

IsFusion . , , ( ) . , .

:

- EXEC - .
- EVAL - IsFusion. , run, .
- EVAL ACTION - IsFusion. \$ ( 1).

:

- **HTTP**

7651, -( ), , -.

URL, , :

- EXEC - http:// :/exec?action=<>. action .
- EVAL - http:// :/eval?script=<>. script , , BODY.
- EVAL ACTION - http:// :/eval/action?script=<>. script , , BODY.

( &p=<>), (BODY). , , URL ( ), BODY.

BODY, , (FILE, PDFFILE ..). . , application, , ( application/abc abc). application/null **NULL**.

BODY , , , . **NULL**.

**System.headers[TEXT]**, , , - .

, , , &return=<>. , . , , **NULL** .

(FILE, PDFFILE ..), , , . , application/<>.

**WRITE**.

, **NULL**, null ( application/null), .

, , text/plain. **NULL** .

**System.headersTo[TEXT]** . , , - .

/ **BODY**

BODY multipart/\* application/x-www-form-urlencoded, . **BODY** .

, , :

- returnmultitype=bodyurl - application/x-www-form-urlencoded, , .
- - multipart/mixed, .

, http HTTP ( , , , )

*Stateful API*

API, REST API, , , , , ( , ), , , , , &session=<>, <>- . , , , , , ( ), \_close ( &session=0\_close).

HTTP cookie, / cookie stateful http- ( , , , , HttpClient Java ..)

( ) , .

http- , . :

- - "Authorization: Basic <credentials>" .
- - :
  - **Authentication.getAuthToken[]** c . ( ). : <http://localhost/exec?action=getAuthToken>.
  - , "Authorization: Bearer <token>" ( JWT, )

*Form API*

, API (JSON API), . stateful API ( HTTP- , , ..), API / , :

- **JavaScript**

JavaScript npm- @lsfusion/core.

API . js-, :

- js-, js-. . js-, , list js-( ). js- ., js- ( js- ) value - , , , js- , , - .
- , ( )js-, :
- :
  - **GRID** - js- list js- .
  - **PANEL, TOOLBAR** - js-
- -js- .

, , ( ).

- create - . :
  1. setState - . , - ( , - , - ), ( , , ). React, , React- setState : updateState => this.setState(updateState).
  2. baseUrl - url - lsFusion - , 'https://demo.lsfusion.org/hockeystats'.
  3. formData - . name ( { name: "MainForm"} ), script ( script:"OBJECTS i=Invoice PROPERTIES (i) date, stock")
- change - . :
  1. setState - .
  2. changes - js-, , . js- js-, , js- / list, . , , value , , , , . change(setState, {game:{value:[30], hostGoals: 40, guestGoals:30}}), 40 30, , id 30. js- (js- ), . change(setState, {game: {doSmthWithGame : true}})
  3. currentState - js- . . UX , , , ( ), , currentState , (render) , .
- close - . :
  1. setState - .
- formCreated - ( state). boolean. :
  1. state - js-
- numberOfPendingRequests - . long. :
  1. state - js-

, / ( , ). Form API, **NEW DELETE** / **NEW DELETE** ( , change(setState, {game : {NEW:true}})). Form API logMessage, ( ).

, stateful form api http- . http- ( -), , , ( ). Form api - .

*Rest API (Python)*

```

import json
import requests
from requests_toolbelt.multipart import decoder

lsfCode = ("run(INTEGER no, DATE date, FILE detail) {\n"
"    NEW o = FOrder {\n"
"        no(o) <- no;\n"
"        date(o) <- date;\n"
"        LOCAL detailId = INTEGER (INTEGER);\n"
"        LOCAL detailQuantity = INTEGER (INTEGER);\n"
"        IMPORT JSON FROM detail TO detailId, detailQuantity;\n"
"        FOR imported(INTEGER i) DO {\n"
"            NEW od = FOrderDetail {\n"
"                id(od) <- detailId(i);\n"
"                quantity(od) <- detailQuantity(i);\n"
"                price(od) <- 5;\n"
"                order(od) <- o;\n"
"            }\n"
"        }\n"
"        APPLY;\n"
"        EXPORT JSON FROM price = price(FOrderDetail od), id = id(od) WHERE order(od) == o;\n"
"        EXPORT FROM orderPrice(o), exportFile();\n"
"    }\n"
"}")

order_no = 354
order_date = '10.10.2017'
order_details = [dict(id=1, quantity=10),
dict(id=2, quantity=15),
dict(id=5, quantity=4),
dict(id=10, quantity=18),
dict(id=11, quantity=1),
dict(id=12, quantity=3)]

order_json = json.dumps(order_details)

url = 'http://localhost:7651/eval'
payload = {'script': lsfCode, 'no': str(order_no), 'date': order_date,
'detail': ('order.json', order_json, 'text/json')}

response = requests.post(url, files=payload)
multipart_data = decoder.MultipartDecoder.from_response(response)

sum_part, json_part = multipart_data.parts
sum = int(sum_part.text)
data = json.loads(json_part.text)

#####

print(sum)
for item in data:
    print('{0:3}: price {1}'.format(int(item['id']), int(item['price'])))

#####
# 205
# 4: price 5
# 18: price 5
# 3: price 5
# 1: price 5
# 10: price 5
# 15: price 5

```