

DOKUMENTACJA

VISUALNETPRINTPLUGIN



wersja: 1.6 (25102012)
producent: visualnet.pl

Dostępne pola dla biletu (dane z formatu JSON):

NAZWA	TYP	OPIS
template	STRING	nazwa szablonu dla biletu
barcode	STRING	kod EAN 13 dla kodu paskowego
id	INTEGER	Identyfikator biletu
invitation	BOOLEAN	Flaga określająca czy dany bilet jest zaproszeniem
amount rawAmount unitAmount unitRawAmount	NUMBER	Wartość biletu (najczęściej wykorzystywana jest wartość unitAmount)
quantity	INTEGER	Ilość sztuk danego biletu
name	STRING	Nazwa wydarzenia
receiptname	STRING	Nazwa na paragonie
ticketlabel	STRING	Nazwa etykiety
row	INTEGER	Numer rzędu dla miejsca
number	INTEGER	Numer miejsca
section	MIXED	Nazwa / numer sekcji

eventname	STRING	Nazwa wydarzenia
eventdate	STRING	Data wydarzenia w formatach iso, timestamp, date
location	STRING	Dane lokalizacji

Przykładowy JSON z danymi:

```
[
  {
    "template": "ticket-test1",
    "barcode": "2000023360127",
    "id": "430",
    "invitation": false,
    "amount": "88,00",
    "rawAmount": 8800,
    "unitAmount": "88,00",
    "unitRawAmount": 8800,
    "quantity": 1,
    "name": "Trasa czerwona",
    "receiptname": "Trasa czerwona",
    "ticketlabel": "Zabili go i ucieł",
    "performers": "",
    "row": "2",
    "number": "1",
    "section": "3",
    "eventname": "Zabili go i ucieł",
    "eventdate": {
      "iso": "2012-06-30T19:00:00+02:00",
      "timestamp": "1341075600",
      "date": "2011-07-07 19:30:25"
    },
    "location": {
      "name": "FMF",
      "street": "testowa",
      "city": "Abramów",
      "zip": ""
    }
  }
]
```

Opis szablonu dla biletu:

Szablony do opisu wydruku biletów są zgodne z notacją JSON, dlatego też plugin sprawdza na wejściu czy wprowadzane dane są zgodne z tym formatem.

Użytecznym narzędziem jest <http://jsonlint.com/> który sprawdza składnię i odpowiednio formatuje zapis.

Przegląd sekcji:

1.) Sekcja name:

W tej sekcji podana jest nazwa szablonu, która jest przypisywana do każdego biletu

2.) Sekcja init:

SCHEMAT

```
"init": {  
  "printerOrientation": "[ENUM]",  
  "widthArea": [INTEGER],  
  "heightArea": [INTEGER],  
  "defaultText": {  
    "family": "[STRING]",  
    "size": [INTEGER],  
    "style": "[ENUM]"  
  }  
}
```

PRZYKŁAD

```
"init": {  
  "printerOrientation": "landscape",  
  "widthArea": 60,  
  "heightArea": 95,  
  "defaultText": {  
    "family": "Arial",  
    "size": 15,  
    "style": "italic"  
  }  
}
```

NAZWA	TYP	OPIS
printerOrientation	ENUM	Określa orientację dla wydruku Dostępne wartości: <code>#define</code> VISUALSTRING_ORIENT_LANDSCAPE "landscape" - orientacja pozioma <code>#define</code> VISUALSTRING_ORIENT_PORTRAIT "portrait" - orientacja pionowa
widthArea	INTEGER	Podana w milimetrach szerokość wydruku
heightArea	INTEGER	Podana w milimetrach wysokość wydruku
defaultText	JSON OBJECT	Domyślne ustawienie globalnego tekstu

defaultText.family	STRING	Określenie domyślnej rodziny czcionki np. Arial, Times New Roman
defaultText.size	NUMBER	Rozmiar domyślnej czcionki dla wydruku
defaultText.style	ENUM	<p>Styl domyślnej czcionki</p> <p>Dostępne wartości:</p> <pre> #define VISUALSTRING_TEXT_STYLE_NORMAL "normal" - krój normalny #define VISUALSTRING_TEXT_STYLE_BOLD "bold" - pogrubienie #define VISUALSTRING_TEXT_STYLE_ITALIC "italic" - kursywa #define VISUALSTRING_TEXT_STYLE_UNDERLINE "underline" - podkreślenie #define VISUALSTRING_TEXT_STYLE_STRIKEOUT "strike" - przekreślenie </pre>

