configurations structurées (YAML) pour générer du code efficacement.

L'outil est particulièrement utile pour automatiser la rédaction de code répétitif, intégrer de nouvelles fonctionnalités rapidement et maintenir des normes de codage cohérentes à travers des projets. Il est applicable dans de nombreux domaines où la rapidité et la précision du développement logiciel sont critiques.

Installation de Beeker



Beeker fonctionne sur Node.js, donc il est essentiel d'installer Node.js sur votre système. Vous aurez également besoin de créer un compte Beeker pour traiter les requêtes d'IA :

<https://www.beeker.tech/>

```
● ● ●

# Installation globale de Beeker avec npm
# npm install -g @beeker-tech/cli
npm install -g @beeker-tech/cli

# Installation globale de Beeker avec yarn
# yarn global add @beeker-tech/cli
yarn global add @beeker-tech/cli

# Installation globale de Beeker avec pnpm
# pnpm add -g @beeker-tech/cli
pnpm add -g @beeker-tech/cli
```

Les actions



Les actions dans Beeker sont des fichiers YAML avec l'extension '.action.yml' qui contiennent des règles, des variables et de la configuration. Les règles peuvent être multiples et on pour but d'améliorer ou de générer des fichiers définis par la variable 'target'.

```
● ● ● update-page-title.action.yml

name: update-page-title
label: Example - Update page title
description: Allow to update example page title
group: Example

parallelize: true

variables:
  - name: title
    label: Title

rules:
  - name: update-page-title
    target: ./page.tsx
```

D'autres paramètres peuvent être ajoutés tel que le contexte (à l'aide de fichiers supplémentaires) afin d'augmenter les connaissances de ChatGPT et d'améliorer son résultat final.

Les règles



Les règles sont définies dans un fichier '`.beek`' qui contiennent l'instruction que Beeker enverra à ChatGPT. L'instruction peut alors être personnalisée avec les variables se trouvant dans le fichier d'action.



update-page-title.beek

The language is typescript and the frameworks are react.js and next.js.

Just update h1 tag with text content "`{{title}}`".

Plus les règles sont précises et meilleur sera le résultat.

Comme il s'agit d'un prompt, il est même possible de lui donner du code en exemple sans problème.

Exécution d'une action



L'exécution de faire passer l'action et ses règles dans un algorithme pour que Beeker puisse l'interpréter finement et l'envoyer à ChatGPT.

Avant de réaliser la modification ou la création des fichiers, Beeker dispose d'un moyen pour prévisualiser le résultat.

```
● ● ●

# Permet de prévisualiser le résultat de l'action
# -p : Permet de donner le chemin du fichier action
beeker action preview -p ./app/update-page-title.action.yml

# Exécution de l'action
beeker action run -p ./app/update-page-title.action.yml
# L'ajout du paramètre -t [Path] permet d'exécuter l'action
depuis un autre dossier

# Liste les actions présents dans le dossier et permet de
prévisualiser/exécuter l'une d'elle au choix
beeker action preview
beeker action run
```

Exemple - Terraform



Maintenant que l'on comprend comment fonctionne Beeker, il est possible de lui assigner des demandes spécifiques à chaque règle.

Imaginons alors une action capable de créer des modules Terraform à la volée.

Comme variable on peut alors s'amuser en lui demandant de générer un module à partir du template de trois fichiers et d'un prompt.

```
● ● ● module.action.yaml

name: module-terraform
label: 'Create a terraform module'
description: 'Make a terraform module'
group: 'Add'

variables:
  - name: prompt
    label: Prompt
    default_value: 'Create a terraform module'

rules:
  - name: create-vars
    target: ./vars.tf
  - name: create-output
    target: ./output.tf
  - name: create-main
    target: ./main.tf
```

Exemple - Terraform



Notre Action va alors avoir besoin de trois fichiers beek comprenant les règles que l'on va lui attribuer.



create-main.beek

The language is tf.

`{{prompt}}`

But This file is main.tf, it file must contain only main, no variables, no output for the module.

Each resources must have "example" as resource identifier.

Each name of the resource must have "name" as var + type like this (for the resource group example)

`name = "${var.name}-rg"`



create-output.beek

The language is tf.

`{{prompt}}`

But This file is output.tf, it must contain only output, no resources, no variables for the module.

Each resources must have "example" as resource identifier.

Exemple - Terraform



create-vars.beek

The language is tf.

```
{{prompt}}
```

But This file is vars.tf, it must contain only vars, no resources, no output for the module.

You must have a variable name because we will use it for the main like this (for the resource group example)
name = "\${var.name}-rg"

Mon action peut alors se trouver n'importe où sur mon poste, il me suffira simplement de créer un dossier ou la stocker avant.



