

Calc Engine

Übersicht Projekt

Überblick

Die CalcEngine ist ein System zur Berechnung von Fibonacci-Zahlen. Es verwendet eine API für die Eingabe, leitet Anfragen über einen RPC Client an eine Message Queue und nutzt Worker für die Berechnung.

Komponenten

API: Ermöglicht die Eingabe von Zahlen für die Berechnung.

RPC Client: Leitet Anfragen von der API an die Message Queue weiter.

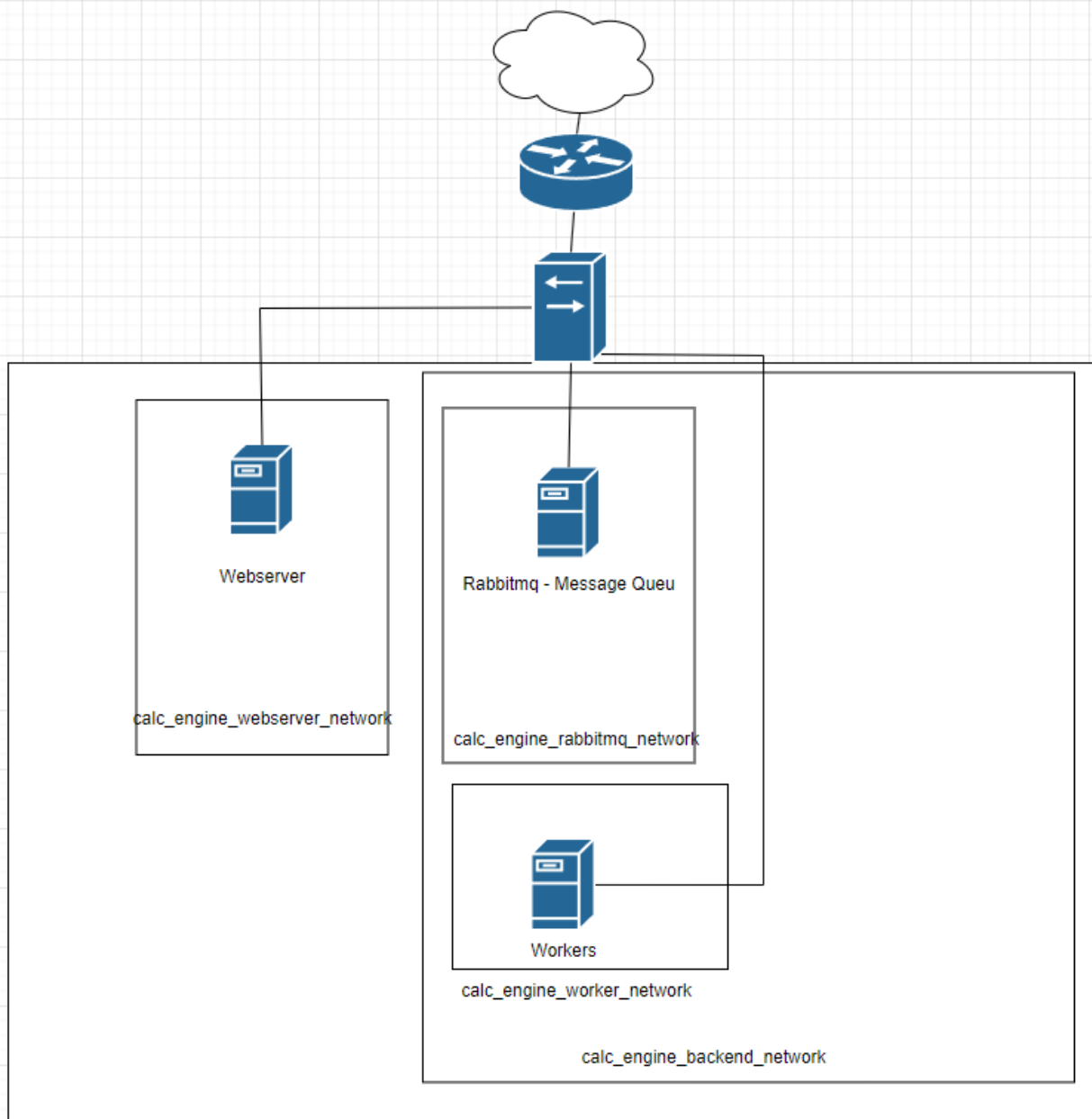
Message Queue: Verwaltet die Anfragen und die Kommunikation zwischen RPC Client und Workern.

