

Tree - единый AST чтобы править всеми

Спикер \Дмитрий Карловский
Место \PiterJS #47
Время 2020-05-20

План

- Проанализировать популярные текстовые форматы данных 
- С нуля разработать новый формат без недостатков 
- Показать примеры применения нового формата 

Форматы

Формат

Пример XML

```
<!DOCTYPE svg
    PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"
>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg version="1.1" xmlns="http://www.w3.org/2000/svg">
    <circle r="30" cx="50" cy="50" fill="orange" />
</svg>
```

Пример JSON

```
{  
  "name": "example",  
  "version": "1.0.0",  
  "description": "example package",  
  "main": "index.js",  
  "repository": "https://example.org",  
  "author": "anonymous",  
  "license": "MIT"  
}
```

Пример YAML

```
Date: 2001-11-23 15:03:17 -5
User: ed
Fatal:
    Unknown variable "bar"
Where:
    file: TopClass.py
    line: 23
    code: |
        x = MoreObject("345\n")
```

Пример TOML

```
[servers]  
  
[servers.alpha]  
ip = "10.0.0.1"  
dc = "eqdc10"  
  
[servers.beta]  
ip = "10.0.0.2"  
dc = "eqdc10"
```

Пример Tree

spoiler

Модели данных

- Какие данные мы можем записать и прочитать без бубна? 
- Как записывать данные не вписывающиеся в модель? 

Модель XML

- NodeList
- Element Node (
)
- Attribute Node (tabindex="1")
- Text Node (Hello, World!)
- CDATA Node (<![CDATA[...]]>)
- Processing Instruction Node (<? ... ?>)
- Comment Node (<!-- ... -->)
- Document Node
- Document Type Node (<!DOCTYPE html>)

Недостатки модели XML

```
<panel>
  <head>Вы уверены?</head>
  <body>
    <button>Да</button>
    <button>Нет</button>
  </body>
</panel>
```

Расширяемость модели XML

```
<xsl:stylesheet
    version="1.0"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="/">
        <html>
            <head>
                <link rel="stylesheet" href="web.css" />
            </head>
            <body>
                <xsl:apply-templates select="*" />
            </body>
        </html>
    </xsl:template>

</xsl:stylesheet>
```

Модель JSON

- Null
- Boolean
- Number
- String
- Array
- Dictionary

Недостатки модели JSON

```
{           [  
  "foo": 777,  
  "bar": 666  
}           [ "foo" , 777 ],  
                  [ "bar" , 666 ]  
}
```

Нерасширяемость модели JSON

```
{  
  "$or": [  
    {  
      "sex": "female",  
      "age": { "$gt": 16 },  
    },  
    {  
      "hobby": {  
        "$regex": "\b(?:java|type)script\b"  
      }  
    }  
  ]  
}
```

Модель YAML

- !!null
- !!bool
- !!int
- !!float
- !!str
- !!timestamp
- !!seq
- !!map
- Anchor & Alias
- Document
- TypeTags

Расширяемость модели YAML

```
--- !!omap
- Mark McGwire: 65
- Sammy Sosa: 63
- Ken Griffy: 58
```

Модель TOML

- Boolean
- Integer
- Float
- String
- DateTime
- Array
- Dictionary

Модель Tree

- Struct Node
- Data Node

Расширяемость модели

	XML	JSON	YAML	TOML	Tree
Расширяемость	✓	✗	✓	✗	✓
Число паттернов	90	30	210	90	10

Удобочитаемость

	XML	JSON	YAML	TOML	Tree
Удобочитаемость	✗	✗	✓	✓	✓
Синтаксис	✗	✗	✓	✓	✓

Удобочитаемость XML

Привет, Алиса!

Как дела?

