



Пользовательская автоматизация профессиональных веб-приложений на Lua

Александр Гладыш
@agladysh

Рязанский IT Workshop
Сентябрь 2018

Профессиональные приложения

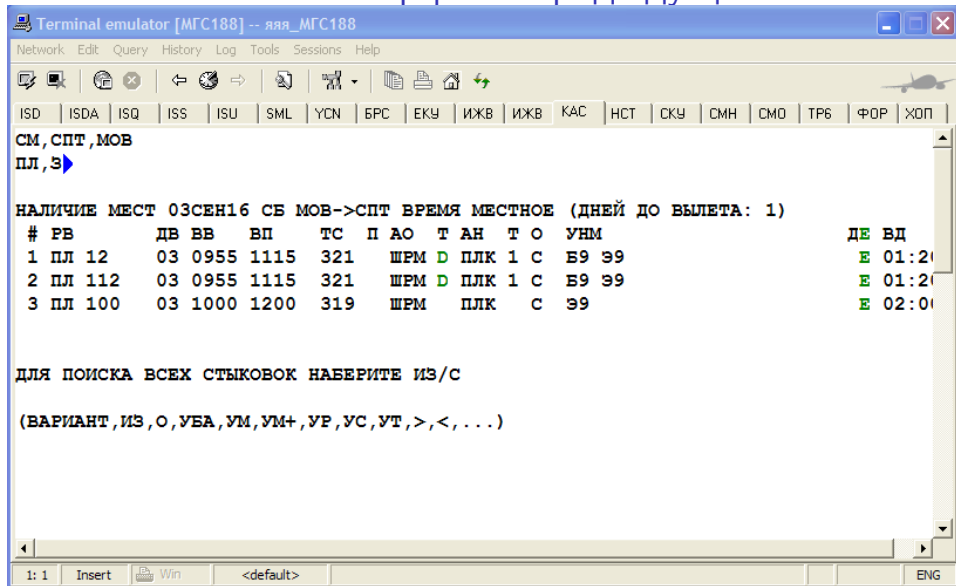
Профессиональное приложение:

- ▶ Для профессионалов, глубоко владеющих предметной областью
- ▶ Которые в большинстве своём не айтишники

Конкретика

- ▶ ТАИС разрабатывает профессиональное ПО для гражданской авиации
- ▶ Глубокая и обширная предметная область с длинной историей
- ▶ Требуется профессионального обучения пользователей

Пользовательский интерфейс предыдущего поколения



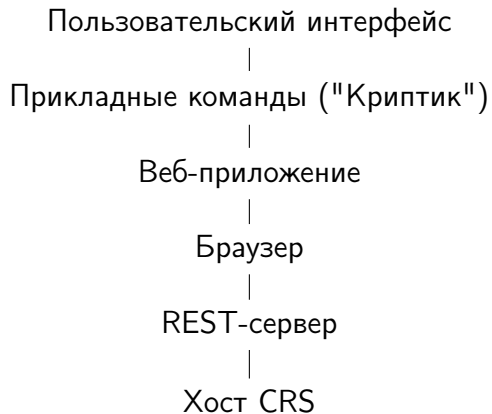
Перспективный пользовательский интерфейс

The screenshot displays a web application for flight status monitoring. The header shows the flight number 'AV 03DEC SVOSVX' and a '12 сек' (12 seconds) timer. The main section is titled 'НАЛИЧИЕ МЕСТ' (Availability of Seats) and lists 13 flights. Each flight entry includes a flight number, aircraft type, departure time, arrival time, and a status indicator. The status indicators are color-coded: green for 'OK' and red for 'Problem'. The flights are sorted by departure time.

№	Flight	Aircraft	Class	Departure	Arrival	Status
6	FV-3236	AIRBUS A320	3 ДЕК	10:00 SVOC	15:00 SVXC	OK
7	FV-4580	320	3 ДЕК	10:00 SVOC	14:00 SVX	OK
8	FV-9001	AIRBUS A320	3 ДЕК	10:00 SVOC	13:00 SVX	Problem
9	FV-9002	AIRBUS A320	3 ДЕК	10:00 SVOC	13:00 SVX	Problem
10	FV-9003	AIRBUS A320	3 ДЕК	10:00 SVOC	13:00 SVX	Problem
11	FV-9004	AIRBUS A320	3 ДЕК	10:00 SVOC	13:00 SVX	Problem
12	FV-9005	AIRBUS A320	3 ДЕК	10:00 SVOC	13:00 SVX	Problem
13	FV-9997	320	3 ДЕК	10:00 SVOC	14:00 SVXC	OK

The interface also includes a sidebar with 'Журнал' (Journal) and 'Избранное' (Favorites) sections, and a bottom status bar with technical details.

