

Api Mtender

# Table of Contents

HTTP verbs.....	1
Endpoints .....	2
Get Citizen .....	2
Public Organization.....	3
Private Organization.....	7
Get IBAN List.....	11
Get Budget .....	12
Check Budget.....	14
Register Contract .....	15
Get Contract Queue .....	18
Confirm Contract .....	20
Register log .....	21
Get log.....	24

# HTTP verbs

RESTful notes tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

Verb	Usage
GET	Used to retrieve a resource
POST	Used to create a new resource

RESTful notes tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

RESTful notes tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

# Endpoints

The index provides the entry point into the service.

## Get Citizen

### Url:

`https://[<SERVER_DOMAIN>]/api/v1/citizen/get?idnp={idnp}`

### Parameters

Parameter	Description
<code>idnp</code>	Person IDNP

### Entity description:

Get Person by IDNP

### HTTP Request

```
GET /api/v1/citizen/get?idnp=423654897456 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json; charset=UTF-8
```

### HTTP Response

#### Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 15:03:14 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000
```

```
{
  "IDNP": "423654897456",
  "FirstName": "NICU",
  "LastName": "ANDREI",
  "SecondName": "",
  "BirthDate": "1989-05-16T00:00:00.000Z",
  "BirthDateStatus": {}
}
```

### Response fields

Parameter	Type	Description
FirstName	string	First Name
LastName	number	Last name
SecondName	number	Second Name
BirthDate	date	Date of birth
BirthDateStatus	number	Status of the date of birth

### Sample

#### Curl

```
curl -X GET "http://127.0.0.1:8800/api/v1/citizen/get?idnp=423654897456" -H "accept: application/json; charset=UTF-8"
```

## Public Organization

#### Url:

[https://\[<SERVER\\_DOMAIN>\]/api/v1/organization/public?idno={idno}](https://[<SERVER_DOMAIN>]/api/v1/organization/public?idno={idno})

#### Parameters

Parameter	Description
idno	Organization IDNO

## Entity description:

Get Contracting Authority by IDNO

## HTTP Request

```
GET /api/v1/organization/public?idno=1008400111879 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json; charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 14:56:46 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000

{
  "IDNO": "1008400111879",
  "FiscalCode": "1008400111879",
  "Name": "Societatea cu Răspundere Limitată TEST SYSTEMS",
  "ShortName": "TEST SYSTEMS S.R.L.",
  "LegalFormCode": "530",
  "PropertyFormCode": "2",
  "RegistrationDate": "2010-09-18T00:00:00.000Z",
  "Address": {
    "Country": {},
    "Region": "CHIȘINĂU CENTRU",
    "Locality": "mun. Chișinău",
    "LocalityCode": {},
    "Street": "Stefan Cel Mare",
    "StreetCode": {},
    "House": "5",
    "Block": {},
    "Flat": {}
  },
  "Activities": {
    "Activity": [
      {
        "ActivityCode": "51900",
        "ActivityType": "0",
        "OrderNumber": "1"
      }
    ]
  }
}
```

```

    },
    {
      "ActivityCode": "52480",
      "ActivityType": "0",
      "OrderNumber": "2"
    },
    {
      "ActivityCode": "51840",
      "ActivityType": "0",
      "OrderNumber": "3"
    },
    {
      "ActivityCode": "72500",
      "ActivityType": "0",
      "OrderNumber": "4"
    },
    {
      "ActivityCode": "72200",
      "ActivityType": "0",
      "OrderNumber": "5"
    },
    {
      "ActivityCode": "64204",
      "ActivityType": "0",
      "OrderNumber": "6"
    }
  ]
},
"Founders": {
  "Founder": {
    "PersonType": "1",
    "PersonShortData": {
      "IDNP": "423654897456",
      "LastName": "NICU",
      "FirstName": "ANDREI"
    }
  }
},
"Conductors": {
  "Conductor": {
    "PersonalData": {
      "IDNP": "423654897456",
      "LastName": "NICU",
      "FirstName": "ANDREI"
    },
    "RoleCode": "18"
  }
},
"ChefAccount": {
  "PersonalData": {
    "IDNP": "2005498433030",

```

