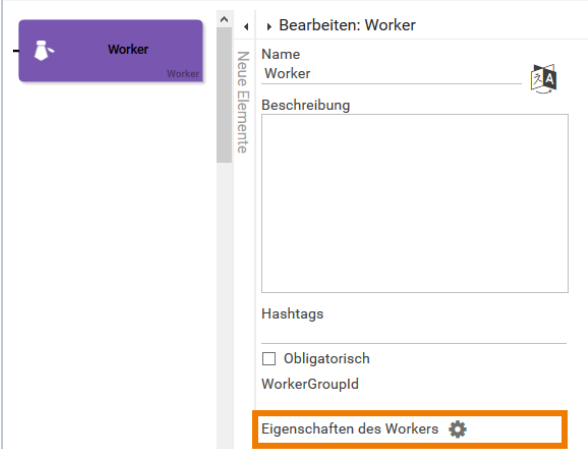


Umgebungsvariablen



Nach der Anlage eines Workers kann der Editor des Elements über die Option **Eigenschaften des Workers** geöffnet werden.

Auch ein Doppelklick auf das Worker-Element öffnet den Editor.

On this Page:

- [Die Umgebungsvariablen](#)

Related Pages:

- [Containerwerte beeinflussen](#)
- [Flüchtige Werte in die Datenbank speichern](#)

Related Documentation:

- [BPaaS](#)

Im Editor jedes Workers befinden sich bereits einige auskommentierte Angaben zu den verfügbaren Umgebungsvariablen:

Eigenschaften des Workers Worker

```
1 //
2 * Variables in scope:
3 * container: The container carrying the application data
4 * assemblyLine: The current assembly line
5 * next(): Runs the next step in the assembly line
6 * abort('abort message'): aborts the assembly line
7 * error('error message'): Reports an error and runs only the following mandatory workers
8 //
```

SPEICHERN ABBRECHEN



Bitte beachten: Worker-Code darf nicht mit einer Deklaration beginnen.

Die Umgebungsvariablen

Entwickler können die verfügbaren Umgebungsvariablen für die fortgeschrittene Anwendung von Workern nutzen.

Variable	Beschreibung
container	Die Klasse container bezeichnet den Container, der die Instanzen enthält. Über die Funktionen <code>container.put()</code> und <code>container.get()</code> kann man die Containerwerte beeinflussen .
assembly Line	Die assemblyLine dient zur Abarbeitung der EPK. Mit Hilfe der assemblyLine werden beispielsweise die Worker der EPK ausgeführt, die Formulare angezeigt etc.
next()	Die Funktion <code>next()</code> stößt den nächsten Schritt des aktuellen Prozesses an.
abort ('abort message')	Mit der Funktion <code>abort('abort message')</code> bricht man den aktuellen Prozess ab. Alle folgenden Prozessschritte werden nicht mehr ausgeführt.
error ('error message')	Die Funktion <code>error('error message')</code> meldet einen Fehler und führt in der Folge nur noch die Worker aus, die als obligatorisch markiert sind.



`next()` sollte ausschließlich zur Flusssteuerung in asynchronem JavaScript-Code genutzt werden. `abort()` und `error()` können auch in speziellen Anwendungsfällen genutzt werden, um die Ausführung des Prozesses zu beeinflussen.