

# Програмні складові Node-RED. Node Storing Context

Node Strong context використовується для збереження інформації в змінних. Використовується 3 варіанти контексту

- Global - visible to all nodes
- Flow - visible to all nodes on the same flow (or tab in the editor)
- Node - only visible to the node that set the value

## Node-RED API Context

### Global context - глобальний контекст

Зберігає глобальні параметри. При чому, глобальні параметри залишаються, навіть коли перебудовується потік. Основні операції з контекстом

- записати в контекст

```
var g_callcounter = g_callcounter +1 ;  
// записати в глобальний контекст з ключем callcount  
global.set('callcount', g_callcounter) ;
```

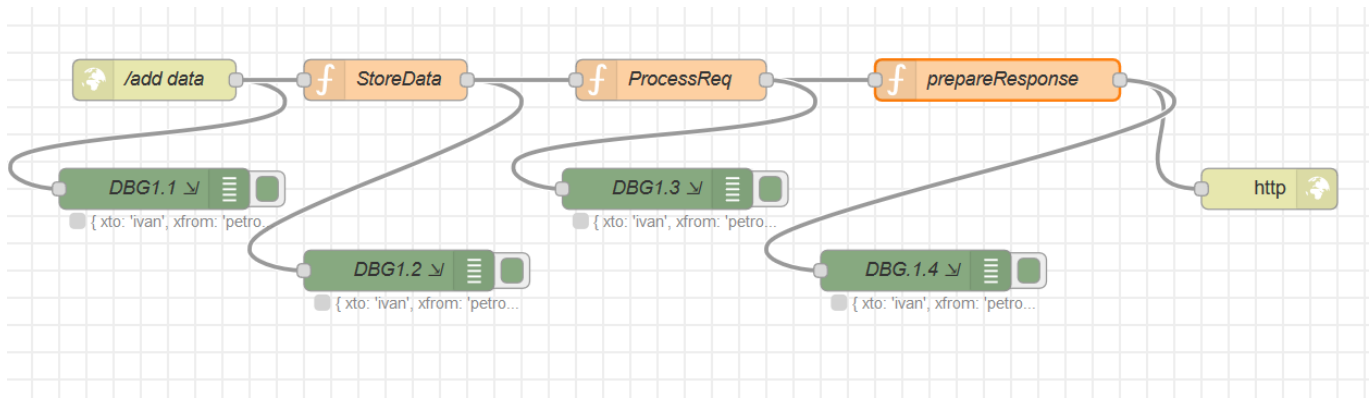
- прочитати параметр глобального контексту

```
// прочитати параметр глобального контексту  
g_callcounter= global.get('callcount');
```

- отримати список всіх параметрів глобального контексту

```
// отримати перелік всіх параметрів глобального контексту в масив,  
// та повернути у відповіді  
var keys = global.keys();  
var v,k;  
var pl = {}  
for(var i in keys){  
    v = global.get(keys[i])  
    k = keys[i]  
    pl[k] = v  
}  
msg.payload.keys = pl ;
```

Для тестування роботи контексту розроблений тестовий потік



Потік у файлі Flows/flow-2-globalcontext.json

node: **StoreData** Створює глобальний контекст, що відраховує звернення до методі

```
/**
 * Демонстрація використання глобального контексту
 * Для демонстрації створена глобальна змінна callcount, що підраховує
 звернення до сервісу
 *
 * При формуванні відповіді сервісу, кількість звернень повертається у
 реквізиті: calls
 *
 */
var g_callcounter;
// прочитати параметр глобального контексту
g_callcounter= global.get('callcount');
if (typeof g_callcounter === 'undefined') {
    global.set('callcount', 0) ;
    g_callcounter = global.get('callcount');
}
g_callcounter = g_callcounter +1 ;
// записати в глобальний контекст
global.set('callcount', g_callcounter) ;
return msg;
```

Node: **prepareResponse** Формує відповідь

```
/**
 * Формування відповіді
 *
 */
// прочитати параметр глобального контексту
g_callcounter= global.get('callcount');
//Передати контекст у відповіді
msg.payload.calls = g_callcounter ;
// отримати перелік всіх параметрів глобального контексту в масив,
// та повернути у відповіді
var keys = global.keys();
var v,k;
```

```
var pl = {}  
for(var i in keys){  
    v = global.get(keys[i])  
    k = keys[i]  
    pl[k] = v  
}  
msg.payload.keys = pl ;  
return msg;
```

## Тестування сервісу

- Сервіс приймає на вхід post запит

url=https://nod-red-wshp.eu-gb.mybluemix.net/add

Headers:

```
Content-Type: application/json
```

### Request:

```
{  
  "xto": "ivan",  
  "xfrom": "petro",  
  "xsubj": "read the book",  
  "xmsg": "Hi I am here!"  
}
```

### Response:

```
{  
  "xto": "ivan",  
  "xfrom": "petro",  
  "xsubj": "read the book",  
  "xmsg": "Hi I am here!",  
  "calls": 18,  
  "keys": {  
    "rsstojson": {},  
    "callcount": 18  
  }  
}
```

З кожним викликом змінюється параметр: "calls": 18

Параметр "keys" показує весь перелік параметрів глобального контексту

```
"keys": {  
  "rsstojson": {}, // клас, що працює з rss стрічками  
  "callcount": 18 // кількість викликів  
}
```

## flow context - контекст потоку

Існують аналогічні функції, як і для глобального контексту

- `flow.get(..)` : get a flow-scoped context property
- `flow.set(..)` : set a flow-scoped context property
- `flow.keys(..)` : return a list of all flow-scoped context property keys Для тестування створемо на вході один параметр, а при підготовці відповіді - прочитаємо його.

У запиті додамо:

```
// запис параметру в flow context  
flow.set( 'flw-orig-req', msg.payload );
```

при підготовці відповіді

```
msg.payload.flowprm = flow.get( 'flw-orig-req' );
```

В прикладі Response видно, що оригінальний запит зберігся у параметрі **flowprm**

```
{  
  "xto": "ivan",  
  "xfrom": "petro",  
  "xsubj": "read the book",  
  "xmsg": "Hi I am here!",  
  "calls": 20,  
  "keys": {  
    "rsstojson": {},  
    "callcount": 20  
  },  
  "flowprm": {  
    "xto": "ivan",  
    "xfrom": "petro",  
    "xsubj": "read the book",  
    "xmsg": "Hi I am here!"  
  }  
}
```