```

    "LastName": "ANDRU ELENA",
    "FirstName": {}
  }
}
}

```

## Response fields

Parameter	Type	Description
IDNO	string	IDNO
FiscalCode	number	Tax code
Name	number	Name
ShortName	date	Abbreviated name
LegalFormCode	number	Organizational-legal form
PropertyFormCode	number	Property form
RegistrationDate	number	Date of registration
Address	AddressData	Legal address
Activities	ActivitiesData[]	Activity List
Founders	FounderData[]	List of founders
Conductors	ConductorData[]	List of administrators
ChefAccount	ChefAccountData[]	List of Accountants

### AddressData

Parameter	Type	Description
Country	string	Country
Region	string	Region
Locality	string	Locality
LocalityCode	string	Locality Code
Street	string	Street
StreetCode	string	Street Code
House	string	House
Block	string	Block
Flat	string	Flat

### ActivitiesData

Parameter	Type	Description
ActivityCode	number	Activity Code
ActivityType	number	Activity Type
OrderNumber	number	Order Number



## FounderData

Parameter	Type	Description
PersonType	number	Type of founder
PersonShortData	PersonShortData	The founder's personal data

## PersonShortData

Parameter	Type	Description
IDNP	number	IDNP
FirstName	string	First Name
LastName	number	Last name

## ConductorData

Parameter	Type	Description
RoleCode	number	Administrator role in the organization
PersonalData	PersonShortData	The founder's personal data

## ChefAccount

Parameter	Type	Description
PersonalData	PersonShortData	The founder's personal data

## Sample

### Curl

```
curl -X GET "http://127.0.0.1:8800/api/v1/organization/public?idno=1008400111879" -H "accept: application/json;charset=UTF-8"
```

## Private Organization

### Url:

[https://\[<SERVER\\_DOMAIN>\]/api/v1/organization/private?idno={idno}](https://[<SERVER_DOMAIN>]/api/v1/organization/private?idno={idno})

### Parameters

Parameter	Description
idno	Organization IDNO

## Entity description:

Get Contracting Authority by IDNO

## HTTP Request

```
GET /api/v1/organization/public?idno=1008400111879 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json; charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 14:56:46 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000

{
  "IDNO": "1008400111879",
  "FiscalCode": "1008400111879",
  "Name": "Societatea cu Răspundere Limitată TEST SYSTEMS",
  "ShortName": "TEST SYSTEMS S.R.L.",
  "LegalFormCode": "530",
  "PropertyFormCode": "2",
  "RegistrationDate": "2010-09-18T00:00:00.000Z",
  "Address": {
    "Country": {},
    "Region": "CHIȘINĂU CENTRU",
    "Locality": "mun. Chișinău",
    "LocalityCode": {},
    "Street": "Stefan Cel Mare",
    "StreetCode": {},
    "House": "5",
    "Block": {},
    "Flat": {}
  },
  "Activities": {
    "Activity": [
      {
        "ActivityCode": "51900",
        "ActivityType": "0",
        "OrderNumber": "1"
      }
    ]
  }
}
```

```

    },
    {
      "ActivityCode": "52480",
      "ActivityType": "0",
      "OrderNumber": "2"
    },
    {
      "ActivityCode": "51840",
      "ActivityType": "0",
      "OrderNumber": "3"
    },
    {
      "ActivityCode": "72500",
      "ActivityType": "0",
      "OrderNumber": "4"
    },
    {
      "ActivityCode": "72200",
      "ActivityType": "0",
      "OrderNumber": "5"
    },
    {
      "ActivityCode": "64204",
      "ActivityType": "0",
      "OrderNumber": "6"
    }
  ]
},
"Founders": {
  "Founder": {
    "PersonType": "1",
    "PersonShortData": {
      "IDNP": "423654897456",
      "LastName": "NICU",
      "FirstName": "ANDREI"
    }
  }
},
"Conductors": {
  "Conductor": {
    "PersonalData": {
      "IDNP": "423654897456",
      "LastName": "NICU",
      "FirstName": "ANDREI"
    },
    "RoleCode": "18"
  }
},
"ChefAccount": {
  "PersonalData": {
    "IDNP": "2005498433030",

```