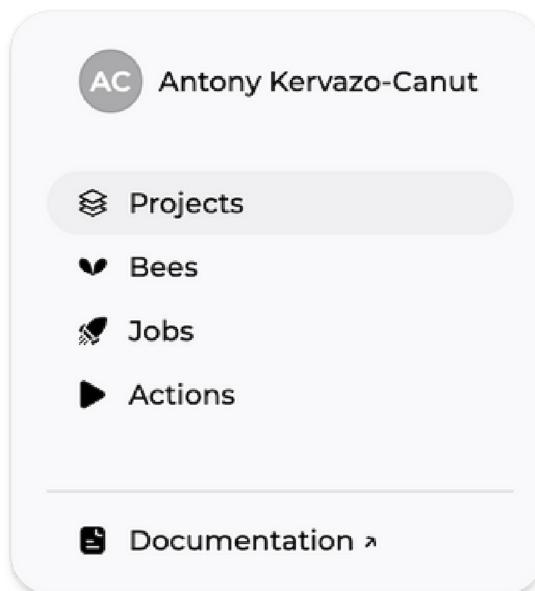
```
mkdir azure_linux_vm_module
cd azure_linux_vm_module
beeker action run -p ~/Actions/Terraform-
Module/module.action.yml -t .

# Une demande d'input me permettra d'écrire dans une langue
comprise par ChatGPT (le français fonctionne donc)
# $ J'ai besoin d'un module pour créer une machine virtuelle
linux sur Azure
# Beeker va alors générer les fichier de mon nouveau module
Terraform
```

GUI Beeker



Beeker fonctionne également avec une interface web. Celle-ci permet alors de pouvoir utiliser les Actions directement dans un environnement conviviale aux utilisateurs moins à l'aise avec des lignes de commandes.



Pour connecter ses projets, celui-ci devra cependant être hébergé sur Github ou Gitlab.

Les Bees



Les Bees sont propre à l'utilisation d'un projet via la GUI. L'exécution d'une action va pousser son résultat sur une nouvelle branche de votre projet.

Sur un projet où travaillent plusieurs personnes, cela permet de mieux compartimenter le travail de chacun.

Bees

ⓘ Bees are a powerful way to handle parallel tasks within your project. Each bee operates in an isolated workspace, ensuring that actions run independently. They will commit and push your work to a specific branch, following the workflow defined in your configuration. If you are not working in a team, you can simply use the default configuration.

Name	Created by
Runner 1	U
Default	U

Your branches

Branch
feature/default/job-3

GUI - Les jobs



Les jobs à travers la GUI fonctionnent en utilisant une Bee (la branche de travail), le projet et une ou plusieurs actions.

Cependant les actions devront être configurés pour placer le résultat des fichiers à l'endroit désiré.

L'avantage de la GUI est également de pouvoir chainer les actions pour provoquer les résultats attendus.

Run Job

Bee

Project

✓ Project repository is up to date

Step 1.

Action	Variables
Create a terraform module <input type="button" value="module-terraform"/>	Prompt * <input type="button" value="Create a terraform mo"/>
Make a terraform module	

Une fois les steps configurées, le run va permettre de lancer les actions qui si succès vont alors commit le résultat directement sur le projet.

Exemple - Page Web



Dans ce nouvel exemple, on va demander à Beeker de nous créer une page web en .cshtml qui utilise en css comme contexte ainsi que des règles définis par notre designer sous la forme d'un fichier exemple.

```
● ● ●

name: cshtml-generator
label: 'Create a cshtml page'
description: 'Make a cshtml page'
group: 'Add'

variables:
- name: prompt
  label: Prompt
  default_value: 'Create a web page'

rules:
- name: create-page
  target: ./Register.cshtml
  context:
    - .. / .. /Stamping/wwwroot/css/website.css
    - .. / .. /Stamping/wwwroot/css/website-computers.css
    - .. / .. /Stamping/wwwroot/css/website-mobile.css
    - .. / .. /Stamping/wwwroot/css/texts.css
    - .. / .. /Stamping/wwwroot/css/margin.css
    - .. / .. /Stamping/wwwroot/css/elements.css
    - .. / .. /Stamping/wwwroot/css/elements.css
    - .. / .. /Stamping/wwwroot/css/colors.css
    - .. / .. /Stamping/Views/Account/Login.cshtml
```

Exemple - Page Web



The language is cshtml.

`{{prompt}}`

Use the login page as example for the design and use css.

L'action est alors très simple et encore une fois, pour rendre l'action modifiable, on lui donne un prompt en paramètre. Puis on exécute l'action avec notre commande en CLI.

```
# On exécute l'action depuis le dossier on l'on souhaite
# créer le fichier.
# -r permet de donner les contextes par rapport à notre
# placement.
beeker action run -p ../../Actions/cshtml-
generator/page.action.yml -t . -r .

# On lui donne le prompt suivant en paramètre (la langue
# importe peu) :
# J'ai besoin d'une page d'inscription avec comme champ
# obligatoire adresse mail et mot de passe (minimum de 8
# caractères + chiffre et lettres). Il faut 2 champs de mot de
# passe pour confirmer que le premier est bien écrit. Une fois
# validé il faudra appeler RegisterForm comme méthode du
# controller mais uniquement si les champs répondent aux
# critères
```

Exemple - Page Web



Beeker va alors créer une page parfaitement en adéquation avec notre page exemple avec mes diverses demandes.

Inscription

Adresse mail

Mot de passe

Confirmez le mot de passe

S'inscrire

En conclusion, Beeker est capable d'accélérer les développements avec la réalisation d'actions. L'outil s'adapte également parfaitement pour des projets d'infrastructure as code ou de scripting.

Beeker s'appuyant sur ChatGPT, l'outil va continuellement gagner en efficacité et précision en plus de voir ses fonctionnalités augmentées en fonction de son propre développement.

Dans la même collection

Orchestration et Gestion de Conteneurs



Infrastructure as Code



Sécurité & Gestion des secrets



Développement & CI/CD



↓ FOLLOW ME ↓



[ANTONYCANUT](#)



[ANTONY KERVAZO-CANUT](#)