UPYA API

API for Last Mile distribution. This API describes all the methods available for interacting with Upya.

Use this API to query your Upya database, validate or make payments, generate codes, create products, contracts or clients and much more.

Upya Manage

All endpoints to interact with your different collections from the Upya Manage platform: clients, contracts, agents, assets, payments, tasks, tickets, etc...

Clients

Endpoints relating to your **clients** collection. Your clients are organized according to their

- **status**: "Pending", "Approved", "Signed", "Rejected" or "Validated"
- **category**: customizable list (settings)
- **stage**: customizable list (settings), representing the client's "progress stage"

Each one of your clients can have multiple *contracts* (see **contracts** section) and multiple *forms* (see **forms** section).

POST Search clients

https://data.upya.io/data/search/clients

Endpoint to search your clients collection. Pass through MongoDb-like queries in the body of your request in the query attribute.

> *Example*: to retrieve all clients updated after the 2018 World Cup final, send the following body:

javascript

```javascript
{
    "query": {
        "updatedAt": {
            "$gte": "5 July 2018"
         }
    }
}
```

You can also use the following MongoDb options: **sort**, **limit** and **select** to optimize your request. See MongoDb documentation for more details.

> *Example*: to get the clientNumbers of the 3 most recently clients send the following request:

javascript

```javascript
{
    "query": {},
    "limit": 3,
    "select": "clientNumber",
    "sort": "updatedAt"
}
```

Finally, to paginate your answer, you can use the MongoDb **skip** option by adding the following attributes to your query: *paginate*, *pageNumber* and *nPerPage*. The skip option overrides any sort value for answer stability reasons.

javascript

```javascript
{
    "query": {
        "profile.lastName": {"$regex":"idane"}
    },
    "paginate": true,
    "pageNumber": 2,
    "nPerPage": 1000
}
```

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
```

```
{
  "query": {
    "clientNumber": "<string>",
    "status": "<string>",
    "category": "<string>",
    "stage": "<string>",
    "signingDate": "<date>",
    "respAgent": "<string>",
    "contact": {
      "mobile": "<number>",
      "email": "<string>",
      "address": "<string>"
    },
    "profile": {
      "firstName": "<string>",
      "lastName": "<string>",
      "gender": "<string>",
      "birthday": "<date>",
      "gps": {
        "longitude": "<string>",
        "latitude": "<string>"
      },
      "village": "<string>",
      "city": "<string>",
      "county": "<string>",
      "district": "<string>",
      "commune": "<string>",
      "province": "<string>",
      "region": "<string>",
      "state": "<string>",
      "country": "<string>"
    }
  }
}
```

POST Count clients

https://data.upya.io/data/count/clients

Endpoint to count the clients matching a given query. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json

Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "clientNumber": "<string>",
    "status": "<string>",
    "category": "<string>",
    "stage": "<string>",
    "signingDate": "<date>",
    "respAgent": "<string>",
    "contact": {
      "mobile": "<number>",
      "email": "<string>",
      "address": "<string>"
    },
    "profile": {
      "firstName": "<string>",
      "lastName": "<string>",
      "gender": "<string>",
      "birthday": "<date>",
      "gps": {
        "longitude": "<string>",
        "latitude": "<string>"
      },
      "village": "<string>",
      "city": "<string>",
      "county": "<string>",
      "district": "<string>",
      "commune": "<string>",
      "province": "<string>",
      "region": "<string>",
      "state": "<string>",
      "country": "<string>"
    }
  }
}
```

GET Get client

https://data.upya.io/data/clients

Endpoint to retrieve information regarding one existing client, specified by its clientNumber

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Accept	application/json

PATH VARIABLES

clientNumber	<string> (Required)

PUT Edit client

https://data.upya.io/data/clients

Endpoint to edit an existing client specified by its clientNumber

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

PATH VARIABLES

clientNumber	<string> (Required)

