

FIRMAO API

Table of contents

1. [Enabling API and authentication](#)
2. [Adding new objects \(POST\)](#)
 - a) [Projects](#)
 - b) [Tasks](#)
 - c) [Products/Services](#)
 - d) [Company](#)
 - e) [Invoices](#)
 - f) [Quotes/Orders/Agreements](#)
 - g) [Contacts](#)
 - h) [Notes](#)
3. [Modifying objects \(PUT\)](#)
 - a) [Product name change](#)
 - b) [Company name change](#)
 - c) [Product's stock level change \(requires enabling 'Direct modification of stock level' in Company settings\)](#)
4. [Getting objects \(GET\)](#)
 - [Pagination](#)
 - [Sorting](#)
 - [Filtering](#)
 - [Response codes](#)
 - a) [Getting projects - projects](#)
 - b) [Getting tasks - tasks](#)
 - c) [Getting products- products](#)
 - d) [Getting companies - customers](#)
 - e) [Getting invoices - transactions](#)
 - f) [Getting invoices for companies - transactions](#)
 - g) [Getting invoice items - transactions](#)
 - h) [Getting quotes/orders - offers](#)
 - i) [Getting contacts - contacts](#)
 - j) [Getting company groups - customergroups](#)
 - k) [Getting notes - salesnotes](#)
 - l) [Getting stock documents - storagedocs](#)
 - m) [Getting files list - documents](#)
 - n) [Getting mails - mails](#)
 - o) [Getting invoice numbering series](#)

5. [Removing objects \(PUT, DELETE\)](#)
6. [Detailed description of creating, editing, finishing and starting work times](#)
7. [Getting PDF files](#)
 - a) [Invoices](#)
 - b) [Quotes/Orders/Agreements](#)
8. [Response codes](#)
9. [Sample PHP scripts](#)
 - a) [Adding a company](#)
 - b) [Getting company's data](#)
 - c) [Adding an invoice](#)
 - d) [Uploading PHP file < 7.0](#)
 - e) [Uploading PHP file >= 7.0](#)
10. [More API options](#)

1. Enabling API and authentication

In order for API to work you have to go to Company settings → Integrations → API, enable API, download the login and generate a password.

All requests need to have an authentication that looks like this:

A header 'Authorization' with value 'Basic' + encoded base63 string username + ':' + password

Example:

Authorization: Basic ai5zbWI0aEBzdGVsbGlzLmJpczoxMjM0NTY=

where ai5zbWI0aEBzdGVsbGlzLmJpczoxMjM0NTY is an encoded base64 string
'j.smith@stellis.bis:123456'

Username - login of the API user in the Firmao system

Password - password of the API user in the Firmao system

2. Adding new objects (POST)

Creation of new objects is done via POST requests and should be directed to address:
`https://system.firmao.net/{organizationID}/svc/v1/{xxx}`

xxx - name of the object type

organizationID - ID of the organization visible in the URL after logging in to Firmao

Data type:

ContentType = application/json; charset=UTF-8

All fields available in objects are described in the help of data import.

Custom fields have keys 'customFields.custom5' and take Strings. Available fields can also be 'previewed' by downloading data with a GET method (except Permission field which is calculated and cannot be entered in a POST/PUT request).

In a response to POST requests we send data in a json format with details about all fields, which were calculated/modified by Firmao. Among others is the ID of a created object
`[{"modification": "CREATE" "objectId": "50007" }]`

Response codes:

API uses standard HTTP errors codes

201 - created correctly

4xx and 5xx - error occurred when creating the object

In a case of an error an intelligible error description appears by default.

For example in a case of an attempt to create a task related to a non-existing project we will get:

Response code: 400.

```
{"errorCode": "entityNotFoundForId", "field": "project", "rejectedValue": "50", "additionalInfo":
```

```
{"id": "50", "cls": "project"}}
```

Always in the Author field an API user will be shown.