2.) Sekcja elements:

2.1.) Element podstawowy:

SCHEMAT

```
{
  "field": "[STRING]",
  "x": [INTEGER],
  "y": [INTEGER],
  "type": [ENUM]
  "content": "[STRING]"
  "text": {
    "family": "[STRING]",
    "size": [INTEGER],
    "style": [ENUM]
  }
}
```

PRZYKŁAD

```
{
  "field": "name",
  "x": 35,
  "y": 10,
  "type": "dynamic",
  "text": {
    "family": "Arial",
    "size": 10,
    "style": "normal"
  }
}
```

NAZWA	TYP	OPIS
field	STRING	Nazwa pola
x	INTEGER	Położenie elementu w osi x (zależne od orientacji wydruku)
y	INTEGER	Położenie elementu w osi y (zależne od orientacji wydruku)
type	ENUM	<p>Typ elementu</p> <p>Dostępne wartości:</p> <p>#define VISUALSTRING_ELEMENT_DYNAMICTYPE "dynamic" – dynamiczny, dane pobierane są wg nazwy elementów z jsona biletu</p> <p>Przykład:</p> <pre>{ "field": "name", "x": 35, "y": 10, "type": "dynamic", "text": { "family": "Arial", "size": 10, "style": "normal" } }</pre>

		<pre>#define VISUALSTRING_ELEMENT_BARCODETYPE "barcode" - element generujący kod paskowy EAN13 wg wartości barcode z danych biletu Przykład: { "field": "barcode", "x": 40, "y": 50, "type": "barcode" }</pre> <pre>#define VISUALSTRING_ELEMENT_STATICTYPE "static" - element statyczny, dane pobierane są z pola content Przykład: { "field": "firmname", "x": 50, "y": 80, "type": "static", "content": "visualnet.pl" }</pre>
content	STRING	Wartość dla elementu typu static
text	JSON OBJECT	Wartości takie same jak w sekcji init elementu defaultText

2.2.) Element date:

Dane dla tego elementu pobierane są z wartości **eventdate** biletu

SCHEMAT

```
{
  "field": "[STRING]",
  "x": [INTEGER],
  "y": [INTEGER],
  "type": "dynamic",
  "category": {
    "name": "date",
    "pattern": [STRING]
  },
  "text": {
    "family": "[STRING]",
    "size": [INTEGER],
    "style": "[ENUM]"
  }
}
```

PRZYKŁAD

```
{
  "field": "date",
  "x": 53,
  "y": 19,
  "type": "dynamic",
  "category": {
    "name": "date",
    "pattern": "%d %B %Y"
  },
  "text": {
    "family": "Arial",
    "size": 10,
    "style": "normal"
  }
}
```

NAZWA	TYP	OPIS
field	STRING	Nazwa pola
x	INTEGER	Położenie elementu w osi x (zależne od orientacji wydruku)
y	INTEGER	Położenie elementu w osi y (zależne od orientacji wydruku)
type	STRING	Dla tego elementu typem jest dynamic
category	JSON OBJECT	Wartość kategorii elementu
category.name	STRING	Dla elementu daty wymaganą wartością jest date
category.pattern	STRING	Wzorzec wyświetlanej daty Dostępne modyfikatory: %A – pełna nazwa dnia tygodnia np. <i>poniedziałek, czwartek</i> %B – pełna nazwa miesiąca np. <i>stycznia, marca</i>

		<p>%d – dzień miesiąca zapisany 2-cyfrowo np. 01, 10</p> <p>%m – miesiąc zapisany 2-cyfrowo np. 01, 12</p> <p>%y – rok zapisany 2-cyfrowo np. 2012 => 12</p> <p>%Y – rok zapisany 4 –cyfrowo np. 2012</p> <p>%H – godzina w systemie 24H np. 12, 13</p> <p>%M – minuta zapisana 2-cyfrowo np. 01, 12</p> <p>%S - sekunda</p>
text	JSON OBJECT	Wartości takie same jak w sekcji init elementu defaultText