```

    "LastName": "ANDRU ELENA",
    "FirstName": {}
  }
}

```

## Response fields

Parameter	Type	Description
IDNO	string	IDNO
FiscalCode	number	Tax code
Name	number	Name
ShortName	date	Abbreviated name
LegalFormCode	number	Organizational-legal form
PropertyFormCode	number	Property form
RegistrationDate	number	Date of registration
Address	AddressData	Legal address
Activities	ActivitiesData[]	Activity List
Founders	FounderData[]	List of founders
Conductors	ConductorData[]	List of administrators
ChefAccount	ChefAccountData[]	List of Accountants

### AddressData

Parameter	Type	Description
Country	string	Country
Region	string	Region
Locality	string	Locality
LocalityCode	string	Locality Code
Street	string	Street
StreetCode	string	Street Code
House	string	House
Block	string	Block
Flat	string	Flat

### ActivitiesData

Parameter	Type	Description
ActivityCode	number	Activity Code
ActivityType	number	Activity Type
OrderNumber	number	Order Number

## FounderData

Parameter	Type	Description
PersonType	number	Type of founder
PersonShortData	PersonShortData	The founder's personal data

## PersonShortData

Parameter	Type	Description
IDNP	number	IDNP
FirstName	string	First Name
LastName	number	Last name

## ConductorData

Parameter	Type	Description
RoleCode	number	Administrator role in the organization
PersonalData	PersonShortData	The founder's personal data

## ChefAccount

Parameter	Type	Description
PersonalData	PersonShortData	The founder's personal data

## Sample

### Curl

```
curl -X GET "http://127.0.0.1:8800/api/v1/organization/private?idno=1008400111879" -H "accept: application/json;charset=UTF-8"
```

## Get IBAN List

### Url:

[https://\[<SERVER\\_DOMAIN>\]/api/v1/budget/iban?idno={idno}&cpv={cpv}&year={year}](https://[<SERVER_DOMAIN>]/api/v1/budget/iban?idno={idno}&cpv={cpv}&year={year})

### Parameters

Parameter	Description
idno	Organization IDNO
cpv	CPV Code
year	Year

## Entity description:

Get iban list by IDNO, CPV and YEAR

## HTTP Request

```
GET /api/v1/budget/iban?idno=1006601243057&cpv=30000000-9&year=2018 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json; charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
< HTTP/1.1 200 OK
< Server: nginx
< Date: Mon, 14 Jan 2019 14:39:27 GMT
< Content-Type: text/html; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< Strict-Transport-Security: max-age=15768000
<
* Connection #0 to host 127.0.0.1 left intact
{"getIbanList_response":{"iban":["MD67JDUPEJ347483D11090PD","MD16KDPFUP222400A01225GD"]}}
```

### Response fields

Parameter	Type	Description
iban	string	Organization iban list

## Sample

### Curl

```
curl -X GET "http://127.0.0.1:8800/api/v1/budget/iban?idno=1006601243057&cpv=30000000-9&year=2018" -H "accept: application/json; charset=UTF-8"
```

## Get Budget

### Url:

[https://\[<SERVER\\_DOMAIN>\]/api/v1/budget/get?iban={iban}&year={year}](https://[<SERVER_DOMAIN>]/api/v1/budget/get?iban={iban}&year={year})

## Parameters

Parameter	Description
iban	Organization IBAN
year	Year

### Entity description:

Get budget by IBAN, YEAR

## HTTP Request

```
GET /api/v1/budget/get?iban=MD67JDUPEJ347483D11090PD&year=2018 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json;charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 14:45:56 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000

{
  "sum": "29850.00"
}
```

### Response fields

Parameter	Type	Description
sum	number	Organization budget by year

## Sample

### Curl

```
curl -X GET
"http://127.0.0.1:8800/api/v1/budget/get?iban=MD67JDUPEJ347483D11090PD&year=2018" -H
"accept: application/json;charset=UTF-8"
```

## Check Budget

### Url:

[https://\[<SERVER\\_DOMAIN>\]/api/v1/budget/check?iban={iban}&year={year}&sum={sum}](https://[<SERVER_DOMAIN>]/api/v1/budget/check?iban={iban}&year={year}&sum={sum})

### Parameters

Parameter	Description
iban	Organization IBAN
sum	Blocking amount
year	Year

### Entity description:

Check amount budget by IBAN, YEAR

### HTTP Request

```
GET /api/v1/budget/check?iban=MD67JDUPEJ347483D11090PD&year=2018&sum=500 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json;charset=UTF-8
```

### HTTP Response

#### Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 14:49:12 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000
```

```
{
  "result": "1"
}
```



## Response fields

Parameter	Type	Description
result	number	Result check amount budget

## Sample

### Curl

```
curl -X GET
"http://127.0.0.1:8800/api/v1/budget/check?iban=MD67JDUPEJ347483D11090PD&year=2018&sum=500" -H "accept: application/json;charset=UTF-8"
```