Example requests

a) Projects

<https://system.firmao.net/{organizationID}/svc/v1/projects>

POST

```
{
  "description": "Description",
  "teamMembers": [{"id": 6}],
  "restrictedTeamMembers": [],
  "managers": [{"id": 10}],
  "tags": [{"id": 3}],
  "startDate": "2012-07-17T00:00:00+02:00",
  "name": "Sample"
}
```

Response if created correctly

Code 201:

```
{"changelog": [{"deletedWith": "", "modification": "CREATE", "newValuelsDictionary": false, "objectClass": "project", "objectId": 50003, "objectLabel": "Sample", "oldValuelsDictionary": false, "property": ""}]}
```

b) Tasks

<https://system.firmao.net/{organizationID}/svc/v1/tasks>

POST

```
{
  "taskType": "PROJECT",
  "priority": "300",
  "tags": [],
  "responsibleUsers": [{"id": 2}],
  "estimatedHours": 5,
  "plannedEndDateType": "EXACTLY",
  "plannedStartDateType": "EXACTLY",
  "description": "Description",
}
```

```
"plannedStartDate":"2012-07-17T00:00:00+02:00",
"plannedEndDate":"2012-07-18T00:00:00+02:00",
"orderCalculateFromGross":false,
"orderVatPercent":null,
"name":"Sample task",
"status":"WAITING"
}
```

Response if created correctly

Code 201:

```
{"changelog":[{"deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"task","objectId":50017,"objectLabel":"Sample task","oldValuesDictionary":false,"property":""},{deletedWith":"","modification":"CHANGE","newValueClass":"date","newValueDisplay":"2012-07-17T14:00:00.000+02:00","newValuesDictionary":false,"objectClass":"task","objectId":50017,"objectLabel":"Sample task","oldValueClass":"date","oldValueDisplay":"2012-07-17T22:00:00.000Z","oldValuesDictionary":false,"property":"plannedEndDate"}]}
```

c) Products/Services

<https://system.firmao.net/{organizationID}/svc/v1/products>

POST

```
{
  "tags":[],
  "purchasePriceGroup.netPrice":10,
  "purchasePriceGroup.grossPrice":12.3,
  "purchasePriceGroup.vat":23,
  "saleable":true,
  "productCode":"PR1",
  "name":"Sample product"
}
```

Response if created correctly

Code 201:

```
{"changelog":[{"deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"product","objectId":50001,"objectLabel":"Sample product","oldValuesDictionary":false,"property":""}]}
```

d) Company

POST

https://system.firmao.net/{organizationID}/svc/v1/customers

```
{ "bankAccountNumber" : "50 1020 5558 1111 1702 0600 0067",
  "correspondenceAddress.city" : "New York",
  "correspondenceAddress.country" : "USA",
  "correspondenceAddress.county" : "",
  "correspondenceAddress.postCode" : "10001",
  "correspondenceAddress.street" : "143 Parker Ave",
  "customerGroups" : [ ],
  "customerType" : "PARTNER",
  "description" : "Company specializing in computer service",
  "emails" : [ "compservice.firmao@mailinator.com",
    "computers@mailinator.com"
  ], "employeesNumber" : 1,
  "faxNumber" : "22 398 78 87",
  "label" : "COMP-SERVICE",
  "name" : "COMP-SERVICE John Smith",
  "nipNumber" : "7291077843",
  "officeAddress.city" : "New York",
  "officeAddress.country" : "USA",
  "officeAddress.postCode" : "10001",
  "officeAddress.street" : "143 Parker Ave",
  "ownership" : "PRIVATE",
  "phones" : [ "223457426" ],
  "status" : null,
  "tags" : [ ],
  "voivodeship" : null,
  "website" : "www.compservice.bis"
}
```

Response if created correctly

Code 201:


```
{"changelog":[{"deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"customer","objectId":50001,"objectLabel":"COMP-SERVICE","oldValuesDictionary":false,"property":""}]}
```

Response in a case of a duplicate field (ID or TIN):

Kod 409:

```
{"errorCode":"duplicateValueFoundException","objectName":"customer","fieldName":"nipNumber","duplicateValues":[{"lineNumber":-1,"objectId":51,"value":"7291077843"}]}
```