Body raw (json)

json

```json
{
  "mobile": "<number>",
  "secondaryMobile": "<number>",
  "firstName": "<string>",
  "lastName": "<string>",
  "gender": "<string>",
  "gps": {
    "longitude": "<string>",
    "latitude": "<string>"
  },
  "village": "<string>",
  "county": "<string>",
  "district": "<string>",
  "region": "<string>",
  "neighbour": "<string>",
  "nextOfKin": "<string>",
  "countryCode": "<string>",
  "clientPictureTag": "<string>"
}
```

POST Create clients

https://data.upya.io/data/clients

Endpoint to create list of new clients

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "clientNumber": "<string>",
  "status": "<string>",
  "signingDate": "<date>",
  "respAgent": "<string>",
  "contact": {
    "mobile": "<number>",
    "email": "<string>",
    "address": "<string>"
  },
  "profile": {
    "firstName": "<string>",
    "lastName": "<string>",
    "gender": "<string>",
    "birthday": "<date>",
    "gps": {
      "longitude": "<string>",
      "latitude": "<string>"
    },
    "village": "<string>",
    "city": "<string>",
    "county": "<string>",
    "district": "<string>",
    "commune": "<string>",
    "province": "<string>",
    "region": "<string>",
    "state": "<string>",
    "country": "<string>"
  }
}
```

Contracts

Use these endpoints to interact with your **contracts** collection.

Contracts are organized according to their:

- **onboardingStatus**: "Pending", "Approved", "Rejected" or "Signed
- **status**: "*LOCKED*", "*ENABLED*", "*REPOSSESSED*", "*PAIDOFF*" or "*WRITEOFF*"
- **categories**: a customizable list that you can define in your settings.

POST Search contracts

https://data.upya.io/data/search/contracts

Endpoint to search your contracts collection. Pass through MongoDb-like queries in the body of your request in the query attribute.

> *Example*: to retrieve all contracts updated after the 1998 World Cup final, send the following body:

javascript

```javascript
{
    "query": {
        "updatedAt": {
            "$gte": "12 July 2018"
         }
    }
}
```

You can also use the following MongoDb options: **sort**, **limit** and **select** to optimize your request. See MongoDb documentation for more details.

> *Example*: to get the contractNumbers of the 3 most recently clients send the following request:

javascript

```javascript
{
    "query": {},
    "limit": 3,
    "select": "contractNumber",
    "sort": "updatedAt"
}
```

Finally, to paginate your answer, you can use the MongoDb **skip** option by adding the following attributes to your query: *paginate*, *pageNumber* and *nPerPage*. The skip option overrides any sort value for answer stability reasons.

javascript

```javascript
{
    "query": {
        "totalpaid": {"$gte":1000}
    },
    "paginate": true,
    "pageNumber": 2,
    "nPerPage": 1000
}
```

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "contractNumber": "<string>",
    "clientNumber": "<string>",
    "paygNumber": "<string>",
    "dealName": "<string>",
    "status": "<string>",
    "onboardingStatus": "<string>",
    "nextStatusUpdate": "<string>",
    "lastStatusUpdate": "<string>",
    "respAgent": "<string>",
    "recorder": "<string>",
    "signingDate": "<string>"
  }
}
```

POST Get contract

https://data.upya.io/data/contracts

Endpoint to retrieve information regarding one contract, specified by its contractNumber

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumber": "<string>"
}
```

POST Count contracts

https://data.upya.io/data/count/contracts

Endpoint to count the contracts matching a given query. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "contractNumber": "<string>",
    "clientNumber": "<string>",
    "paygNumber": "<string>",
    "dealName": "<string>",
    "status": "<string>",
    "onboardingStatus": "<string>",
    "signingDate": "<string>",
    "nextStatusUpdate": "<string>",
    "lastStatusUpdate": "<string>",
    "respAgent": "<string>",
    "recorder": "<string>",
    "singingDate": "<string>"
  }
}
```

