# Общие сведения

Обработка предназначена для формирования HTML-страниц по шаблонам в стиле Django / Flask.

Вводный пример - подстановка значения переменной.

**Шаблон**

<!DOCTYPE html>

<html>

<head>

</head>

<body>

**{{ &w.Title }}**

</body>

</html>

**Результат**

DOCTYPE html>

<html>

<head>

</head>

<body>

**Creating HTML by templates in Django/Flask style**

</body>

</html>

Концепция

Данные для шаблона передаются в виде структуры, доступ к которой из шаблона осуществляется через специальную переменную «**&w**».

Структура может содержать как простые переменные (число, строка, дата), так и коллекции, т.е. массивы, структуры и ссылочные объекты.

**Массив.**

Доступ к элементам осуществляется операцией [].

**Структуры.**

Доступ к полям осуществляется операцией [] и обращением через точку.

**Ссылочные объекты.**

Доступ к полям осуществляется операцией [] и обращением через точку.

Также реализована обработка циклов двух видов (подробности – ниже) и условий.

# Синтаксис шаблонов.

Шаблоны используют язык 1С с некоторыми ограничениями. Возможны русскоязычный и англоязычный варианты синтаксиса. Не допускается смешивание вариантов синтаксиса в одной конструкции.

Для указания в тексте переменной используется тег {{ }}.

## Вывод переменной

{{ &w.ИмяПеременной }}

## Доступ к полям коллекции

{{ &w.Коллекция.Поле }}

В этом случае «Коллекция» должна быть либо структурой, либо ссылочным объектом. У любого из этих объектов должен быть реквизит (поле) с именем «Поле».

Возможен доступ через две и более точки. В этом случае получение полей выполняется рекурсивно.

Пример:

<p> Контрагент {{ &w.Контрагент.Имя }} </p>

<p> Контрагент.Адрес.Город {{ &w.Контрагент.Адрес.Город }} </p>

Переменная «Контрагент» может быть объявлена следующим образом (json):

*"Контрагент": {*

*"Имя": "Банк ФК Открытие",*

*"Адрес": {*

*"Город": "Москва"*

*}*

*},*

Также это может быть ссылочный объект.

### Ссылочные объекты.

В случае ссылочного объекта можно использовать переменную только по имени. Представление объекта будет сформировано платформой.

<p> Контрагент {{ &w.Контрагент }} </p>

## Условия.

Для оформления условий служит тег {% %}.

{% If 1=1 Then %}

<p>секция ЕСЛИ</p>

{% ElsIf 1=2 Then %}

<p>секция ИНАЧЕ ЕСЛИ</p>

{% Else %}

<p>секция ИНАЧЕ</p>

{% EndIf %}

{% Если 1=1 Тогда %}

<p>секция ЕСЛИ</p>

{% Иначе 1=2 Тогда %}

<p>секция ИНАЧЕ ЕСЛИ</p>

{% Иначе %}

<p>секция ИНАЧЕ</p>

{% КонецЕсли %}

В условии допустимы следующие выражения.

Получение элемента коллекции

{% If &w.МассивЭлементов[0] = 1 Then %}

Сравнение переменных, объявленных в данных шаблона

{% If &w.КоличествоЗаказов = 1 Then %}

Получение через две точки и более

{% Если &w.Контрагент.Адрес.Город="" Тогда %}

Получение итератора цикла

Подробнее об этом – в разделе Циклы.

### Вычисление условий.

При сравнении строки с пустой строкой также выполняется сравнение с Неопределено.

Для вычисления условий используется метод языка 1С Вычислить().

## Циклы.

### Цикл Для Каждого Элемент Из Коллекция Цикл

Выполняет обход коллекции

{% For Each СтрокаДанных In &w.ТаблицаДанных Do %}

{% EndDo %}

{% Для Каждого СтрокаДанных Из &w.ТаблицаДанных Цикл %}

{% КонецЦикла %}

Внутри цикла возможно обращение к итератору «СтрокаДанных» через точку, если он представляет собой коллекцию. Пример: массив структур (представление таблицы значений на клиенте и для передачи через HTTP-ответ JSON).