## Register Contract

### Url:

[https://\[<SERVER\\_DOMAIN>\]/api/v1/citizen/get?idnp={idnp}](https://[<SERVER_DOMAIN>]/api/v1/citizen/get?idnp={idnp})

### Constraints

#### Header (header)

Parameter	Type	Description
id_dok	string	System contract id
nr_dok	string	Contract number
da_dok	date	Contract date
suma	double	Contract amount
kd_val	string	Currency code
pkd_fisk	string	Tax Code of the Contractor
pkd_sdiv	string	Number of contractor subdivision
pname	string	Name of contractor
bkd_fisk	string	The beneficiary of the contract
bkd_sdiv	string	Beneficiary subdivision
bname	string	Name of the beneficiary
desc	string	Description of the contract
reg_nom	string	The registration number in the treasury
reg_date	date	Date of registration in the treasury

Parameter	Type	Description
achiz_nom	string	Contract number at the Public Procurement Agency
achiz_date	date	The date of registration of the contract with the Public Procurement Agency
avans	string	Prepaid expense
da_expire	date	Date of expiration of the contract
c_link	string	Link to contract in .pdf format

#### Beneficiary (benef)

Parameter	Type	Description
id_dok	string	System contract id
bbic	string	BIC Code
biban	number	IBAN Code

#### Details (details)

Parameter	Type	Description
id_dok	string	System contract id
suma	number	Contract amount
piban	number	IBAN Code
byear	number	Year

#### Entity description:

Get Person by IDNP

#### HTTP Request

```
POST /api/v1/contract/register HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json;charset=UTF-8
Content-Type: application/json
Content-Length: 604
```

```
{
  "header":{
    "id_dok":"id123123",
    "nr_dok":"123",
    "da_dok":"2019-01-13T14:16:59+00:00",
    "suma":10,
    "kd_val":"123",
    "pkd_fisk":"string",
    "pkd_sdiv":"string",
    "pname":"string",
    "bkd_fisk":"string",
    "bkd_sdiv":"string",
    "bname":"string",
    "desc":"string",
    "reg_nom":"123",
    "reg_date":"2019-01-01T14:15:59",
    "achiz_nom":"123",
    "achiz_date":"2019-01-13T14:15:59+00:00",
    "avans":5,
    "da_expire":"2020-01-14T14:15:59+00:00",
    "c_link":"string"
  },
  "benef":[
    {
      "id_dok":"id123123",
      "bbic":"string",
      "biban":0
    }
  ],
  "details":[
    {
      "id_dok":"id123123",
      "suma":10,
      "piban":"string",
      "byear":2019
    }
  ]
}
```

## HTTP Response

## Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 14:23:59 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000
```

```
{
  "id_dok": "string",
  "num_row": "1"
}
```

## Response fields

### Sample

#### Curl

```
curl -X POST "http://[<SERVER_DOMAIN>]/api/v1/contract/register" -H "accept: application/json; charset=UTF-8" -H "Content-Type: application/json" -d "{ \"header\": { \"id_dok\": \"id123123\", \"nr_dok\": \"123\", \"da_dok\": \"2019-01-13T14:16:59+00:00\", \"suma\": 10, \"kd_val\": \"123\", \"pkd_fisk\": \"string\", \"pkd_sdiv\": \"string\", \"pname\": \"string\", \"bkd_fisk\": \"string\", \"bkd_sdiv\": \"string\", \"bname\": \"string\", \"desc\": \"string\", \"reg_nom\": \"123\", \"reg_date\": \"2019-01-01T14:15:59\", \"achiz_nom\": \"123\", \"achiz_date\": \"2019-01-13T14:15:59+00:00\", \"avans\": 5, \"da_expire\": \"2020-01-14T14:15:59+00:00\", \"c_link\": \"string\" }, \"benef\": [ { \"id_dok\": \"string\", \"bbic\": \"string\", \"biban\": 0 } ], \"details\": [ { \"id_dok\": \"string\", \"suma\": 0, \"piban\": \"string\", \"byear\": 0 } ] }"
```

## Get Contract Queue

### Get

#### Url:

[http://\[<SERVER\\_DOMAIN>\]/api/v1/contract/queue?status={status}](http://[<SERVER_DOMAIN>]/api/v1/contract/queue?status={status})

#### Parameters

Parameter	Description
<code>status</code>	Contract status

## Entity description:

List of contracts by status in the Treasury Information System.

## HTTP Request