POST Create contract

https://data.upya.io/data/contracts/create

Endpoint to add one new contract to an existing client

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "clientNumber": "<string>",
  "contractNumber": "<string>",
  "dealName": "<string>",
  "signingDate": "<date>"
}
```

PUT Approve contracts

https://data.upya.io/data/contracts/approve

Endpoint to approve a list of contracts, specified by their contractNumbers

AUTHORIZATION Basic Auth

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumber": "<string>",
  "userNumber": "<string>"
}
```

PUT Reject contracts

https://data.upya.io/data/contracts/reject

Endpoint to reject a list of contracts, specified by their contractNumbers

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
{
  "contractNumber": "<string>",
  "noteToAgent": "<string>",
  "rejectClient": "<string>",
  "userNumber": "<string>"
}
```

PUT Add payments

https://data.upya.io/data/contracts/addPayments

Endpoint to add a payment to given contracts. The query updates the contracts accordingly (totalPaid, debt, status and nextStatusUpdate).

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
{
  "contractNumber": "<string>",
  "amount": "<string>",
  "transactionId": "<string>",
  "ccy": "<string>",
  "date": "<date>",
  "note": "<string>"
}
```

PUT Update acontract

https://data.upya.io/data/contracts/updateOne

Endpoint to "manually" update a contract. Equivalent to the "hardreset" option on the web platform. Updates made via this endpoint override the specified contract object without being linked to any payment.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumber": "<string>",
  "status": "<string>",
  "totalPaid": "<string>",
  "nextStatusUpdate": "<date>",
  "flag": "<string>",
  "categoriesToAdd": "list",
  "categoriesToRemove": "list"
}
```

PUT Write off contracts

https://data.upya.io/data/contracts/writeoff

Endpoint to write-off a list of contracts. For each contract in the list:

- if its status is "ENABLED" or "LOCKED", both the contract and the attached unit(s) are written-off
- if its status is "REPOSSESSED", only the contract is written-off
- if its status is "PAIDOFF", the action fails
- if its status is "WRITEOFF", the contract is un-writtenoff and the unit is restored (unless the contract was repossessed)

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
{
  "contractNumbers": "list"
}
```

PUT Pay off contracts

https://data.upya.io/data/contracts/unlock

Endpoint to pay-off a list of contracts. The following changes will be applied to each contract in the list:

- contract status moved to "PAIDOFF"
- paidOffDate set to today
- nextStatusUpdate removed
- remainingDebt set to 0

An unlock code is sent to each client if the "sendCodeToClients" flag is set to true

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumbers": "list",
  "sendCodeToClients": "Boolean"
}
```

PUT Edit terms

https://data.upya.io/data/contracts/editterms

Endpoint to edit the terms of an existing contract (totalCost, monthly payment, frequency of payments). Use this action to modify the economic terms of a given contrat without actually changing the "deal" under which the contract sits.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumber": "<string>",
  "edits": {
    "totalCost": "<string>",
    "backPay": "<string>",
    "pricingSchedule": "<string>"
  }
}
```

PUT Change deal

https://data.upya.io/data/contracts/changedeal

Endpoint to change the deal linked to an existing contract. The request will update the "totalCost" (and hence the "remainingDebt") of the contract. It will also edit the "pricingSchedule" of the contract to the one of the new deal.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumber": "<string>",
  "dealName": "<string>",
  "dealNumber": "<string>"
}
```

POST Send bonus

https://data.upya.io/data/codes/send-bonus

Use this endpoint to send bonus days for a given contract.

This action will generate an activation code and will update the "nextStatusUpdate" field of the contract. It will however not change the economics of the contract (totalPaid, totalCost, ...).

An event will be logged in the contract history

AUTHORIZATION Basic Auth

Username	<username>

Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "contractNumber": "<string>",
  "numberOfDays": "<string>"
}
```

Assets

Endpoints related to your **assets** collection. Your assets are organized according to their **status**:

- "*DEPLOYED*": unit has been sold
- "*INSTOCK*": unit is in stock and physically with an agent (or warehouse)
- "*PENDING*": unit is on its way from the manufacturer. Unit cannot be scanned or sold yet
- "*DEFECTIVE*": unit is defective and cannot as it stands be sold again

POST Search units

https://data.upya.io/data/search/assets

Endpoint to search your asset collection. Pass through MongoDb-like queries in the body of your request in the query attribute.