Не могла бы ты принести мне сейчас кофе?

```
<message>
  <greeting>
    Привет, <a href="http://example.org/user/alice">Алиса</a>!
  </greeting>
  <body>
    <s>Как дела?</s><br/>
    Не могла бы ты принести мне
    <time datetime="1979-10-14T12:00:00.001-04:00">сейчас</time>
    кофе?
  </body>
</message>
```

Удобочитаемость JSON

```
{ "greetings": "Привет, Алиса!\nКак дела?\nНе могла бы ты принести мне кофе?"}
```

Строгость

	XML	JSON	YAML	TOML	Tree
Однозначный синтаксис	✓	✓	✗	✓	✓

Нестрогость YAML

```
a: true # boolean  
b: tru # string  
c: (-: # error  
d: :- ) # string
```

Экранирование

- Нужно отличать конструкции формата от собственно данных 😊
- Желательно не терять в наглядности данных 😎
- Желательно не пере усложнять редактирование 😂

Экранирование в XML

```
foo > 0 && foo < 10      <code>foo &gt; 0 &amp;& foo &lt; 10</code>
```

Экранирование в JSON

```
"/\"[\s\S]*"/      "\\"[\s\S]*\""
```

Экранирование в YAML

- 5 типов строк 😊
- 4 модификатора обработки пробелов 😢

Экранирование в Tree

Нет 

Минификация

- Читаемое форматирование много весит 🐘
- Компактное форматирование плохо читается 💀

Минификация XML

```
<users>
  <user>
    <name>Alice</name>
    <age>20</age>
  </user>
</users>
```

```
<!-- 13% less -->
<users><user><name>Alice</name><age>20</age></user></users>
```

Минификация JSON

```
{ // 30% less
  "users": [
    {
      "name": "Alice",
      "age": 20
    }
  ]
}
```

```
{"users": [{"name": "Alice", "age": 20}]}
```

Минификация Tree

Нет 

Статистика по минификации

	XML	JSON	YAML	TOML	Tree
Читаемый	195%	140%	125%	110%	100%
Минифицированный	170%	101%	-	-	-

Священные войны

- Табы или пробелы? 
- 2 или 4 пробела? 
- возврат каретки нужен? 
- выравнивание делаем? 
- правила линтера/форматтера? 
- при сохранении/коммите/пуше? 

Скорость обработки

serialization: `foo\bar` => `"foo\\bar"`

parsing: `"foo\\bar"` => `foo\bar`

Координаты ошибки

	XML	JSON	YAML	TOML	Tree
--	-----	------	------	------	------

Поточная обработка

	XML	JSON	YAML	TOML	Tree
Поточная обработка	✗	✗	✓	✓	✓

Формат Tree

- Простота синтаксиса 
- Никакого экранирования 
- Никаких вольностей 
- Никакой минификации 
- Минимальный размер 
- Гарантированная читаемость 
- Поточная обработка 
- Точные координаты узлов 

Просто tree-узел

house

Список tree-узлов

house

roof

wall

door

window

floor

Вложение tree-узлов

```
house
  roof
  wall
  door
  window
  floor
```

Глубокая tree-иерархия

```
house
  roof
  wall
    door
    window
      glass
  floor
```

Один дома

street

house

wall

door

window

street house wall

window

door

Сырые данные

\Любые данные \(^_^\)/

Многострочные данные

```
\  
\Тут   
\    очень   
\        МНОГО   
\            строк 
```

Разные типы узлов

```
user
  name \Jin
  age \35
  hobby
    \kendo 🏜
    \dance 💃
    \role play 🎭
  default
```

Языки основанные на форматах

Формат

Языки

Язык grammar.tree

```
tree .is .optional .list_of line          data .is .sequence  
line .is .sequence                      data_prefix  
    .optional indent                     .optional .list_of .byte  
    .optional nodes                     .except new_line  
new_line  
  
nodes .is .sequence                      special .is .any_of  
    .optional .list_of struct           new_line  
    .optional data                     data_prefix  
    .with_delimiter space             indent  
                                     space  
  
struct .is .list_of .byte                new_line .is .byte \0A  
    .except special                   indent .is .list_of .byte \09  
                                     data_prefix .is .byte \5C  
                                     space .is .list_of .byte \20
```