```
GET /api/v1/contract/queue?status=3001 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json; charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 Jan 2019 14:01:06 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=15768000

{
  "contract": [
    {
      "id_dok": "id123",
      "id_hist": "1006",
      "status": "3001",
      "st_date": "2019-01-11T16:15:39.000+02:00",
      "reg_nom": "string",
      "reg_date": "2019-01-11+02:00",
      "descr": {}
    },
    {
      "id_dok": "id144",
      "id_hist": "1007",
      "status": "3001",
      "st_date": "2019-01-11T16:16:31.000+02:00",
      "reg_nom": "string",
      "reg_date": "2019-01-11+02:00",
      "descr": {}
    }
  ]
}
```

## Response fields

Parameter	Type	Description
<code>id_dok</code>	string	System contract id
<code>id_hist</code>	string	Change the status of the contract to the Treasury Information System
<code>status</code>	number	Status
<code>st_date</code>	date	Date of change of status
<code>reg_nom</code>	string	The registration number in the treasury
<code>reg_date</code>	string	Date of registration in the treasury
<code>descr</code>	string	Description of the contract

## Sample

### Curl

```
curl -X GET "http://[<SERVER_DOMAIN>]/api/v1/contract/queue?status=3001" -H "accept: application/json;charset=UTF-8"
```

## Confirm Contract

### Url:

`http://[<SERVER_DOMAIN>]/api/v1/contract/confirm`

### Constraints

Parameter	Type	Description
<code>id_dok</code>	string	Contract id
<code>desc</code>	string	Description (Optional)

### Entity description:

Set confirm status by contract id

## HTTP Request

```
POST /api/v1/contract/confirm?id_dok=id144 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json; charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
connection: keep-alive
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Mon, 14 Jan 2019 14:04:12 GMT
server: nginx
strict-transport-security: max-age=15768000
transfer-encoding: chunked

{
  "id_dok": "id123",
  "num_row": "1"
}
```

### Response fields

Parameter	Type	Description
<code>id_dok</code>	string	Document number in the system
<code>num_row</code>	number	Number of entries

## Sample

### Curl

```
curl -X POST "http://[<SERVER_DOMAIN>]/api/v1/contract/confirm?id_dok=id123" -H
"accept: application/json; charset=UTF-8"
```

## Register log

### Url:

`http://[<SERVER_DOMAIN>]/api/v1/mlog`

## Constraints

Parameter	Type	Description
user_session	string	User session in which context the action happened. This attribute permits to split the action taken by a user if the action is a step of a flow.
amount	string	Amount
event_type	string	The type of the event according to IS definition, which usually represents the action taken that resulted in this event (ex. Created, Authenticated, Deleted, etc.).
documents	string	List Documents (where url = url to document, id = id document in system, hash = hash document)
user_address	string	User IP address or location or any another form which identifies where the user acted from.
event_source	string	Place where the event was generated. Ex: logging class name, IS sub-component, or server name, etc.
legal_entity	string	The legal entity (organization) on behalf of which the action was performed (by a user or automatically). Usually it is the IDNO of the organization.
event_message	string	Free text that describes the event, indexed by MLog for advanced text search.
event_time	string	The moment when the event happened at the source system (i.e. not the time of the logging).

### Entity description:

Register log

### HTTP Request



```
POST /api/v1/mlog HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json;charset=UTF-8
Content-Type: application/json
Content-Length: 268

{
  "event_time": "2019-01-14T23:12:37.334+02:00",
  "event_type": "MTender.Bid.Open",
  "event_source": "string",
  "event_message": "string",
  "legal_entity": "string",
  "user_session": "string",
  "user_address": "string",
  "amount": "string",
  "documents": [
    "url1|id1|hash1"
  ]
}
```

## HTTP Response

### Success HTTP response

```
Http-Code: 200
Server: nginx
Date: Tue, 15 Jan 2019 09:52:59 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Strict-Transport-Security: max-age=31536000

{
  "timestamp": "1547545979199", "UID": "e2c57431-0a53-4b2b-b235-27f4944ed3bf"
}
```

### Response fields

Parameter	Type	Description
<code>timestamp</code>	string	Registration time
<code>UID</code>	string	Registered id from log system

## Sample

### Curl