You can also use the following MongoDb options: **sort**, **limit** and **select** to optimize your request. See MongoDb documentation for more details.

Finally, to paginate your answer, you can use the MongoDb **skip** option by adding the following attributes to your query: *paginate*, *pageNumber* and *nPerPage*. The skip option overrides any sort value for answer stability reasons.

javascript

```javascript
{
```

```json
{
    "query": {
        "status": "INSTOCK"
    },
    "paginate": true,
    "pageNumber": 2,
    "nPerPage": 1000
}
```

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "serialNumber": "<string>",
    "paygNumber": "<string>",
    "productReference": "<string>",
    "batchNumber": "<string>"
  }
}
```

POST Count units

https://data.upya.io/data/count/assets

Endpoint to count the units matching a given query. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "serialNumber": "<string>",
    "paygNumber": "<string>",
    "productReference": "<string>",
    "batchNumber": "<string>"
  }
}
```

Payments

Endpoints relating to your **payments** collection. Your payments are organized according to their **status**:

- "*ACCEPTED*": payments that have been successfully processed *and* allocated to their target contract
- "*UNASSIGNED*": payments that have been successfully processed *but* that could not be allocated to any contract
- "*FAILED*": payments that have been rejected and that have *not* hit your Mobile Money account
- "*PREACCEPTED*": payments that have been accepted but for which the relevant contract has not been synced yet by the sale agent

POST Search payments

https://data.upya.io/data/search/payments

Endpoint to search your payments collection. Pass through MongoDb-like queries in the body of your request in the query attribute.

You can also use the following MongoDb options: **sort**, **limit** and **select** to optimize your request. See MongoDb documentation for more details.

Finally, to paginate your answer, you can use the MongoDb **skip** option by adding the following attributes to your query: *paginate*, *pageNumber* and *nPerPage*. The skip option overrides any sort value for answer stability reasons.

javascript

```
{
    "query": {
        "date": {"$gte":"01 November 2022"}
    },
    "paginate": true,
    "pageNumber": 2,
    "nPerPage": 1000
}
```

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
{
  "query": {
    "date": "<string>",
    "amount": "<string>",
    "ccy": "<string>",
    "country": "<string>",
    "mobile": "<string>",
    "status": "<string>",
    "paymentCode": "<string>",
    "paymentReference": "<string>",
    "reasonForFailing": "<string>",
    "contractNumber": "<string>"
  }
}
```

POST Count payments

https://data.upya.io/data/count/payments

Endpoint to count the payments matching a given query. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "date": "<string>",
    "amount": "<string>",
    "ccy": "<string>",
    "country": "<string>",
    "mobile": "<string>",
    "status": "<string>",
    "paymentCode": "<string>",
    "paymentReference": "<string>",
    "reasonForFailing": "<string>",
    "contractNumber": "<string>"
  }
}
```

Forms

Forms are created each time data is collected on the ground. Forms can be "**client-linked**" or "**standalone**" (meaning not linked to any client).

Client-linked forms can be "onboarding" forms or forms completed by existing clients.

You can attach forms to contracts, to tasks and to tickets.

POST Search forms

https://data.upya.io/data/search/forms

Endpoint to search your form collection. Pass through MongoDb-like queries in the body of your request in the query attribute.

> *Example*: to retrieve all forms completed for the "Example" questionnaire, send the following body:

javascript

```
{
    "query": {
        "name": "Example"
    }
}
```

You can also use the following MongoDb options: **sort** and **limit** to optimize your request. See MongoDb documentation for more details.