e) Invoices

POST

<https://system.firmao.net/{organizationID}/svc/v1/transactions>

```
{
  "transactionEntries":[
    {
      "classificationNumber":"","
      "discount":null,
      "productCode":"ST0003",
      "quantity":1,
      "unit":null,
      "vatPercent":0,
      "product":52, // if we enter a text (string) instead of a number then an entry
with this text not related to any product will be created
      "unitNettoPrice":"1000.00"
    }
  ],
  "transactionNettoPrice":"1000.00",
  "transactionBruttoPrice":"1000.00",
  "transactionVatPrice":"0.00",
  "invoiceAnnotations":null,
  "paid":false,
  "currency":"USD",
  "paymentType":"CASH",
  "restOfPaidValue":1000,
  "paidValue":0,
  "invoiceDate":"2012-08-08",
```

```

"paymentDate":"2012-08-15",
"transactionDate":"2012-08-08",
"type":"INVOICE",
"mode":"SALE",
"customerAddress.postCode":"20001",
"customer":50,
"customerAddress.city":"Washington",
"customerAddress.country":"USA",
"customerAddress.street":"523 Island Ave",
"nipNumber":"1060000062",
"issuingPerson":"Jacob Smith",
"transactionNumber":"1/8/2012"
}

```

Response if created correctly

Code 201:

```

{"changelog":[{"deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"transaction","objectId":50035,"objectLabel":"1/8/2012 - Internet Center","oldValuesDictionary":false,"property":""},{deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"commisionuser","objectId":35,"objectLabel":"","oldValuesDictionary":false,"property":""},{deletedWith":"","modification":"CHANGE","newValueClass":"long","newValueDisplay":"10","newValuesDictionary":false,"objectClass":"customer","objectId":50,"objectLabel":"Internet Center","oldValueClass":"long","oldValueDisplay":"9","oldValuesDictionary":false,"property":"transactionsCount"}], "showCustomVatWarning":"true"}

```

If a sale transaction with this number already exists it won't be created and as a response we will get:

Code 400:

```

{"errorCode":"transactionNumberAlreadyExists","additionalInfo":{"suggestedNumber":"2/8/2012"}}

```

In a case of a **corrective invoice** items need to have two entries for each item from the original document: an item with a value before the correction and with a value after it.

Example of a corrective invoice for the transaction above:

```

{
"invoicePatternId": 1,
"exchangeRate": 1,

```

```
"exchangeRateDate": "2012-08-07",
"factor": 1,
"correctionCause": "Incorrectly entered data.",
"paid": false,
"bankAccount": 1,
"currency": "USd",
"paymentType": "CASH",
"restOfPaidValue": 1000,
"paidValue": 0,
"invoiceDate": "2022-04-08",
"invoiceReceptionDate": "2022-04-08",
"paymentDate": "2022-04-15",
"transactionDate": "2012-08-08",
"type": "CORRECTION",
"mode": "SALE",
"customerAddress.postCode": "20001",
"customer": 50,
"customerAddress.city": "Washington",
"customerAddress.country": "USA",
"customerAddress.street": "523 Island Ave",
"nipNumber": "106-00-00-062",
"recipientPhoneNumber": null,
"connectedDocDate": "2012-08-08", // date of the original document
"issuingPerson": "Jacob Smith",
"splitPayment": false,
"transactionNumber": "2022/04/004/KOR",
"connectedDocNumber": "1/8/2012", // number of the original document
"calculateFromGross": false,
"transactionEntries": [
  { // item before
    "unit": null,
    "classificationNumber": "",
    "baseEntryId": 50106, // ID of the corrected item from the original document
    "productCode": "ST0003",
    "quantity": 1,
    "entryPaid": "NOT_PAID",
    "entrySize": 1,
    "variant": 1,
    "vatPercent": "0",
```

```

    "recordId": 16, // number identifying the item within the document.
    "salePriceGroup": "A",
    "unitPurchaseNettoPrice": 0,
    "unitNettoPrice": "1000.00",
    "unitNettoPriceWithoutDiscount": "1000.00",
    "product": 52
  },
  { // item after
    "unit": null,
    "classificationNumber": "",
    "productCode": "ST0003",
    "quantity": 2,
    "entryPaid": "NOT_PAID",
    "entrySize": 1,
    "variant": 1,
    "vatPercent": 0,
    "recordId": 17, // number identifying the item within the document.
    "salePriceGroup": "A",
    "unitPurchaseNettoPrice": 0,
    "unitNettoPrice": "1000.00",
    "unitNettoPriceWithoutDiscount": "0.00",
    "invalidRecordId": 16, // recordId of the 'before' item
    "product": 52
  }
],
"transactionNettoPrice": "1000.00",
"transactionBruttoPrice": "1000.00",
"transactionVatPrice": "0.00",
"lastSelectedInvoiceReport": "CorrectiveInvoice2"
}