Worker: Führt die Berechnungen durch und sendet Ergebnisse zurück zur Message Queue, welche dann über die API abrufbar sind.

Funktionsweise

1. Benutzer sendet über die API eine Zahl zur Berechnung.
2. Der RPC Client übergibt die Anfrage an die Message Queue.
3. Ein Worker holt sich die Anfrage, berechnet die Fibonacci-Zahl und gibt das Ergebnis zurück an die Message Queue.
4. Das Ergebnis wird über die API zugänglich gemacht

Netzwerkplan

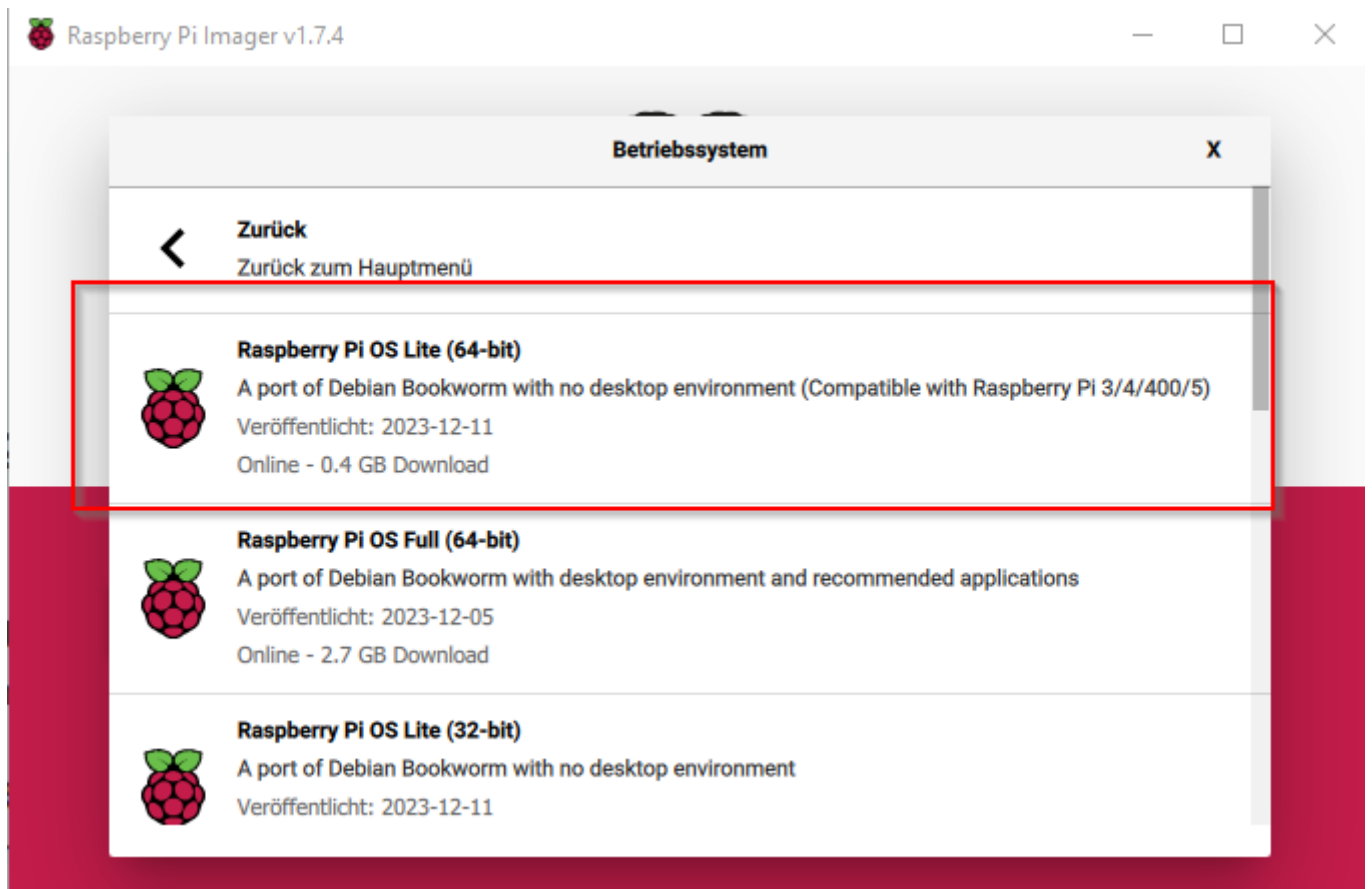
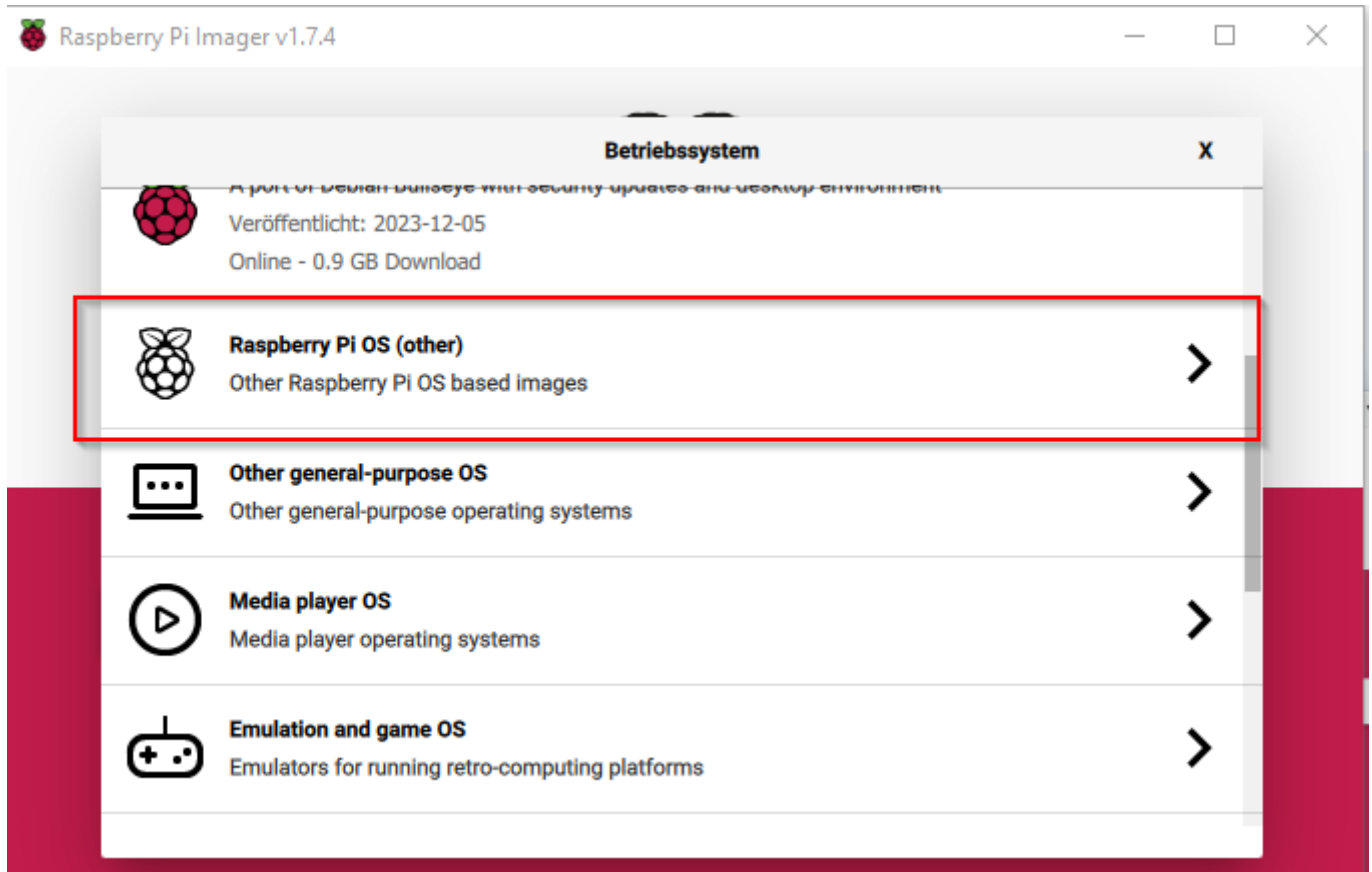