Finally, to **paginate** your answer, you can use the MongoDb **skip** option by adding the following attributes to your query: paginate, pageNumber and nPerPage. The skip option overrides any "sort" input for answer stability reasons.

javascript

```
{
    "query": {
        "name": {"$regex":"Example"}
    },
    "paginate": true,
    "pageNumber": 2,
    "nPerPage": 1000
}
```

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

json

```json
{
  "query": {
    "name": "<string>",
    "score": "<string>",
    "status": "<string>",
    "clientNumber": "<string>",
    "firstName": "<string>",
    "lastName": "<string>"
  }
}
```

POST Count forms

https://data.upya.io/data/count/forms

Endpoint to count the forms matching a given query. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "name": "<string>",
    "score": "<string>",
    "status": "<string>",
    "clientNumber": "<string>",
    "firstName": "<string>",
    "lastName": "<string>"
  }
}
```

POST Pictures SignedUrls

https://data.upya.io/data/forms/signedUrls

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "listOfIds": "<string>"
}
```

Agents

Endpoints relating to your **agents** collection.

POST Search agents

https://data.upya.io/data/search/agents

Endpoint to search the agent collection. Pass through MongoDb-like queries in the body of your request.

Body raw (json)

json

```json
{
```

```
{
  "query": {
    "agentNumber": "<string>",

    "firstName": "<string>",
    "lastName": "<string>",
    "email": "<string>",
    "role": "<string>",
    "n1AgentNumber": "<string>",
    "n2AgentNumber": "<string>",
    "mobile": "<string>",
    "village": "<string>",
    "region": "<string>",
    "district": "<string>",
    "county": "<string>"
  }
}
```

POST Create agents

https://data.upya.io/data/agents

Endpoint to create a list of agents

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
{
  "agentNumber": "<string>",
  "role": "<string>",
  "country": "<string>",
  "contact": {
    "mobile": "<number>",
    "email": "<string>"
  },
  "profile": {
    "firstName": "<string>",

    "lastName": "<string>",
    "gender": "<string>",
    "birthday": "<date>"
  },
  "location": {
    "gps": {
      "longitude": "<string>",
      "latitude": "<string>"
    },
    "village": "<string>",
    "city": "<string>",
    "county": "<string>",
    "district": "<string>",
    "commune": "<string>",
    "province": "<string>",
    "region": "<string>",
    "state": "<string>",
    "country": "<string>"
  }
}
```

PUT Edit agent

https://data.upya.io/data/agents

Endpoint to edit an existing agent specified by its agentNumber

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

PATH VARIABLES

agentNumber	<string> (Required)

Body raw (json)

json

```json
{
  "agentNumber": "<string>",
  "role": "<string>",
  "country": "<string>",
  "contact": {
    "mobile": "<number>",
    "email": "<string>"
  },
  "profile": {
    "firstName": "<string>",
    "lastName": "<string>",
    "gender": "<string>",
    "birthday": "<date>"
  },
  "location": {
    "gps": {
      "longitude": "<string>",
      "latitude": "<string>"
    },
    "village": "<string>",
    "city": "<string>",
    "county": "<string>",
    "district": "<string>",
    "commune": "<string>",
    "province": "<string>",
    "region": "<string>",
    "state": "<string>",
    "country": "<string>"
  }
}
```

Products

Set up endpoints. Your products are the type of goods that your company sells with Upya.

> Example: Samsung S20

Products can be **serialized** or **non-serialized** and **lockable** or **non-lockable**.

Lockable products are products that can be remotely locked if the client doesn't pay for his/her device. The locking mechanism can be either time-based or communication-based.

POST Create new products

https://data.upya.io/data/products/create

Use this endpoint to set up new products

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "productReference": "<string>",
  "name": "<string>",
  "category": "<string>",
  "manufacturer": "<string>"
}
```

POST Get list of products

https://data.upya.io/data/products/search

Endpoint to retrieve information regarding available products

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json

Accept	application/json

Body raw (json)

json

```json
{
  "category": "<string>",
  "manufacturer": "<string>",
  "name": "<string>",
  "productReference": "<string>"
}
```

Deals

Set up endpoints. **Deals** are the different terms under which your company sells its products.