```

f) Quotes/Orders/Agreements

POST

<https://system.firmao.net/{organizationID}/svc/v1/offers> (Quotes / Orders)

<https://system.firmao.net/{organizationID}/svc/v1/agreements> (Agreements)

```

{
  "entries":[ // entries entry can be omitted if there are no quote items
    {
      "unit":"unit",
      "product":51, // if we enter a text (string) instead of a number then an entry with
this text not related to any product will be created. If we enter a number then the
system searches for an existing product and assigns it to the item
      "classificationNumber":"","
      "productCode":"ST0002",
      "quantity":1,
      "vatPercent":22,
      "unitNettoPrice":"818.85"
    }
  ],
  "nettoPrice":"818.85",
  "bruttoPrice":"999.00",
  "vatPrice":"180.15",
  "currency":"USD",
  "paymentType":"CASH",
  "paymentDate":"2012-11-09",
  "offerDate":"2012-11-09",
  "validFrom":"2012-11-09",
  "type":"OFFER", // OFFER - quote, ORDER - order, AGREEMENT - agreement,
  "mode":"SALE", // SALE, PURCHASE
  "offerStatus":"NEW", // NEW, SENT, DURING_NEGOTIATIONS, ACCEPTED,
REJECTED, EXECUTED
  "customerAddress.postCode":"20001",
  "daysToDueDate":7,
  "customer":50, // id kontrahenta w systemie - można użyć parametru
customerName, jeżeli nie chcemy łączyć z kontrahentem
  "customerAddress.city":"Washington",
  "customerAddress.country":"USA",
  "customerAddress.street":"523 Island Ave",
  "nipNumber":"1060000062",
  "issuingPerson":"Jacob Smith",
  "number":"1/11/2012/U" // field is required and has to be unique

```

```
}
```

Response if created correctly

Code 201:

```
{"changelog":[{"deletedWith":"","modification":"CREATE","newValuelsDictionary":false,"objectClass":"offer","objectId":2,"objectLabel":"1/11/2012/U - Internet Center","oldValuelsDictionary":false,"property":""},{deletedWith":"","modification":"CHANGE","newValueClass":"long","newValueDisplay":"2","newValuelsDictionary":false,"objectClass":"customer","objectId":50,"objectLabel":"Internet Center","oldValueClass":"long","oldValueDisplay":"1","oldValuelsDictionary":false,"property":"offersCount"}]}
```

g) Contacts

POST

<https://system.firmao.net/{organizationID}/svc/v1/contacts>

```
{
  "firstName":"Victoria",
  "lastName":"Johnson",
  "position":"Salesman",
  "description":null,
  "emails":[
    "v.johnson@mailinator.com"
  ],
  "phones":[
    "226548484",
    "226548485",
    "500111222"
  ],
  "tags":[]
},
"customer":51, // company ID in the system - Field is required
"label":"Victoria Johnson"
}
```

Response if created correctly

Code 201:

```
{"changelog":[{"deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"contact","objectId":50001,"objectLabel":"Victoria Johnson","oldValuesDictionary":false,"property":""},{deletedWith":"","modification":"CHANGE","newValueClass":"long","newValueDisplay":"0","newValuesDictionary":false,"objectClass":"customer","objectId":51,"objectLabel":"KOMP-SERVICE","oldValueClass":"long","oldValueDisplay":"0","oldValuesDictionary":false,"property":"mailsCount"},{"deletedWith":"","modification":"CHANGE","newValueClass":"long","newValueDisplay":"2","newValuesDictionary":false,"objectClass":"customer","objectId":51,"objectLabel":"COMP-SERVICE","oldValueClass":"long","oldValueDisplay":"0","oldValuesDictionary":false,"property":"contactsCount"},{"deletedWith":"","modification":"ATTACH","newValueClass":"contact","newValueDisplay":"Victoria Jonshon","newValueId":"50001","newValuesDictionary":false,"objectClass":"customer","objectId":51,"objectLabel":"COMP-SERVICE","oldValuesDictionary":false,"property":"contact_change"}]}
```

h) Notes

POST

<https://system.firmao.net/{organizationID}/svc/v1/salesnotes>

```
{
  "description":"Sample note",
  "type":"NOTE",
  "customer":51, // company ID in the system - Field is required
  "tags":[

  ],
  "date":"2013-04-10T17:12:20+02:00"
}
```

