



# Пользовательская автоматизация профессиональных веб-приложений на Lua

Александр Гладыш  
@agladysh

Lua in Moscow  
Сентябрь 2016

# Профессиональные приложения

Профессиональное приложение:

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

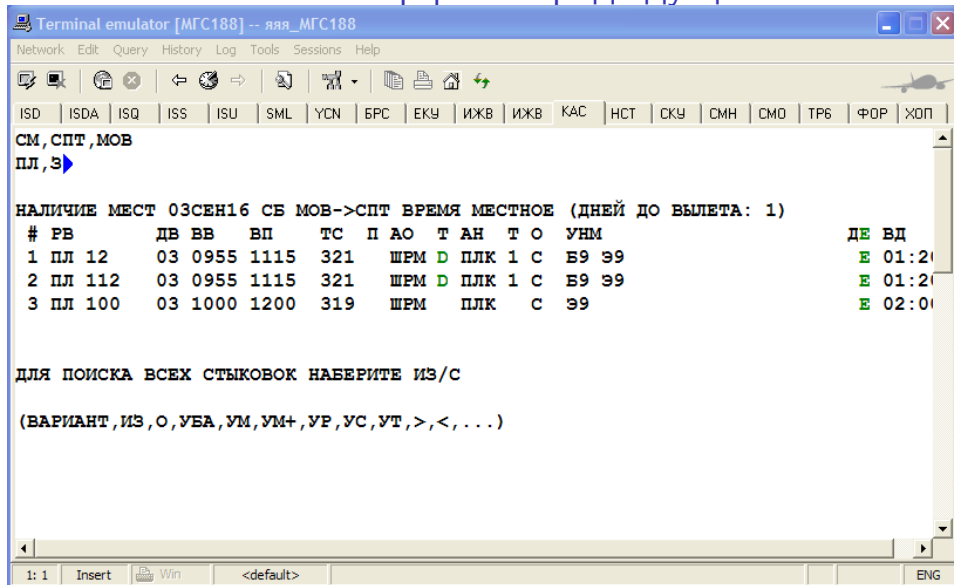
Примеры:

- ▶ CAD'ы
- ▶ Приложения для управления бизнес-логикой "больших" систем
- ▶ Бэкофисы сложных проектов
- ▶ и так далее

# 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

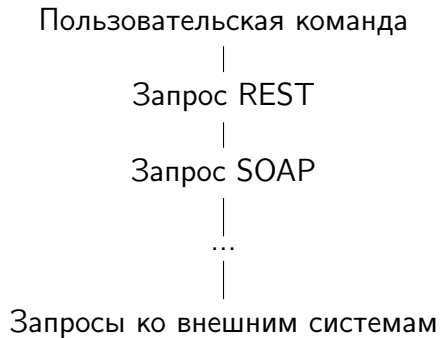
The interface displays a list of flight options for the route ST. PETERSBURG → MOSCOW on 3 SEP. The left sidebar contains navigation links: Terminal, My profile, and Info. The top header shows the flight status AV LED MOW 3SEP. The main content area lists 9 flight options, each with a flight number, class, departure time, arrival time, and a list of aircraft types. The right sidebar includes a Log button, a Favorites button, and a MAIN button.

Flight Number	Class	Departure Time	Arrival Time	Aircraft
1 SU-037	3 SEP	LED <sup>1</sup> 03:50	SVO <sup>0</sup> 05:05	BUS J7 C7 D7 I7 Z7 ECO Y7 B7 M7 U7 P7 K7 H7 L7 Q7 T6
2 SU-035	3 SEP	LED <sup>1</sup> 05:10	SVO <sup>0</sup> 06:25	BUS J7 C7 D4 I3 Z3 ECO Y7 B7 M7 U7 P7 K7 H7 L7 Q7
3 SU-033	3 SEP	LED <sup>1</sup> 06:15	SVO <sup>0</sup> 07:30	BUS J7 C7 D7 I7 Z7 ECO Y7 B7 M7 U7 P7 K7 H7 L7 Q7
4 SU-6026	3 SEP	LED 06:25	VKO 07:55	BUS J7 C6 D6 I5 Z3 ECO Y7 B7 M7 U7 P7 K7 H7 L7
5 SU-6022	3 SEP	LED 08:00	VKO 09:30	BUS J7 C7 D6 I5 Z4 ECO Y7 B7 M7 U7 P7 K7 H7 L7 Q7 T7 E3
6 SU-003	3 SEP	LED <sup>1</sup> 08:15	SVO <sup>0</sup> 09:30	BUS J7 C7 D7 I7 Z7 ECO Y7 B7 M7 U7 P7 K7 H7 L7 Q2
7 SU-6012	3 SEP	LED 09:10	VKO 10:40	ECO Y7 B7 M7 U7 P7 K7 H7 L7
8 SU-007	3 SEP	LED <sup>1</sup> 09:25	SVO <sup>0</sup> 10:40	BUS J7 C7 D7 I7 Z7 ECO Y7 B7 M7 U7 P7 K7 H7 L7 Q6
9 SU-011	3 SEP	LED <sup>1</sup> 10:40	SVO <sup>0</sup> 11:55	BUS J7 C7 D7 I6 Z6 ECO Y7 B7 M7 U7 P7 K7
10 SU-013	3	LED <sup>1</sup>	SVO <sup>0</sup>	BUS J7 C7 D7 I7 Z7