Deals can of different types:

- *PAYGO* - for lockable products
- *INSTALMENTS* - for non-lockable products
- *FULL* - meaning full upfront
- *LOAN* - if you have the loan option set up
- *TOPUP* - if you have the topup option set up

POST Create new deal

https://data.upya.io/data/deals

Endpoint to create a new deal (terms) for a product specified by its productReference.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "type": "<string>",
  "productReference": "<string>",
  "externalId": "<string>",
  "name": "<string>",
  "description": "<string>",
  "costPerItem": "<string>",
  "totalCost": "<string>",
  "ccy": "<string>",
  "backPay": "<string>",
  "noApprovalNeeded": "<boolean>",
  "noQuestionnaireNeeded": "<boolean>",
  "pricingSchedule": "<string>",
  "topUpSchedule": "<string>"
}
```

POST Search deals

https://data.upya.io/data/deals/search

Endpoint to retrieve list of **deals** available given provided criteria. Deals are the terms or payment plans under which you sell your different **products**.

You can also use the following MongoDb options: **sort**, **limit** and **select** to optimize your request. See MongoDb documentation for more details

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
    "query": {
        "type": "<string>",
        "productReference": "<string>",
        "category": "<string>",
        "dealName": "<string>",
        "totalCost":"<string>"
    }
}
```

PUT Edit existing deal

https://data.upya.io/data/deals

Endpoint to edit a given deal specified by its dealNumber.

Editing any numerical value (totalCost, recurring payment, etc) with an economical impact will create a **new deal version**. Editing non-economic fields (dealName, description, etc) will not create a new deal version.

A new deal version only affects future customers picking the edited deal. Existing customers (including pending ones) will not be affected by an edit creating a new deal version.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
    "dealNumber": "<string>",
    "totalCost": "<string>",
    "pricingSchedule": {
        "days": "<string>",
        "days": "<string>",
        "recurring": "<string>",
        "upfront": "<string>",
        "upfrontDays": "<string>"
    },
    "agentNumbersToAdd": "<string>",
    "agentNumbersToRemove": "<string>",
    "eligibleCategories": "<string>",
    "eligibleAssetCategories": "<string>",
    "description": "<string>"
}
```

Tickets

Tickets are "actions" received by *web users*.

POST Create tickets

https://data.upya.io/data/tickets/create

Endpoint to send **tickets** to web app users. Requires to have the "*task*" option activated and to have set up a "*ticket template*" in the Templates >> Set Up section of Manage.

Use an existing **templateNumber** and an existing **contractNumber. Text** is a free text.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
[{
  "templateNumber": "<string>",
  "contractNumber": "<string>",
  "text": "<string>"
}]
```

POST Search tickets

https://data.upya.io/data/search/tickets

Endpoint to search the ticket database. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {}
}
```

Tasks

Tasks are "actions" assigned to mobile app users.

POST Search tasks

https://data.upya.io/data/search/tasks

Endpoint to search the task database. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {}
}
```

POST Create tasks

https://data.upya.io/data/tasks/create

Endpoint to send **tasks** to mobile app users. Requires to have the "*task*" option activated and to have set up a "*task template*" in the Templates >> Set Up section of Manage.

Tasks can be created in bulk.

Attribute	Type	Description
templateNumber	String	Unique identifier of the task being sent to the agent
agentNumber	String	Unique identifier of the agent the task is being assigned to
contractNumber	String	*Optional*: identifier of the contract the task relates to
instructions	String	*Optional:* if populated, overrides the template's intructions

Body raw (javascript)

javascript

```
[{
  "templateNumber": "<string>",
  "contractNumber": "<string>",
  "agentNumber": "<string>",
  "instructions": "<string>"
}]
```

Events

On Upya, any changes to a client, asset or contract are monitored and logged as "events." These events can be forwarded to **webhook** as they occur. Alternatively, you can query your **clientEvents**, **contractEvents** and **assetEvents** database.