Response if created correctly

Code 201:

```
{"changelog":[{"deletedWith":"","modification":"CREATE","newValuesDictionary":false,"objectClass":"salesnote","objectId":50011,"objectLabel":"","oldValuesDictionary":false,"property":""},{deletedWith":"","modification":"CHANGE","newValueClass":"long","newValueDisplay":"1","newValuesDictionary":false,"objectClass":"customer","objectId":51,"objectLabel":"KOMP-SERVICE","oldValueClass":"long","oldValueDisplay":"0","oldValuesDictionary":false,"property":"salesNotesCount"},{"deletedWith":"","modification":"CHANGE","newValueClass":"date","newValueDisplay":"2017-09-21T13:49:11.691Z","newValuesDictionary":false,"objectClass":"customer","objectId":51,"objectLabel":"Name","oldValuesDictionary":false,"property":"lastContactDate"}]}
```

3. Modifying objects (PUT)

Modification of objects is done via PUT requests and should be directed to the address
`https://system.firmao.net/{organizationID}/svc/v1/{xxx}/{id}`

xxx - object type name

id - object ID

organizationID - ID of the organization visible in the URL after logging in to Firmao

Data type:

ContentType = application/json; charset=UTF-8

All fields available in objects are described in the help of data import. Custom fields have keys 'customFields.custom5' and take String values. Available fields can also be previewed by downloading data with a GET method (except the Permission field which is calculated and cannot be entered in a POST/PUT request).

Response codes:

API uses default HTTP error codes.

200 - modified correctly

4xx i 5xx - error occurred when modifying the object

Overall API is adapted to modify multiple fields in one request but it is tested mainly in modifying one field at the time. So when modifying multiple fields in one request unexpected errors may occur.

a) Product name change:

Modifying name of the product with ID 50001:

https://system.firmao.net/{organizationID}/svc/v1/products/50001

PUT

```
{
  "name": "Sample product name change"
}
```

Response:

```
{"changelog":[{"deletedWith":"","modification":"CHANGE","newValueClass":"string","newValueDisplay":"Sample product name change","newValuesDictionary":false,"objectClass":"product","objectId":50001,"objectLabel":"Sample product name change","oldValueClass":"string","oldValueDisplay":"Sample product","oldValuesDictionary":false,"property":"name"}]}
```

**In response the most important field is
property - name of the modified field
newValueDisplay - new field value
oldValueDisplay - old field value**

b) Company name change:

Modifying name of the company with ID 50001:

https://system.firmao.net/{organizationID}/svc/v1/customers/50001

PUT

```
{
  "name": "COMP-SERVICE"
}
```

Odpowiedź:

```
{"changelog":[{"deletedWith":"","modification":"CHANGE","newValueClass":"string","newValueDisplay":"COMP-SERVICE","newValuesDictionary":false,"objectClass":"customer","objectId":50001,"objectLabel":"q","oldValueClass":"string","oldValueDisplay":"q","oldValuesDictionary":false,"property":"name"}]}
```

c) Product's stock level change (requires enabling in the company settings: Direct modification of stock level)

<https://system.firmao.net/{organizationID}/svc/v1/products/50001>

PUT

```
{  
  "currentStoreState":50  
}
```

Response (Response is so long because the server is calculating various data. In this case it calculates the value of products in the warehouse)

```
{"changelog":[{"deletedWith":"","modification":"CHANGE","newValueClass":"float","newValueDisplay":"50","newValuesDictionary":false,"objectClass":"product","objectId":50001,"objectLabel":"Sample product name change","oldValueClass":"float","oldValueDisplay":"0.000","oldValuesDictionary":false,"property":"currentStoreState","subClass":"warehouse","subId":1,"subLabel":"Warehouse 1"},{"deletedWith":"","modification":"CHANGE","newValueClass":"float","newValueDisplay":"0.00","newValuesDictionary":false,"objectClass":"product","objectId":50001,"objectLabel":"Sample product name change","oldValueClass":"float","oldValuesDictionary":false,"property":"netPriceInStore"}]}
```

4. Getting objects (GET)

Getting is done with GET requests and should be directed to address:

<https://system.firmao.net/{organizationID}/svc/v1/{xxx}>

xxx - object type name

organizationID - ID of the organization visible in the URL after logging in

Paging

In GET requests we are dealing with results paging. At our disposal we have parameters 'start' (from which result the paging starts), 'limit' (max amount of results on a page). Max value of the limit is dependable on the bought license.