Язык grammar.tree vs EBNF

```
tree .is .optional .list_of line          tree = { line };

line .is .sequence
    .optional indent
    .optional nodes
    new_line

nodes .is .sequence
    .optional .list_of struct
    .optional data
    .with_delimiter space
```

```
line = [ indent ],
[ nodes ],
new_line;

nodes = data |
        struct,
{ space , struct },
[ space , data ];
```

Язык xml.tree vs XML

```
! doctype html
html
meta @ charset \utf-8
link
  @ href \web.css
  @ rel \stylesheet
script @ src \web.js
body
  h1 \Procter & Gamble
```

```
<!doctype html>
<html>
<meta charset="utf-8" />
<link href="web.css" rel="stylesheet" />
<script src="web.js"></script>

<body>
  <h1>Procter & Gamble</div>
</body>

</html>
```

Язык json.tree vs JSON

```
* user * {  
    name \Jin  
    age 35  
    hobby /  
        \kendo 🏜  
        \dance 💃  
    home \C:\users\jin\  
}  
  
"user": {  
    "name": "Jin",  
    "age": 35,  
    "hobby": [  
        "kendo 🏜",  
        "dance 💃",  
    ],  
    "home": "C:\\users\\jin\\"  
}
```

Расширения json.tree

```
*  
# \If disabled will be used platform specific delimiters  
# \CRLN on windows and LN on others  
unix_delimiters true  
  
{  
"unix_delimiters#1": "If disabled will be used platform specific delimit  
"unix_delimiters#2": "CRLN on windows and LN on others",  
"unix_delimiters": true,  
}  
}
```

Язык view.tree vs TypeScript

```
$my_details $mol_view
sub /
    <= Pager $mol Paginator
        value?val <=> page?val 0

class $my_details extends $mol_view {

    sub() { return [ this.Pager() ] }

    @ $mol_mem Pager() {
        const Pager = new $molPaginator
        Pager.value = val => this.page( val )
        return Pager
    }

    @ $mol_mem page( val = 0 ) {
        return val
    }

}
```

API

Формат

Языки

API

JSON AST

```
{                                {  
  "user": {}                      "type" : "Object",  
}                                "children" : [  
                           {  
                             "type" : "Property",  
                             "key" : {  
                               "type": "Identifier",  
                               "value": "user"  
                             }  
                             "value": {  
                               "type": "Object",  
                               "children": []  
                             }  
                           }  
                         ]  
}
```

AST Tree

user

name \Jin

age 35

hobby

\kendo 

\dance 

\role play 

user

name \Jin

age 35

hobby

\kendo 

\dance 

\role play 

Свойства узла Tree

```
interface $mol_tree2 {  
    readonly type: string  
    readonly value: string  
    readonly kids: $mol_tree2[]  
    readonly span: $mol_span  
}
```

```
interface $mol_span {  
    readonly uri: string  
    readonly row: number  
    readonly col: number  
    readonly length: number  
}
```

Производные Tree узлы

```
interface $mol_tree2 {  
    struct : ( type , kids )=> $mol_tree2  
    data : ( value , kids )=> $mol_tree2  
    list : ( kids )=> $mol_tree2  
    clone : ( kids )=> $mol_tree2  
}
```

Сообщения об ошибках в Tree

```
const config_path = './config.tree'
const config_text = fs.readFileSync( config_path )
const config = $mol_tree2.fromString( config_text , config_path )
const password = config.select( 'server' , 'auth' , 'password' , '' )

if( !auth( password.text() ) ) {
    // AuthError: Wrong password
    // \default
    // ./config.tree#5:3-11
    throw password.error( 'Wrong password' , AuthError )
}
```