Installationsanleitung und Dokumentation

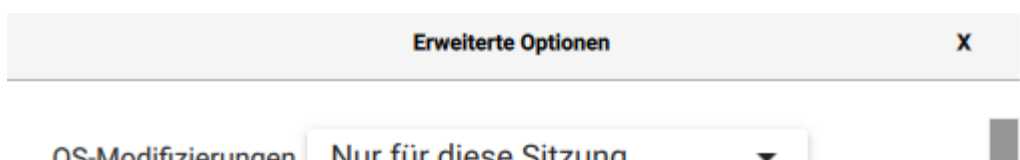
Installation Raspian

Wir installieren als erstes das Raspian Image neu auf unserem Raspberry Pi:

Mit dem Raspberry PI Imager installieren wir die Lite version des 64-BIT raspberry PI OS:



Im nächsten Schritt setzen wir die notwendigen Einstellungen und schreiben dann die SD Karte:



☒

Hostname:

☒

SSH aktivieren

☒

Password zur Authentifizierung verwenden

☐

Authentifizierung via Public-Key

authorized_keys für 'mario':

☒

Benutzername und Passwort setzen:

Benutzername:

Passwort:

☐

Wifi einrichten

SSID:

☐

Verborgene SSID

Passwort:

☒

Passwort anzeigen

Wifi-Land:

☒

Spracheinstellungen festlegen

Zeitzone:

Tastaturlayout:

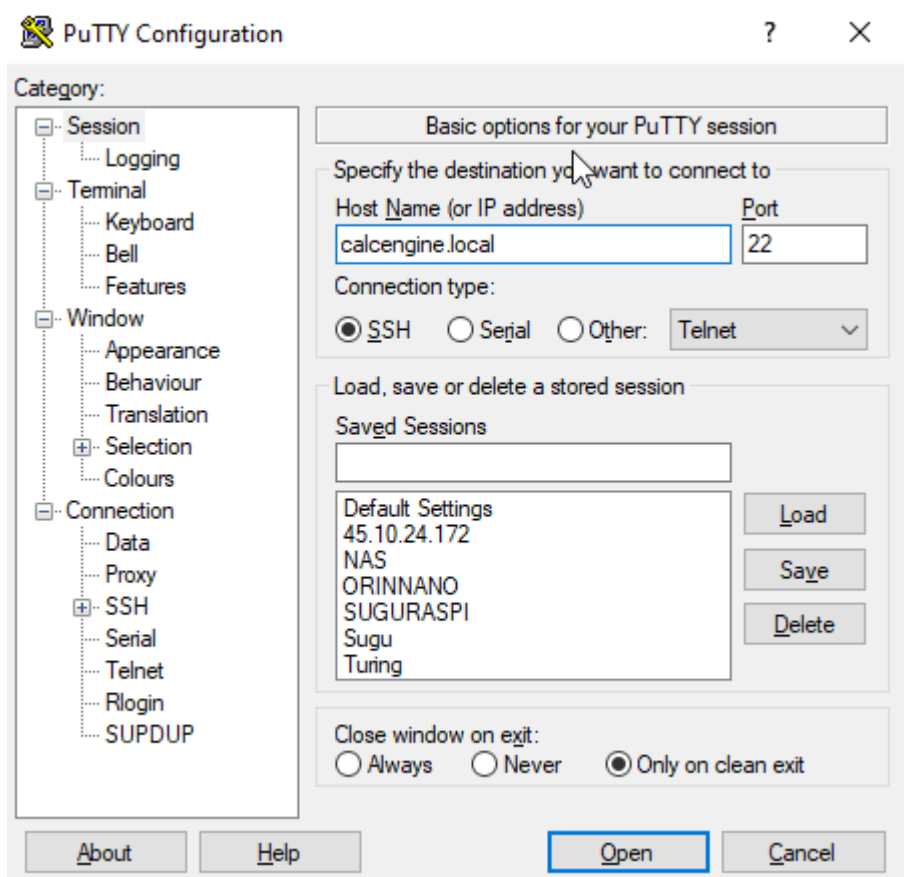
Dauerhafte Einstellungen

- ☐ Tonsignal nach Beenden abspielen
- ☒ Medien nach Beenden auswerfen
- ☒ Telemetry aktivieren