In the server's response there is a field called 'totalSize' that states the total amount of objects so we know if there are any other objects on the next page.

WARNING: In some cases in the response we get totalSize with value -1. This means that there are at least as many results as the limit, which means that there is most likely another page of results.

Sorting

In the request, we can also specify the field by which we sort (sort parameter) and the direction of sorting (dir parameter). If there are no sorting parameters, the results will be sorted by default

Filtering

We can specify filtering in the request. The filtering format is:

fieldName(comparisonType), where field name is the field which we filter and comparisonType is:

contains - does field contain value,

notContains - does field not contain value,

eq - is field equal to value ,

ne - is field not equal to value ,

ge - is field bigger than (only for number and date fields),

nge - is field bigger (so smaller or equal) than (only for number and date fields),

le - is field smaller than (only for number and date fields)

nle - is field not smaller (so bigger or equal) than (only for number and date fields)

Filtering examples:

https://system.firmao.net/{organizationID}/svc/v1/projects?id(eq)=15 gets project with ID 15

https://system.firmao.net/{organizationID}/svc/v1/projects?id(ge)=15 gets projects with ID bigger than 15

https://system.firmao.net/{organizationID}/svc/v1/projects?name(contains)=proj gets projects containing the letter string 'proj'.

Response codes

API uses default HTTP error codes

200 - correct getting of data

4xx i 5xx - error occurred when getting data

a) Getting projects - projects

GET

https://system.firmao.net/{organizationID}/svc/v1/projects?start=0&limit=50&sort=name&dir=ASC&name(contains)=sam&dataFormat=MEDIUM

Response:

```
{
  "data": [
    {
      "actualCost": 0,
      "actualCostSummaryBrutto": 0,
      "actualCostSummaryNetto": 0,
      "actualIncomesBrutto": 0,
      "actualIncomesNetto": 0,
      "actualIncomeSummaryBrutto": 0,
      "actualIncomeSummaryNetto": 0,
      "actualOutgoingsBrutto": 0,
      "actualOutgoingsNetto": 0,
      "actualProfitSummaryBrutto": 0,
      "actualProfitSummaryNetto": 0,
      "actualTaskProfit": 0,
      "actualTransactionProfitBrutto": 0,
```

```
"actualTransactionProfitNetto": 0,
"actualWorkHours": null,
"approved": false,
"attachmentsCount": 0,
"autoCalcEstimatedHours": true,
"autoCalcPlannedCost": true,
"autoCalcPlannedIncome": true,
"balance": 0,
"budget": 0,
"contact": null,
"costRealizationPercent": 0,
"createdBy": {
  "id": 1,
  "label": "Jacob Smith"
},
"creationDate": "2017-09-21T13:08:59Z",
"customer": null,
"customFields": null,
"description": "Opis",
"endDate": null,
"estimatedHours": null,
"finalProfitSummaryBrutto": 0,
"finalProfitSummaryNetto": 0,
"finalTransactionProfitBrutto": 0,
"finalTransactionProfitNetto": 0,
"hourlyPrice": null,
"hourlyRate": null,
"id": 50001,
"income": 0,
"incomeRealizationPercent": 0,
"lastUpdateDate": "2017-09-21T13:08:59Z",
"managers": [
  {
    "id": 10,
    "label": "Denise Taylor"
  }
],
"margin": null,
"name": "Sample",
```

```
"overdraftSummaryBrutto": 0,
"overdraftSummaryNetto": 0,
"permissions": {
  "project": [
    "basic_update",
    "costs",
    "create",
    "extended_update",
    "read",
    "update"
  ],
  "task": [
    "basic_update",
    "costs",
    "create",
    "read",
    "update"
  ]
},
"plannedCost": 0,
"plannedCostSummaryBrutto": 0,
"plannedCostSummaryNetto": 0,
"plannedIncome": 0,
"plannedIncomesBrutto": 0,
"plannedIncomesNetto": 0,
"plannedIncomeSummaryBrutto": 0,
"plannedIncomeSummaryNetto": 0,
"plannedOutgoingsBrutto": 0,
"plannedOutgoingsNetto": 0,
"plannedProfitSummaryBrutto": 0,
"plannedProfitSummaryNetto": 0,
"plannedTaskProfit": 0,
"plannedTransactionProfitBrutto": 0,
"plannedTransactionProfitNetto": 0,
"purchaseTransactionBrutto": 0,
"purchaseTransactionNetto": 0,
"restrictedTeamMembers": [],
"saleTransactionBrutto": 0,
"saleTransactionNetto": 0,
```

```
"startDate": "2012-07-16T22:00:00Z",
"tags": [
  {
    "color": "#FFFFFF",
    "id": 3,
    "label": "Project approved"
  }
],
"teamMembers": [
  {
    "id": 6,
    "label": "Emma Clark"
  }
],
"transactionsBalanceBrutto": 0,
"transactionsBalanceNetto": 0
}
],
"dir": "ASC",
"sort": "name",
"totalSize": 1
}
```