Многослойная система



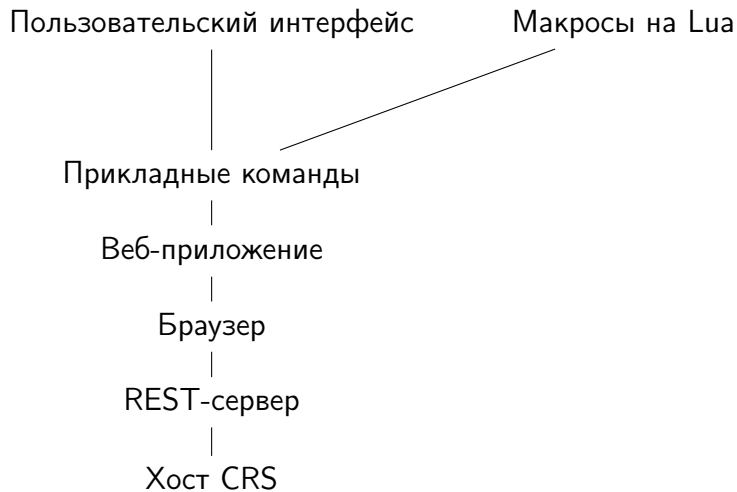
Задачи

- ▶ Автоматизация редких но сложных операций пользователя-эксперта
- ▶ В дальнейшем — удобное API для более широкого круга пользователей

Почему макросы на клиенте?

- ▶ Запускать код пользователя на сервере — головная боль
- ▶ Особенно если уже есть система пользовательских команд
- ▶ Даже если её нет — у вас же есть серверное HTTP-API, правда?


Многослойная система с макросами



Почему Lua?

- ▶ Lua легче освоить непрограммистам
- ▶ JavaScript тяжело изолировать от "кишок" проекта
- ▶ Lua легко изолировать
- ▶ Сам факт использования другого языка делает дизайн API чище

Внешний вид: редактор скриптов

 SCRIPTS SHOW MASSCNL.LUA

examples/avearliest.lua

examples/masscnl.lua

examples/pnr.lua

init.lua

lua-nucleo/strict.lua

main.lua

twl.lua

localfile.lua

masscreate.lua

masscnl.lua

New file

masscnl.lua

```
1 local twt = require 'twl'
2
3 local DATN = "1DEC2018"
4 local DATK = "31DEC2018"
5 local AK = "FV"
6 local FLIGHTNOS = {
7   "9001";
8   "9002";
9   "9003";
10  "9004";
11  "9005"
12  ...
13 }
14 local RULE = "OTM"
15
16 local commands = { }
17
18 for _, FLIGHTNO in ipairs(FLIGHTNOS) do
19   commands[#commands + 1] = "SKD "..AK.."-"..FLIGHTNO.." REA "..DATN.."-"..DATK.."
20 end
21
22 twt(table.concat(commands, ';'))
23
```

Внешний вид: результат выполнения

SKD FV LOG 194507

22 сек

Журнал

Избранное

↶ FV-9005 ОТМЕНА ПЕРИОДА

ОПРЕДЕЛЕНА
26 СЕН 18 17:29:00

ПАРАМЕТРЫ ПЕРИОДА

С 3 ДЕК 2018 ПО 31 ДЕК 2018

☒ ПН 1 ☒ ВТ 2 ☒ СР 3 ☒ ЧТ 4 ☒ ПТ 5 ☒ СБ 6 ☒ ВС 7 ☐ Q

ПАРАМЕТРЫ ПУБЛИКАЦИИ

☒ ПРОПУСКАТЬ АДНОС ☐ ВЕРНУТЬ БИЛЕТЫ ☐ РЕВАЛИДИРОВАТЬ БИЛЕТЫ ☐ НЕ МЕНЯТЬ СТАТУС СЕГМЕНТОВ

ОТМЕНЕН 1 ПЕРИОД:

▼ 03 ДЕК — 31 ДЕК
ЕЖЕДНЕВНО

26 СЕН 2018 17:29:00

УЧАСТКИ	САЛОНЫ	ПИТАНИЕ	ОГРАНИЧЕНИЯ	КОДШЕР
10:00 → 13:00 SVO ⁰ 01:00 SVX	БИЗНЕС 12 мест C		НЕ НАЗНАЧЕНО	НЕ НАЗНАЧЕНО
A320 / A320X127240	ЭКОНОМ 240 мест Y N T U			

▶ ПРОТОКОЛ

▶ ПРАВИЛА ЗАЩИТЫ ЗАКАЗОВ

Закрыть

VL4.2-NST-4825-G007B772 - VL3.6-NST-232-G0412442 - 1681167825584887C7446C2815616540

Помощь

SKD FV LOG 194507

☆ SKD FV LOG 194506

☆ SKD FV LOG 194505

☆ SKD FV LOG 194504

☆ SKD FV LOG 194503

☆ LUA REQUIRE(MAS...

☆ SCRIPTS SHOW MAS...

☆ SCRIPTS SHOW MAS...

☆ SCRIPTS

☆ AV 03DEC SVOSVX

Lua в браузере: основные варианты

- ▶ lua.vm.js
- ▶ fengari

См. также: Sailor MVC

Наш стек

- ▶ lua.vm.js
- ▶ API для выполнения команд и анализа результатов
- ▶ Маленькая самописная библиотека для VFS
- ▶ ACE (brace)
- ▶ webpack

Немного кода: Пример макроса

```
local twt = require 'twt'

local DATN, DATK = "23JUL2018", "31DEC2018"
local AK, FLIGHTNOS = "R3", { "401"; "756" } -- ...

local commands = { }
for _, FLIGHTNO in ipairs(FLIGHTNOS) do
  -- SKD R3-401 REA 23JUL2018-31DEC2018 !EXEC
  commands[#commands + 1] = "SKD " .. AK .. "-" .. FLIGHTNO
    .. " REA " .. DATN .. "-" .. DATK .. " !EXEC"
end

twt(table.concat(commands, ';'))
```

Немного кода: обёртка над VM

```
var LuaVM = require('lua.vm.js');

var execute = function(code) { return this.lua_.execute(code); };
var provideModule = function(name, value) {
    this.lua_._G.get("package").get("loaded").set(name, value);
};

var Vm = function() { this.lua_ = new LuaVM.Lua.State(); };

Vm.prototype = {
    execute: execute,
    provideModule: provideModule,
    constructor: Vm
};
module.exports = Vm;
```


Немного кода: пример модуля Lua на JS

```
vm.provideModule("tw.t.terminal", {  
  alert: function() {  
    var msg = this;  
    alert("Lua says:\n\n" + msg);  
  },  
  
  log: function() {  
    var args = (arguments.length === 1)  
      ? [ arguments[0] ]  
      : Array.apply(null, arguments)  
    ;  
    args.unshift(this); // First argument passed as this  
    log.log.apply(log, args);  
  }  
});
```

Немного кода: запуск Lua

```
vm.execute([  
  "assert(loadstring(",  
    "assert(require('twl.vfs'):get('init.lua'),  
      'missing init.lua'),",  
      "'@init.lua'",  
    ")))()",  
].join("\n"));
```

Немного кода: init.lua, работа с модулями

```
local vfs = require 'twl.vfs'
local searchers = package.searchers
-- Keep package.preload handler intact
package.searchers = { searchers[1] }
package.searchers[#package.searchers + 1] = function(filename)
    if not filename:find("%.lua$") then
        filename = filename .. ".lua"
    end
    local file = vfs:get(filename)
    if not file then
        return "\ncan't find the module"
    end
    return assert(loadstring(file, "@" .. filename))
end
end
```

Что дальше?

- ▶ Технически — Lua в браузере отлично работает и выполняет свои задачи.
- ▶ Главное — это удобство использования, а это уже вопрос дизайна API.

Вопросы?

Alexander Gladyshev
@agladyshev

t.me/luainmoscow