

Inhaltsverzeichnis

YAML/JSON Reportgenerator mit EasyGen 3

YAML/JSON Reportgenerator mit EasyGen

Yaml und JSON sind wunderbare Datenformate für viele Arten von Datenmodellen, nur wie bekommt man diese Daten mal eben schnell in andere Formate wie CSV, LaTeX oder HTML konvertiert, ohne dafür einen eigenen Konverter programmieren zu müssen?

[EasyGen](#) bietet sich da als Lösung an. EasyGen lädt das Datenmodell direkt aus einer JSON oder Yaml-Datei und verwendet dann die [Go Template Engine](#), um ein Template aus einer Template-Datei mit den Werten aus dem Modell zu füllen.

Zugegebermaßen erfordert der Umgang mit der richtigen Syntax im Template etwas Lernschmerz und Übung, bis man ein Gefühl dafür bekommt, wo man sich gerade im Modell aufhält und wo genau gerade eigentlich der aktuelle Key und sein Wert zu finden ist, den man dann ja wieder für die nächste darunterliegende Iterationsschleife braucht.

Am Ende jedoch ist das Template viel kürzer und mit etwas Übung und Copy & Paste aus existierenden Macros dann schneller gemacht, als erst ein eigenes Programm zu frickeln, was die Daten konvertiert.

Hat man z.B. eine Liste von hinterlegten Requirements

[result.json](#)

```
{
  "mymodule": {
    "obj": {
      "requirements": [
        {
          "sev": "5",
          "text": [
            "\n",
            "Lorem ipsum dolor sit amet, consetetur
sadipscing elit,",
            "\n",
            "sed diam nonumy eirmod tempor invidunt ut
labore et dolore",
            "\n",
            "magna aliquyam erat, sed diam voluptua."
          ]
        },
        {
          "sev": "6",
          "text": [
            "\n",
            "At vero eos et accusam et",
            "\n",
            "justo duo dolores et ea rebum. Stet clita kasd
gubergren,",
            "\n",
            "no sea takimata sanctus est Lorem ipsum dolor
```



```

    {{- range $name, $values := .obj.candb -}}
        {{- if eq $name "messages" -}}
            <h2>{{ $name -}}</h2>
<table>
<tr>
<th>element</th>
<th>nodeName</th>
<th>messagename</th>
<th>length</th>
<th>messageid</th>
<th>signalname</th>
<th>name</th>
<th>bitlength</th>
<th>bitstart</th>
<th>endian</th>
<th>max</th>
<th>min</th>
<th>name</th>
<th>offset</th>
<th>scale</th>
<th>unit</th>
</tr>
        {{- range $messageid, $messageStructs := $values -}}
            {{- $length := $messageStructs.length -}}
            {{- $messagename := $messageStructs.name -}}
            {{- range $signalname, $signalvalues := $messageStructs
-}}
                {{- if eq $signalname "signals" -}}
                    {{- range $signalvalues -}}
                        <tr>
                            <td>{{ $element -}}:</td>
                            <td>{{- $nodeName -}}:</td>
                            <td>{{- $messagename -}}:</td>
                            <td>{{- $length -}}:</td>
                            <td>{{- $messageid -}}:</td>
                            <td>{{- $signalname -}}:</td>
                            <td>{{- .name }}</td>
                            <td>{{- .bitlength -}}:</td>
                            <td>{{- .bitstart -}}:</td>
                            <td>{{- .endian -}}:</td>
                            <td>{{- .max -}}:</td>
                            <td>{{- .min -}}:</td>
                            <td>{{- .name -}}:</td>
                            <td>{{- .offset -}}:</td>
                            <td>{{- .scale -}}:</td>
                            <td>{{- .unit -}}:</td>
                        <tr>
                            {{ end -}}
                    {{- end -}}
                {{- end -}}
            {{- end -}}
        {{- end -}}
    {{- end -}}

```

```
</table>
  {{- end -}}
{{- end -}}
{{- end -}}
```

From:

<http://koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://koehlers.de/wiki/doku.php?id=pc:easygen>

Last update: **2023/01/12 07:58**