b) Getting tasks - tasks

GET

[https://system.firmao.net/{organizationID}/svc/v1/tasks?start=0&limit=50&sort=name&dir=ASC&name\(contains\)=task&dataFormat=MEDIUM](https://system.firmao.net/{organizationID}/svc/v1/tasks?start=0&limit=50&sort=name&dir=ASC&name(contains)=task&dataFormat=MEDIUM)

Response:

```
{
  "data": [
    {
      "actualCost": 0,
      "actualEndDate": null,
      "actualStartDate": null,
      "actualWorkHours": 0,

```

```
"additionalCosts": 0,
"address": null,
"attachmentsCount": 0,
"autoCalcActualCost": true,
"autoCalcIncome": true,
"autoCalcPlannedCost": true,
"autoCalcPlannedIncome": true,
"autoStatus": false,
"balance": 0,
"contact": null,
"createdBy": {
  "id": 1,
  "label": "Jacob Smith"
},
"creationDate": "2017-09-21T13:11:51Z",
"customer": null,
"customFields": null,
"description": "Opis",
"discount": null,
"estimatedHours": 8,
"hasEndDateReminder": false,
"hasStartDateReminder": false,
"hasSubtasks": false,
"id": 50033,
"income": 0,
"lastUpdateDate": "2017-09-21T13:11:51Z",
"letter": null,
"name": "Sample task",
"orderBruttoPrice": null,
"orderCalculateFromGross": false,
"orderNettoPrice": null,
"orderQuantity": null,
"orderUnit": null,
"orderUnitPrice": null,
"orderVatPercent": null,
"orderVatPrice": null,
"parent": null,
"percentDone": 0,
"permissions": {
```



```

    "task": [
      "basic_update",
      "costs",
      "create",
      "read",
      "update"
    ]
  },
  "plannedCost": 200,
  "plannedEndDate": "2012-07-17T22:00:00Z",
  "plannedEndDateType": "EXACTLY",
  "plannedIncome": 400,
  "plannedStartDate": "2012-07-16T22:00:00Z",
  "plannedStartDateType": "EXACTLY",
  "priority": 300,
  "product": null,
  "project": null,
  "reminderEndDateAdvance": null,
  "reminderStartDateAdvance": null,
  "resource": null,
  "responsibleUsers": [
    {
      "id": 2,
      "label": "Ethan Lam"
    }
  ],
  "status": {
    "group": "OPEN",
    "icon": null,
    "key": "WAITING",
    "label": "Waiting",
    "label_de": "Wird erwartet",
    "label_en": "Waiting"
  },
  "subcontact": null,
  "subcontractor": null,
  "subtasksCount": 0,
  "tags": [],
  "taskType": "PROJECT",

```

```
        "transactionBruttoPrice": 0,  
        "transactionNettoPrice": 0  
    }  
],  
"dir": "ASC",  
"sort": "name",  
"totalSize": 1  
}
```

c) Getting products - products

GET

<https://system.firmao.net/{organizationID}/svc/v1/products?start=0&limit=50&sort=name&dir=ASC&dataFormat=MEDIUM>