2.3.) Element replace:

Umożliwia podmianę wartości w zdefiniowanym wzorcu. Wartości **name** w sekcji **fields** odpowiadają wartością z biletu. Wartość **root** jest opcjonalna i oznacza element który jest obiektem np. pobieranie wewnętrznych wartości z **location**

SCHEMAT

```
{
  "field": "[STRING]",
  "x": [INTEGER],
  "y": [INTEGER],
  "type": "dynamic",
  "category": {
    "name": "replace",
    "root": "[STRING]"
  },
  "pattern": "[STRING]",
  "fields": [
    {
      "name": "[STRING]"
    },
    {
      [...]
    }
  ]
}
```

PRZYKŁAD

```
{
  "field": "price",
  "x": 84,
  "y": 45,
  "type": "dynamic",
  "category": {
    "name": "replace",
    "pattern": "{0} {1} PLN",
    "fields": [
      {
        "name": "name"
      },
      {
        "name": "unitAmount"
      }
    ]
  }
}
```

NAZWA	TYP	OPIS
field	STRING	Nazwa pola
x	INTEGER	Położenie elementu w osi x (zależne od orientacji wydruku)
y	INTEGER	Położenie elementu w osi y (zależne od orientacji wydruku)
type	STRING	Dla tego elementu typem jest dynamic
category	JSON OBJECT	Wartość kategorii elementu
category.name	STRING	Dla elementu daty wymaganą wartością jest replace
category.root	STRING	Nazwa obiektu z biletu w którym umieszczone są docelowe dane Przykład: Element location z biletu

		<pre> "location": { "name": "FMF", "street": "testowa", "city": "Abramów", "zip": "" } </pre> <p>Szablon elementu:</p> <pre> { "field": "price", "x": 84, "y": 45, "category": { "name": "replace", "root": "location" }, "pattern": "Miejsce: {0} {1} {2}", "fields": [{ "name": "name" }, { "name": "street" }, { "name": "city" }] } </pre> <p>Wynik: „Miejsce: FMF testowa Abramów”</p>
category.pattern	STRING	<p>Wzorzec wyświetlanego stringu. Wartości w nawiasach klamrowych (zaczynając od 0 .. n) odpowiadają numerom elementów w sekcji fields</p> <p>Przykład:</p> <p>unitAmount z biletu 88,00 dla</p> <pre> { "field": "price", "x": 84, "y": 45, "category": { "name": "replace", "pattern": "{0} PLN", "fields": [{ "name": "unitAmount" }] } } </pre> <p>Na wydruku pokaże się „88,00 PLN”</p>

category.fields	JSON OBJECT	Tablica z elementami pól. Każde pole posiada swoją nazwę name
text	JSON OBJECT	Wartości takie same jak w sekcji init elementu defaultText

2.3.) Element choice:

Umożliwia warunkowe wyświetlanie danych.

Wartość **root** jest opcjonalna i oznacza element który jest obiektem np. pobieranie wewnętrznych wartości z **location**

SCHEMAT

```
{
  "field": "[STRING]",
  "x": [INTEGER],
  "y": [INTEGER],
  "type": "dynamic",
  "category": {
    "name": "choice",
    "pattern": "Zaproszenie|{0} PLN",
    "fields": [
      {
        "name": "unitAmount"
      }
    ],
    "condition": "[string]"
  },
  "text": {
    "family": "Arial",
    "size": 12,
    "style": "normal"
  }
}
```

PRZYKŁAD

```
{
  "field": "invitation-desc",
  "x": 80,
  "y": 10,
  "type": "dynamic",
  "category": {
    "name": "choice",
    "pattern": "Zaproszenie|{0} PLN",
    "fields": [
      {
        "name": "unitAmount"
      }
    ],
    "condition": "invitation"
  },
  "text": {
    "family": "Arial",
    "size": 12,
    "style": "normal"
  }
}
```

NAZWA	TYP	OPIS
field	STRING	Nazwa pola
x	INTEGER	Położenie elementu w osi x (zależne od orientacji wydruku)
y	INTEGER	Położenie elementu w osi y (zależne od orientacji wydruku)
type	STRING	Dla tego elementu typem jest dynamic
category	JSON OBJECT	Wartość kategorii elementu
category.name	STRING	Dla elementu daty wymaganą wartością jest replace
category.root	STRING	Nazwa obiektu z biletu w którym umieszczone są docelowe dane