POST Search contract events

https://data.upya.io/data/search/contract-events

Endpoint to search the contractEvent database. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```
{
  "query": {
    "type": "<string>",
    "date": "<string>",
    "endDate": "<string>",
    "info": "<string>",
    "additionalInfo": "<string>"
  }
}
```

POST Search asset events

https://data.upya.io/data/search/asset-events

Endpoint to search the assetEvent database. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "type": "<string>",
    "date": "<string>",
    "paygNumber": "<string>",
    "prevPaygNumber": "<string>",
    "prevContractNumber": "<string>",
    "note": "<string>",
    "author": "<string>"
  }
}
```

Communicate

Endpoints to communicate with your clients and agents

POST Send notification

https://data.upya.io/data/communication/notification

Send a pop up message, or "notification", to the mobile app of your agents. There is no charge for sending notifications.

AUTHORIZATION Basic Auth

Body raw (json)

json

```json
{
    "agentNumber":"ABC",
    "message":"Thank you for your hard work!",
    "title":"New message"
}
```

POST Send message

https://data.upya.io/data/communication/message

Send a text message to your clients or agents. Standard SMS charges apply.

> Specify a "*agentNumber*" to send a message to an agent
>
> Specify a "*clientNumber*" to send one to a client

AUTHORIZATION Basic Auth

Body raw (json)

json

```json
{
    "agentNumber":"ABC",
    "clientNumber": "DEF",
    "message":"Welcome!"
}
```

Users

Users log in to the Manage web platform. Their credentials do not work on the mobile app.

POST Get users

https://data.upya.io/data/search/users

Endpoint to get users. Pass through MongoDb-like queries in the body of the request.

You can also use the following MongoDb options: **sort** and **limit** to optimize your request. See MongoDb documentation for more details. **Select** is not available for security reasons.

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
      "role": "<string>",
      "profile.firstName":"<string>",
      "profile.lastName": "<string>",
      "username":"<string>",
      "email":"<string>"
  }
}
```

Optionals

Optionals

POST Search penalties

https://data.upya.io/data/search/penalties

Endpoint to search the penalty database. Requires the dueDates or the financialPenalties option. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "type": "<string>",
    "date": "<string>",
    "amount": "<string>",
    "debt": "<string>",
    "principalRpmt": "<string>",
    "ccy": "<string>",
    "contract": "<string>"
  }
}
```

POST Search due dates

https://data.upya.io/data/search/schedules

Endpoint to search the dueDate database. Requires the dueDates option. Pass through MongoDb-like queries in the body of the request of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "type": "<string>",
    "date": "<string>",
    "amount": "<string>",
    "debt": "<string>",
    "principalRpmt": "<string>",
    "ccy": "<string>",
    "contract": "<string>"
  }
}
```

POST Search commissions

https://data.upya.io/data/search/commissions

Requires the challenge option. Endpoint to search the commission database. Pass through MongoDb-like queries in the body of the request

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "query": {
    "agentNumber": "ABC",
    "contractNumber": "ABC",
    "challengeNumber": "ABC"
  }
}
```

Schemas

List of Upya Models

Client Model

Attribute	Type	Description
clientNumber	String	Unique client identifier
status	String	enum: ["Validated", "Signed", "Approved", "Rejected", "Pending", "Qualifying"]
category	String	enum customizable by Upya user
stage	String	enum customizable by Upya user
profile.firstName	String	Client firstname
profile.lastName	String	Client lastname
profile.gender	String	Client gender
profile.village	String	Client village. Other similar attributes include: city, commune, district, county, region, province, state, country, landmark, address, route, directions, postcode,
contact.countryCode	String	Country code (+251)
contact.mobile	Number	Mobile number excluding country code

Contract model

Attribute	Type	Description
contractNumber	String	Unique contract identifier
client	Object	Mongodb reference of client linked to contract
onboardingStatus	String	enum ["Pending", "Approved", "Signed", "Rejected"]
status	String	enum ["LOCKED", "ENABLED", "PAIDOFF", "REPOSSESSED", "WRITEOFF"]
product	Object	Mongodb reference of product linked to contract
deal	Object	Mongodb reference of deal linked to contract
lockable	Boolean	TRUE for contracts linked to lockable products
totalCost	Number	Total cost of contract
ccy	String	Currency of contract
remainingDebt	Number	Debt remaining on contract