# Многослойная система



# Поток сообщений



# 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?

# Почему Lua?

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

# Внешний вид: редактор скриптов

The screenshot shows a web-based code editor for a Lua script. The interface is divided into several sections:

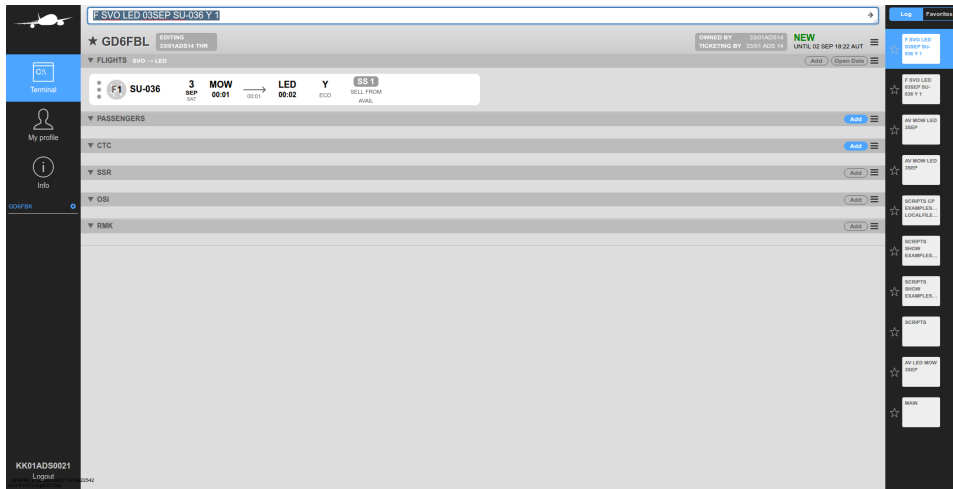
- Left Sidebar:** Contains a file explorer with a list of files: `examples/avearliest.lua`, `examples/brt.lua`, `init.lua`, `lua-nucleo/brt.lua`, `main.lua`, `test.lua`, and `localfile.lua`. Below the list are icons for 'Terminal', 'My profile', and 'Info'.
- Top Header:** Displays the title 'SCRIPTS CP EXAMPLES/AVEARLIEST.LUA LOCALFILE.LUA' and a 'Log' button.
- Main Editor:** The central area where the Lua code is written. The code is for a function `localfile.lua` that finds the earliest flight. It includes comments in Russian and Lua code for handling flight data, sorting, and formatting.
- Right Sidebar:** Contains a 'Log' button and a 'Favorites' section. Below these are several buttons labeled 'SCRIPTS CP EXAMPLES...', 'SCRIPTS SHOW EXAMPLES...', 'SCRIPTS', 'AVIATED MORE', and 'MAIN'.

```
1 local twt = require 'twt'
2
3 -- TODO: Move these globals to requireable modules
4 -- Checks: globals avearliest twt assert declare
5 local pick_earliest_flight = function(flights)
6   local flights = {}
7   for i = 1, #flights do
8     flights[i] = flights[i]
9   end
10  if #flights == 0 then
11    return false
12  end
13  table.sort(
14    flights,
15    function(lhs, rhs) return lhs.departure.date < rhs.departure.date end
16  )
17  return flights[1]
18 end
19
20 local months = {
21   "JAN", "FEB", "MAR", "APR", "MAY", "JUN",
22   "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"
23 }
24 local date = function(date)
25   return date:gsub("%d/%d/%d", "%d/%d/%d").gsub("%d", "%d")
26 end
27
28 local fformat = function(flight, rbd, seats)
29   return "
30     .. flight.departure.point.point
31     .. flight.arrival.point.point
32     .. date(flight.departure.date)
33     .. flight.carrier.iataCode .. "-" .. flight.flight
34     .. rbd
35     .. seats
36   "
37 end
38
39 declare 'avearliest'
40 avearliest = function(query, rbd, seats)
41   twt('av .. query')next(function()
42     -- TODO: Brille
43     -- Make sure that the RPC reply format is fully supported here.
44     local flights = assert(twt.flights)
45     local direct = assert(pick_earliest_flight(flights.direct))
46     local reverse = pick_earliest_flight(flights.reverse)
47     local fdirect = fformat(direct, rbd, seats)
48     local freverse = reverse
49     local freverse = fformat(reverse, rbd, seats)
50     return "
51       .. fdirect .. freverse
52     "
53   end)
54   twt(fdirect .. freverse)
55 end
56
57 return avearliest
58
```

# Внешний вид: команда

```
LUA AVEARLIEST("MOW LED 03SEP", "Y", "1")
```

## Внешний вид: результат выполнения



# Lua в браузере: варианты

- ▶ lua.vm.js
- ▶ moonshine.js
- ▶ lua5.1.js
- ▶ starlight
- ▶ ... и многие другие.

См. также: Sailor MVC.

# Наш стек

- ▶ lua.vm.js
- ▶ Маленькая самописная библиотека для VFS
- ▶ ACE (brace)
- ▶ webpack

## Немного кода

```
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;
```



## 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