Tworzenie szablonów

Szablony dokumentów w THTG – wstęp

System **THTG** pozwala na generowanie wypełnionych dokumentów na podstawie:

- danych znajdujących się w systemie,
- szablonu dokumentu.

Do tego celu wykorzystywany jest **system znaczników znany z AngularJS**.

W momencie wyboru szablonu do generowania dokumentu system:

1. tworzy dane z systemu w formacie **JSON**,
2. pobiera **szablon dokumentu**,
3. przekazuje dane i szablon do aplikacji generującej dokument,
4. aplikacja buduje **uzupełniony dokument**.

Wyświetlanie tekstu

Dane można wyświetlać w dokumencie przy użyciu zapisu: {{pole}}

gdzie `pole` jest nazwą klucza z danych JSON.

Przykład

Dane:

```
{
  "imie": "Adam"
```

```
}
```

Szablon: Mam na imię {{imie}}.

Wynik: Mam na imię Adam.

Parser wyszukuje klucz imie w JSON i wstawia jego wartość.

Wyświetlanie danych z zagnieżdżonego JSON

Często dane JSON mają strukturę zagnieżdżoną.

- Dane:

```
{
  "imie": "Adam",
  "adres": {
    "ulica": "Poziomkowa",
    "miasto": "Poznań"
  }
}
```

Aby odwołać się do zagnieżdżonych danych, należy użyć kropki.

Mam na imię {{imie}}. Mieszkam na ulicy {{adres.ulica}} w mieście {{adres.miasto}}.

- Wynik: Mam na imię Adam. Mieszkam na ulicy Poziomkowa w mieście Poznań.

Głębsze zagnieżdżenia

Zagnieżdżenia mogą być wielopoziomowe.

Dane:

```
{
  "imie": "Adam",
  "adres": {
    "ulica": "Poziomkowa",
    "miasto": "Poznań",
```

```
    "poczta": {
      "kod": "61-100",
      "miasto": "m. Poznań"
    }
  }
}
```

- Szablon

Mam na imię {{imie}}. Mieszkam na ulicy {{adres.ulica}} w mieście {{adres.miasto}}, kod pocztowy {{adres.poczta.kod}} poczta {{adres.poczta.miasto}}.

- Wynik

Mam na imię Adam. Mieszkam na ulicy Poziomkowa w mieście Poznań, kod pocztowy 61-100 poczta m. Poznań.

Wchodzenie do zagnieżdżenia

Aby uniknąć wielokrotnego wpisywania nazwy obiektu, można wejść w jego kontekst.

- Do tego służy zapis:

{{#klucz}} ... {{/}}

- Przykład

Mam na imię {{imie}}. Mieszkam na ulicy {{#adres}}{{ulica}} w mieście {{miasto}}, kod pocztowy {{poczta.kod}} poczta {{poczta.miasto}}{{/}}

Po użyciu {{#adres}} parser przechodzi do wnętrza obiektu adres. {{/}} oznacza powrót do wcześniejszego poziomu.

Wyszukiwanie kluczy przez parser

Jeśli parser nie znajdzie klucza w bieżącym poziomie, zacznie szukać wyżej w strukturze.

- Dane

```
{
  "imie": "Adam",
  "adres": {
    "ulica": "Poziomkowa"
```

```
  },
  "zona": {
    "imie": "Natalia",
    "nazwisko": "Nowak"
  }
}
```

- Szablon

{{#adres}}Mam na imię {{imie}}, mieszkam na ulicy {{ulica}}.{{/}}

- Wynik

Mam na imię Adam, mieszkam na ulicy Poziomkowa.

Parser nie przechodzi do innych gałęzi struktury JSON – szuka tylko w górę struktury.

Instrukcje warunkowe

Instrukcje warunkowe pozwalają wyświetlać treść tylko wtedy, gdy spełniony jest określony warunek.

- Dane:

```
{
  "imie": "Adam",
  "plec": "mezczyzna"
}
```

- Szablon:

{{#plec=='mezczyzna'}}Pan{{/}} {{imie}}

- Wynik

Pan Adam

Jeśli warunek nie zostanie spełniony, tekst nie zostanie wyświetlony.

- Przykład:

{{#plec=='kobieta'}}Pani{{/}} {{imie}}

- Wynik:

Adam

Złożone warunki

Warunki mogą być bardziej rozbudowane, tak jak w JavaScript.

- Przykład

{{#plec=='mezczyzna' && imie=='Adam'}}Ok{{/}}

Tekst Ok pojawi się tylko wtedy, gdy:

- plec = mezczyzna
- imie = Adam

Filtry

Filtry pozwalają formatować dane wyświetlane w dokumencie.

Formatowanie ceny (currency)

- Dane:

```
{
  "price": 10032.23
}
```

- Szablon:

{{price | currency}}

- Wynik:

10 032,23 zł

Wyświetlanie kwoty bez waluty

- Szablon:

Cena komórki lokatorskiej: {{price | currency:''}} PLN

- Wynik

Cena komórki lokatorskiej: 10 032,23 PLN

Wyświetlanie kwoty z własną walutą

- Szablon:

Cena komórki lokatorskiej: {{price | currency:'PLN'}}

- Wynik:

Cena komórki lokatorskiej: 10 032,23 PLN

Wersja #1
Utworzono 2026-03-09 13:13:03 CET przez Patrycja
Zaktualizowano 2026-03-09 13:25:24 CET przez Patrycja