Пример вывода таблицы.

<table>

<tr>

<th>#</th>

<th>Account</th>

<th>Total amount</th>

</tr>

{% For Each СтрокаДанных In &w.ТаблицаДанных Do %}

<tr>

<td>{{ СтрокаДанных.RowNum }}</td>

<td style="text-align: left;">{{ СтрокаДанных.Account }}</td>

<td>{{ СтрокаДанных.TotalAmount }}</td>

</tr>

{% EndDo %}

</table>

ТаблицаДанных задается, как массив структур. Фрагмент JSON:

*{*

*"ТаблицаДанных": [*

*{*

*"RowNum": 1,*

*"Account": "Account #5678967890 at Standard Chartered Bank",*

*"TotalAmount": "USD 8.993.340"*

*},*

*{*

*"RowNum": 2,*

*"Account": "Account #1100000 at HSBC Singapore",*

*"TotalAmount": "USD 1.000.000"*

*},*

*{*

*"RowNum": 3,*

*"Account": "Account #7674839302 at Barclays bank",*

*"TotalAmount": "GBP 34.554"*

*}*

*]*

*}*

### Цикл Для Счетчик = 1 по 10 Цикл

{% For count = 0 To 4 Do %}

{% EndDo %}

{% Для count = 0 По 4 Цикл %}

{% КонецЦикла %}

Пример вывода массива.

<ul>

{% For count = 0 To 4 Do %}

<li>счетчик {{ count }}, Элемент массива {{ &w.МассивЭлементов[count] }}</li>

{% EndDo %}

</ul>

JSON-фрагмент массива

"МассивЭлементов": [

1,

2,

3,

4,

5

],

# Работа с типом «Дата».

Чтение JSON предполагает знание о том, какие поля являются датой. Это следует из описания метода глобального контекста ReadJSON(). Для правильного чтения дат служат следующие параметры этого метода:

* PropertiesWithDateValuesNames
* ExpectedDateFormat

Преобразование JSON в структуру происходит в методе  
*GetTemplateDataFromJSON().*

Дата должна быть в формате ‘2019-04-29T00:00:00.0’.

Для правильного чтения таких полей, следует передать их список в параметре метода:

*ResultHTML = DataProc.CreateHTMLServer( "ReportDate" );*

# Особенности.

## Имена итераторов.

При использовании вложенных циклов имена итераторов следует использовать уникальные, не совпадающие друг с другом по подстроке. Это связано с тем, что имена итераторов заменяются на конструкцию

*mIterators["ИмяИтератора"]*

Пример.

*{% For Each СтрокаТаблицы In &w.ТаблицаДанныхДляТестаУсловия Do %}*

*{% For Each Строка\_Таблицы2 In &w.ВложеннаяТаблицаДанныхДляТестаУсловия Do %}*

*{% Если СтрокаТаблицы.Amount=5000*

*и Строка\_Таблицы2.rownum = 1 Тогда %}*

*<p> Проверка доступа к переменной верхнего уровня</p>*

*<p> из вложенного циклаа - ОК</p>*

*{% КонецЕсли %}*

*{% EndDo %}*

*{% EndDo %}*

В данном примере выражение  
*СтрокаТаблицы.Amount=5000*будет заменено на  
*mIterators["строкатаблицы "].Amount=5000*

*mIterators* - это соответствие, которое хранит текущее значение итератора цикла.

Если итератор вложенного цикла *Строка\_Таблицы2* будет называться *СтрокаТаблицы2*, произойдет ошибка. Это случится из-за того, что выражение  
 *СтрокаТаблицы2.rownum = 1*    
будет преобразовано к   
 *mIterators["строкатаблицы"]2.rownum =1*,  
т.к. все вхождения подстроки «*строкатаблицы*» будут заменены на обращение к соответствию *mIterators.*

По этой же причине нельзя использовать имена итераторов в строковых константах внутри выражений условия.