		<p>Przykład:</p> <p>Element location z biletu</p> <pre>"location": { "name": "FMF", "street": "testowa", "city": "Abramów", "zip": "" }</pre> <p>Szablon elementu:</p> <pre>{ "field": "price", "x": 84, "y": 45, "category": { "name": "replace", "root": "location" }, "pattern": "Miejsce: {0} {1} {2}", "fields": [{ "name": "name" }, { "name": "street" }, { "name": "city" }] }</pre> <p>Wynik: „Miejsce: FMF testowa Abramów”</p>
category.pattern	STRING	<p>Wzorzec wyświetlanego stringu. Wartości w nawiasach klamrowych (<i>zaczynając od 0 .. n</i>) odpowiadają numerom elementów w sekcji fields</p> <p>Przykład:</p> <p>unitAmount z biletu 88,00 dla</p> <pre>{ "field": "price", "x": 84, "y": 45, "category": { "name": "replace", "pattern": "{0} PLN", "fields": [{ "name": "unitAmount" }] } }</pre>

		Na wydruku pokaże się „88,00 PLN”
category.fields	JSON OBJECT	Tablica z elementami pól. Każde pole posiada swoją nazwę name
category.condition	STRING	<p>Warunek dla pobierania danych.</p> <p>Dostępne wartości:</p> <p>#define VISUALSTRING_CATEGORY_INVITATIONCONDITION "invitation" – dane z pola wzorca (pattern) są wypełniane w zależności od tego czy bilet jest zaproszeniem (pole invitation z biletu)</p> <p>Przykład:</p> <pre> { "field": "description", "x": 80, "y": 10, "type": "dynamic", "category": { "name": "choice", "pattern": "Zaproszenie {0} PLN", "fields": [{ "name": "unitAmount" }], "condition": "invitation" }, "text": { "family": "Arial", "size": 12, "style": "normal" } } </pre> <p>Jeśli wartość invitation z biletu będzie <i>true</i> otrzymamy na wyjściu „Zaproszenie” w wypadku <i>false</i> „88,00 PLN”</p> <p>#define VISUALSTRING_CATEGORY_PLACECONDITION "place" – dla tego warunku dane z pola wzorca (pattern) wypełniane są zależnie od pól row, number, section z biletu</p> <p>Przykład:</p>

		<pre> { "field": "description", "x": 80, "y": 15, "type": "dynamic", "category": { "name": "choice", "pattern": "Miejsca nienumerowane {0} {1} {2}", "fields": [{ "name": "section", "text": "Sekcja:" }, { "name": "row", "text": "Rząd:" }, { "name": "number", "text": "Miejsce:" }], "condition": "place" }, "text": { "family": "Arial", "size": 12, "style": "normal" } } </pre>
text	JSON OBJECT	Wartości takie same jak w sekcji init elementu defaultText

Tabela wartości wyjściowych dla pól sektor, rząd, miejsce – warunek „**place**”.

Dla wzorca: „**Miejsca nienumerowane|{0} {1} {2}**”

Wartości:

```

{
  "name": "section",
  "text": "Sekcja:"
},
{
  "name": "row",
  "text": "Rząd:"
},
{
  "name": "number",
  "text": "Miejsce:"
}

```

Sektor	Rząd	Miejsce	Przypadek	Wartość
pusty	pusty	pusty	0	<i>Miejsca nienumerowane</i>
pusty	pusty	1	1	<i>Miejsce: 1</i>
pusty	2	pusty	2	<i>Rząd: 2 Miejsca nienumerowane</i>
pusty	2	1	3	<i>Rząd: 2 Miejsce: 1</i>
3	pusty	pusty	4	<i>Sektor: 3 Miejsca nienumerowane</i>
3	pusty	1	5	<i>Sektor: 3 Miejsce: 1</i>
3	2	pusty	6	<i>Sektor: 3 Rząd: 2 Miejsca nienumerowane</i>
3	2	1	7	<i>Sektor: 3 Rząd: 2 Miejsce: 1</i>