```
curl -X POST "http://[<SERVER_DOMAIN>]/api/v1/mlog" -H "accept: application/json;charset=UTF-8" -H "Content-Type: application/json" -d "{ \"event_time\": \"2019-01-14T23:12:37.334+02:00\", \"event_type\": \"MTender.Bid.Open\", \"event_source\": \"string\", \"event_message\": \"string\", \"legal_entity\": \"string\", \"user_session\": \"string\", \"user_address\": \"string\", \"amount\": \"string\", \"documents\": [ \"url1|id1|hash1\" ]}"
```

## Get log

### Url:

[http://\[<SERVER\\_DOMAIN>\]/api/v1/mlog](http://[<SERVER_DOMAIN>]/api/v1/mlog)

### Parameters

Parameter	Type	Description
legal_entity	string	Legal entity that performs the search. (Optional)
legal_basis	string	Legal base for search
legal_reason	string	Legal reason for search (Optional)
user	string	IDNP of the user that searches for events. (Optional)
event_time_from	string	Start time for period to search
event_time_to	string	Start time for period to search
filter	string	A list of key/value for the known fields to search. MLog will filter only those events that match these given fields. The format is field1=value1,field2=value2, etc (Optional)
page	string	The page number to be returned in case there are more than 1 page on the results. By default this is considered to be 0 (first page). (Optional)
page_size	string	The chosen page size. Default: 50. (Optional)

## Entity description:

Get logs by start and end date, and filter

## HTTP Request

```
GET
/api/v1/mlog?legal_entity=Test&legal_basis=Test&legal_reason=Test&user=Test&event_time
_from=2019-01-14&event_time_to=2019-01-15 HTTP/1.1
Host: 127.0.0.1:8800
User-Agent: curl/7.63.0
accept: application/json;charset=UTF-8
```

## HTTP Response

### Success HTTP response

```
Http-Code: 200
Server: nginx
Date: Tue, 15 Jan 2019 09:58:14 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Strict-Transport-Security: max-age=31536000
```

```
{"timestamp":"1547546294631","result":[{"user_session":"string","amount":"string","event_type":"MTender.Bid.Open","subject_type":"Organization","documents":["url1|id1|hash1"],"user_address":"string","event_level":"info","event_source":"string","legal_entity":"string","event_message":"string","event_time":"2019-01-14T23:12:37.334+02:00","@uid":"e2c57431-0a53-4b2b-b235-27f4944ed3bf"},{"user_session":"string","amount":"string","event_type":"MTender.Bid.Open","subject_type":"Organization","documents":["url1|id1|hash1"],"user_address":"string","event_level":"info","event_source":"string","legal_entity":"string","event_message":"string","event_time":"2019-01-14T23:12:37.334+02:00","@uid":"219c594b-a8eb-4ab3-943d-128c5bd41225"}],"pagination":{"record-count":2,"page-size":50}}
```

### Response fields

Parameter	Type	Description
<code>user_session</code>	string	User session in which context the action happened. This attribute permits to split the action taken by a user if the action is a step of a flow.
<code>amount</code>	string	Amount

Parameter	Type	Description
event_type	string	The type of the event according to IS definition, which usually represents the action taken that resulted in this event (ex. Created, Authenticated, Deleted, etc.).
subject_type	string	The type of the subject.
documents	string	List Documents (where url = url to document, id = id document in system, hash = hash document)
user_address	string	User IP address or location or any another form which identifies where the user acted from.
event_level	string	Event classifier. Each IS which register events to MLog can use its own definition for this, e.g. relevance (ex. high/medium/low) or impact (warning/critical/fatal), etc.
event_source	string	Place where the event was generated. Ex: logging class name, IS sub-component, or server name, etc.
legal_entity	string	The legal entity (organization) on behalf of which the action was performed (by a user or automatically). Usually it is the IDNO of the organization.
event_message	string	Free text that describes the event, indexed by MLog for advanced text search.
event_time	string	The moment when the event happened at the source system (i.e. not the time of the logging).
uid	string	Log id

## Sample

### Curl

```
curl -X GET  
"http://[<SERVER_DOMAIN>]/api/v1/mlog?legal_entity=Test&legal_basis=Test&legal_reason=  
Test&user=Test&event_time_from=2019-01-14&event_time_to=2019-01-15" -H "accept:  
application/json;charset=UTF-8"
```