Asset model

Attribute	Type	Description
assetNumber	String	Unique asset identifier (number internal to Upya)
status	String	enum: ["INSTOCK", "DEPLOYED", "DEFECTIVE", "PENDING", "LINKED"]
serialNumber	String	Serial number of unit (unique)
paygNumber	String	PAYGO number of unit (unique). Can be the same or different to serial number
product	Object	Reference to MongoDb's product object
manageBy	String	For lockable products: locking serve provider. Example: Nexus or GLP Link
lockable	Boolean	TRUE for units that can be activated or locked

nonSerialized	Boolean	TRUE for non-serialized items
isPart	Boolean	TRUE for parts

Payment model

Attribute	Type	Description
transactionId	String	Unique transaction Id. Usually the Mobile Money Operator's unique transaction identifier
amount	Number	Value of payment
type	String	Mobile money, Float, Cash, Bank transfer, ...
ccy	String	Currency of payment
reference	String	Reference used by client when making payment
subscriber	Number	Mobile phone having made the payment
status	String	enum: ["ACCEPTED", "UNASSIGNED", "PREACCEPTED", "FAILED"]
contract	Object	MongoDb reference of contract to which payment has been assigned
client	Object	MongoDb reference of client linked to contract

Agent model

Attribute	Type	Description
agentNumber	String	Unique identifier of agent
role	String	enum customizable by Upya client
email	String	Login username of mobile app
profile.firstName	String	Firstname
profile.lastName	String	Lastname
contact.countryCode	String	+251
contact.mobile	Number	Phone number excluding country code
language	String	Language preference for mobile app
n1	Object	MongoDb reference of agent above agent
n2	Object	MongoDb reference of agent above n1
admin.joinedOn	Date	When was the agent hired
admin.leftOn	Date	When did the agent left

Payment Gateway

Payment Gateway endpoints

Use this section to connect to the Upya Payment Gateway.

POST Retrieve payment options for subscriber

```
https://mm.api.upya.io/api/payment/options/:clientIdentifier
```

Endpoint to retrieve payment options for a given mobile number (subscriber) OR for a given payment reference.

Returns a list of contracts associated with the subscriber, if you sent a "subscriber"; returns the details of the identified contract if you sent a "reference".

You cannot send both a "subscriber" and a "reference".

These contract details can then be passed along as suggestions to the customer to reduce input errors

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

PATH VARIABLES

clientIdentifier	<string> (Required)

Body raw (json)

json

```json
{
  "subscriber": "<string>",
  "reference": "<string>"
}
```

POST Validate incoming payment

https://mm.api.upya.io/api/payment/validation

Endpoint to validate incoming payments in a 2-step payment process.

Returns Accepted or Rejected.

If the payment will be accepted, returns additional information regarding client.

If rejected, returns the likely reason why.

AUTHORIZATION Basic Auth

Username	<username>

Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Body raw (json)

json

```json
{
  "reference": "<string>",
  "amount": "<string>",
  "currency": "<string>",
  "subscriber": "<string>",
  "transactionId": "<string>"
}
```

POST Confirm incoming payment

https://mm.api.upya.io/api/payment/confirmation

Endpoint to notify Upya of an already **confirmed** payment.

Returns the message sent to customer.

A payment arriving to that endpoint has already been received by the client and has already been deducted from the end-customer's account.

If the reference is not recognized, the payment will have the "*UNASSIGNED*" status

AUTHORIZATION Basic Auth

Username	<username>
Password	<password>

HEADERS

Content-Type	application/json
Accept	application/json

Accept	application/json

Body raw (json)

json

```json
{
  "reference": "<string>",
  "amount": "<string>",
  "currency": "<string>",
  "subscriber": "<string>",
  "transactionId": "<string>",
  "mno": "<string>"
}
```