Поддержка редакторами

-  VSCode
-  Atom
-  Sublime
-  SynWrite

Поддержка языками

-  TypeScript
-  D

Итоги

	XML	JSON	YAML	TOML	Tree
--	-----	------	------	------	------

Идеи

- Запросы к СУБД
- Логирование
- Общение консольных утилит
- LISP-подобный язык
- Универсальный AST

sql.tree - запросы к СУБД

```
select
  from $users
  fetch
    @name
    @phone
    @photo *
    @uri
    @width
    @height
  where or
    and
      @sex = female
      @age > 16
    @hobby ~ \b(?:java|type)script\b
```

```
select
  from $users
  fetch *
  where @hobby ~
    word-edge
    or
      \java
      \type
      \script
    word-edge
```

Логи доступа

```
2020-05-01 17:31:00:137 WARNI Record com.orientechnologies.orient.core.storage.impl.local.paginated.wal.cas.0EmptyWALRecord{lsn=0LogSequenceNumber{segment=7, position=18}} will be skipped during data restore [OLocalPaginatedStorage]
2020-05-01 17:31:00:139 INFO 1 operations were processed, current LSN is 0LogSequenceNumber{segment=7, position=18} last LSN is 0LogSequenceNumber{segment=8, position=18} [OLocalPaginatedStorage]
2020-05-01 17:31:00:146 WARNI Record OFuzzyCheckpointStartRecord{lsn=0LogSequenceNumber{segment=7, position=8210}} com.orientechnologies.orient.core.storage.impl.local.paginated.wal.OFuzzyCheckpointStartRecord{lsn=null, previousCheckpoint=0LogSequenceNumber{segment=6, position=18}} will be skipped during data restore [OLocalPaginatedStorage]
2020-05-01 17:31:00:148 WARNI Record com.orientechnologies.orient.core.storage.impl.local.paginated.wal.OFuzzycCheckpointEndRecord{lsn=0LogSequenceNumber{segment=7, position=8248}} will be skipped during data restore [OLocalPaginatedStorage]
2020-05-01 17:31:00:149 WARNI Record com.orientechnologies.orient.core.storage.impl.local.paginated.wal.cas.0EmptyWALRecord{lsn=0LogSequenceNumber{segment=8, position=18}} will be skipped during data restore [OLocalPaginatedStorage]
2020-05-01 17:31:00:173 INFO  Storage data recover was completed [OLocalPaginatedStorage]
2020-05-01 17:31:00:655 INFO  Storage 'plocal:C:\proj\mam\hyoo\api\-\orient\databases/OSystem' is opened under OrientDB distribution : 3.0.30 - Veloce (build 355adf5e012545012ea2ef7682bcb95f77b0934e, branch 3.0.x) [OLocalPaginatedStorage]
2020-05-01 17:31:01:909 INFO  Listening binary connections on 0.0.0.0:2424 (protocol v.37, socket=default) [OServerNetworkListener]
2020-05-01 17:31:01:926 INFO  Listening http connections on 0.0.0.0:2480 (protocol v.10, socket=default) [OServerNetworkListener]
2020-05-01 17:31:01:943 INFO  Installing dynamic plugin 'orientdb-etl-3.0.30.jar'... [OServerPluginManager]
2020-05-01 17:31:01:969 INFO  Installing dynamic plugin 'orientdb-neo4j-importer-plugin-3.0.30-dist.jar'... [OServerPluginManager]
2020-05-01 17:31:01:992 INFO  Installing dynamic plugin 'orientdb-studio-3.0.30.zip'... [OServerPluginManager]
2020-05-01 17:31:02:024 INFO  Installing dynamic plugin 'orientdb-teleporter-3.0.30.jar'... [OServerPluginManager]
```

access.log.tree - структурированные логи

```
193.34.12.132 - - [2011-10-20T12:46:08+04:00] GET /nin-jin/slides/edit/master/tree/readme.md HTTP/1.1 200 4435
193.34.12.132 - - [2011-10-20T12:46:09+04:00] GET /nin-jin/slides/edit/master/tree/readme.html HTTP/1.1 404 4435
```