Installationen auf dem Raspberry PI:

Wenn wir die SD Karte beschrieben haben, können wir via SSH zum raspi connecten und installationen vornehmen, die wir benötigen. In unserem Beispiel verwenden wir Putty:



Nun gibt es 2 Möglichkeiten: entweder kann alles über unser erstelltes Script installiert werden (erfordert cleane Raspberr PI Installation) oder es kann der manuellen Installation gefolgt werden.

Installation mit Script

```
wget https://gist.githubusercontent.com/tekoWeMa/e0a87d1d7929a10232e1fdf071a7ad3b/raw/0d88cfb881b1536a34f31d3d5253f792a6ddb133/install.sh
```

```
chmod +x install.sh
```

```
./install.sh
```

Du musst uns den SSH Key für das Repo übergeben, dass du das Repo klonen kannst

Beim Kopieren des Keys: Ctrl+Shift+c

Nach der Installation um auszuführen:

```
cd csgo-teko-docker/
```

```
make
```

Manuelle Installation

Nach dem Login führen wir wie immer ein `sudo apt update` und `sudo apt upgrade -y` aus.

Als nächster Schritt installieren wir GIT und makefile.

Dies können wir mit den Folgenden Commands:

```
sudo apt install git make
```

```
y
```

Als nächstes installieren wir Docker, dafür folgen wir der Docker-Dokumentation:

Add Docker's official GPG key:

```
sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL [https://download.docker.com/linux/debian/gpg]
(https://download.docker.com/linux/debian/gpg) -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc
### Add the repository to Apt sources:

echo \

    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
[https://download.docker.com/linux/debian]
(https://download.docker.com/linux/debian) \

    $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \

    sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

```
y
```

Mit dem Testcommand können wir die Docker-installation testen:

```
sudo docker run hello-world
```

Als nächstes klonen wir das Repo. Dafür müssen wir im GIT den public SSH key anlagen:

```
ssh-keygen
```

`cat .ssh/id_rsa.pub` - Mit diesem Command können wir den Key auslesen, dass wir das Raspi hinterlegen können im GIT. (@Thomas: Wir benötigen diesen Key, dass wir dich hinterlegen können, dass du das Repo klonen kannst)

```
git clone git@github.com:sirh3e/csgo-teko-docker.git
```

Dann fügen wir den User der Dockergruppe hinzu (in unserem Fall ist der User mario, je nach Benutzernamen ist das ein anderer Benutzer)

```
Sudo usermod -aG docker mario
```

Als nächstes wechseln wir in das Verzeichnis und führen das Makefile aus:

```
cd csgo-teko-docker
```

```
make
```

Nach der Installation

Mit dem Link können wir nun auf die Webpage zugreifen:

<http://calcengine:5081/swagger/index.html>

Sirh3e.Teko.Docker.API1.0OAS3

http://calceengine-5081/swagger/v1/swagger.json

Sirh3e.Teko.Docker.API

GET/api/fib/{number}

Try it out

Parameters

Name	Description
number * required integer(\$int64) (path)	<input type="text" value="number"/>

Responses

Code	Description	Links
200	OK	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

0

Swagger

Select a definitionSirh3e.Teko.Docker.API v1

Sirh3e.Teko.Docker.API1.0OAS3

http://calceengine-5081/swagger/v1/swagger.json

Sirh3e.Teko.Docker.API

GET/api/fib/{number}

Cancel

Parameters

Name	Description
number * required integer(\$int64) (path)	<input type="text" value="12"/>

Execute

Responses

Code	Description	Links
200	OK	No links

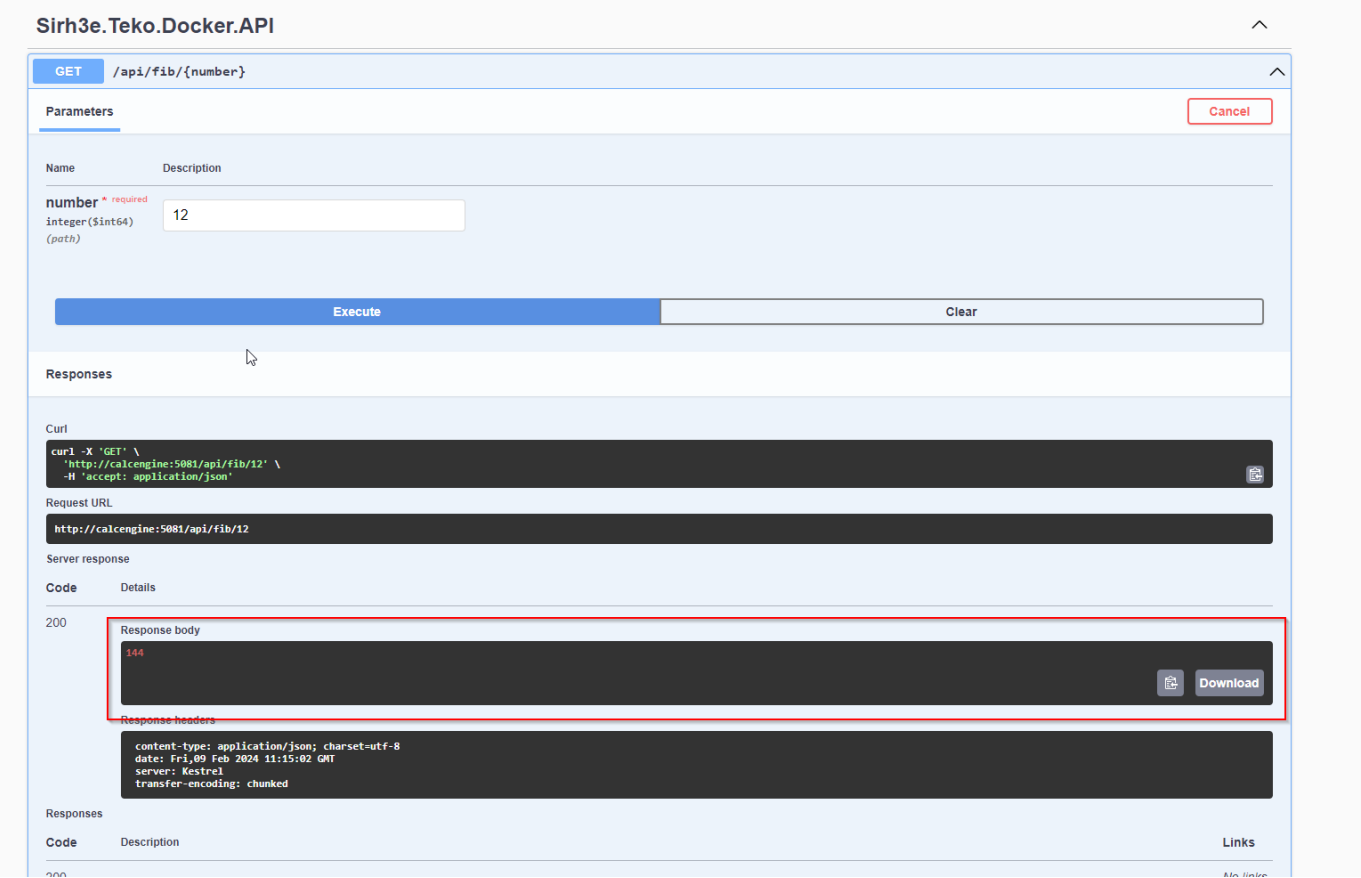
Media type

application/json

Controls Accept header.

Example Value | Schema

0



Fertigstellung

Nun können wir wie oben in den Bildern beschrieben die x-te Fibonacci-Zahl berechnen.