access

```
ip \193.34.12.132
time \2011-10-20T12:46:08+04:00
method \GET
uri \/nin-jin/slides/edit/master/tree/readme.md
protocol \HTTP/1.1
response \200
size \4435
```

tree-tools - CLI утилиты обработки деревьев

```
> cat access.log.tree | pick ip time method uri | table
```

```
\193.34.12.132 2011-10-20T12:46:08+04:00    GET /index.html
\193.34.12.132 2011-10-20T12:46:10+04:00    GET /index.css
\193.34.12.132 2011-10-20T12:46:20+04:00    GET /index.js
```

```
> cat access.log.tree | filter time >= 2019-09 | pick ip uri | table
```

```
\193.34.12.132 /index.html
\193.34.12.132 /index.css
\193.34.12.132 /index.js
```

tree как протокол общения

```
> git log
```

```
commit
```

```
  message \$mol_style: TS@3.9 compatibility
  sha \b1a8f07c839604d0d34430a186246f0c1f71e628
  date \2020-05-15T23:24:32+0300
  author \nin-jin <sairi-na-tenshi@ya.ru>
```

```
commit
```

```
  message \$mol_regexp: concurrent parse ability
  sha \be1abfa50542728dd5c156517ea31f469e7fb4d4
  date \2020-05-15T23:03:30+0300
  author \nin-jin <nin-jin@ya.ru>
```

```
> git log | pick date message | table
```

\2020-05-15T23:24:32+0300	\\$mol_style: TS@3.9 compatibility
\2020-05-15T23:03:30+0300	\\$mol_regexp: concurrent parse ability

```
(func $fact (param $x i64) (result i64)
  (if $x (result i64)
    (i64.eqz
      (local.get $x))
    (then
      (i64.const 1)))
    (else
      (i64.mul
        (local.get $x)
        (call $fact
          (i64.sub
            (local.get $x)
            (i64.const 1)))))))
```

wasm.tree - ассемблер без мишуры

```
func          00
  name $fact      61
  param $x i64     73
  result i64       6d
  body switch      01
    test i64.eqz local.get $x 00
    then i64.const 1 00
    else i64.mul   00
      local.get $x .
      call $fact i64.sub .
      local.get $x .
      64.const 1 .
```

jack.tree - LISP без скобочек

jack

 \$mol_jack

```
import wasm
tree func $fact
  > $x #8
  < #8 switch
    test is-zero $x
    then #8 1
    else mul
      $x
      $fact sub
        $x
        #8 1
```

Упраздняя LLVM

compile pipelines:

```
jack.tree => wasm.tree =====> bin.tree  
jack.tree => wasm.tree => arm.tree => bin.tree  
any-dsl.tree => jack.tree => wasm.tree => arm.tree => bin.tree
```

optimization middlewares:

```
jack.tree => jack.tree  
wasm.tree => wasm.tree  
arm.tree => arm.tree
```

Единый AST чтобы править всеми

code =(P)=> loader =(P)=> compiler =(SP)=> bundler =(SP)=> terser =(S)=> bun

P - Parse

S - Serialize

code =(P)=> loader =====> compiler =====> bundler =====> terser =(S)=> bun

Куда пойти, куда податься

- Эти слайды: [!\[\]\(93082c4cf4bb94ad18d54a6e75db819c_img.jpg\) nin-jin/slides/tree](https://nin-jin.github.io/slides/tree/#slide=75)
- Всё о Tree: [!\[\]\(7e49094497f81e4ac66f9e771a5ec93b_img.jpg\) nin-jin/tree.d](https://nin-jin.github.io/tree.d)
- Чат о языках: [!\[\]\(4f16ceaa796b638a379834431060ad28_img.jpg\) lang_idioms](https://t.me/lang_idioms)