Response:

```
{  
  "data": [  
    {  
      "classificationNumber": null,  
      "createdBy": {  
        "id": 1,  
        "label": "Jacob Smith"  
      },  
      "creationDate": "2012-07-17T08:39:15Z",  
      "currentStoreState": 50.00,  
      "customFields": null,  
      "description": null,  
      "endSellingDate": null,  
      "grossPrice": 12.30,  
      "grossPriceInStore": 615.00,  
      "id": 50001,  
      "lastUpdateDate": "2012-07-17T08:39:15Z",  
      "margin": 20.00,  
      "name": "Sample product name change",  
      "netPrice": 10.00,  
      "netPriceInStore": 500.00,  
      "overrunState": true,  
    },  
  ],  
}
```

```

"permissions":{
  "product":[
    "buyprice",
    "create",
    "extended_update",
    "read",
    "update"
  ],
  "semiproduct":[
    "create",
    "read",
    "update"
  ],
  "transactionentry":[
    "create",
    "read",
    "update"
  ]
},
"productCode":"PR1",
"saleGrossPrice":14.76,
"saleGrossPriceInStore":738.00,
"saleNetPrice":12.00,
"saleNetPriceInStore":600.00,
"saleable":true,
"startSellingDate":null,
"tags":[

],
"unit":"szt.",
"vat":23.00,
"warningStoreState":88.00
}
],
"dir":"ASC",
"sort":"name",
"totalSize":1
}

```

d) Getting companies - customers

GET

https://system.firmao.net/{organizationID}/svc/v1/customers?start=0&limit=50&sort=name&dir=ASC&dataFormat=MEDIUM

Response:

```
{
  "data":[
    {
      "actualCost":0.00,
      "balance":0.00,
      "bankAccountNumber":"50 1020 5558 1111 1702 0600 0067",
      "correspondenceAddress":null, "createdBy":null,

      "creationDate":"2002-12-13T11:40:45Z",
      "customFields":null,
      "customerGroups":[

    ],
      "customerType":"PARTNER",
      "description":"Company specializing in computer service",
      "email":"compervice.firmao@mailinator.com",
      "email2":"computers@mailinator.com", "email3":null,

      "emails":[
        "compervice.firmao@mailinator.com",
        "computers@mailinator.com"
      ],
      "employeesNumber":1,
      "faxNumber":"22 398 78 87",
      "hourlyRate":null,
      "id":51,
      "identificationNumber":null,
      "income":0.00,
      "industry":{"
        "key":"3",
        "label":"IT"
      },
      "krs":null,
      "label":"COMP-SERVICE",
```

```
"lastSalesNote":null,
"lastUpdateDate":"2010-05-29T20:53:30Z",
"name":"COMP-SERVICE John Smith",
"nipNumber":"7291077843",
"officeAddress":{"
  "city":"New York",
  "country":"USA",
  "county":null,
  "postCode":"10001",
  "street":"143 Parker Ave"
},
"ownership":"PRIVATE",
"permissions":{"
  "vatrecord":[
    "create",
    "read",
    "update"
  ],
  "task":[
    "basic_update",
    "costs",
    "create",
    "read",
    "update"
  ],
  "accountingoperation":[
    "create",
    "read",
    "update"
  ],
  "customer":[
    "costs",
    "create",
    "read",
    "update"
  ]
},
"phone":"22 345 74 26",
"phoneOther":null,
"phoneOther2":null,
"phones":[
  "22 345 74 26"
],
```

```

    "plannedCost":0.00,
    "plannedIncome":0.00,
    "stats":null,
    "status":null,
    "tags":[

    ],
    "vatUE":"PL7291077843",
    "voivodeship":null,
    "website":"www.compservice.bis"
  }
],
"dir":"DESC",
"sort":"id",
"totalSize":1
}

```

e) Getting invoices - transactions

GET

<https://system.firmao.net/{organizationID}/svc/v1/transactions?start=0&limit=50&sort=name&dir=ASC&dataFormat=MEDIUM>

Response:

```

{
  "data":[
    {
      "actualPaymentDate":null,
      "actualPaymentType":null,
      "advanceInvoiceGroupId":null,
      "approved":false,
      "calculateFromGross":false,
      "createdBy":null,
      "creationDate":"2002-12-13T11:40:45Z",
      "currency":"PLN",
      "customFields":null,
      "customer":{
        "emails":[
          "icenter.firmao@mailinator.com",
          "iceneter.biuro1@mailinator.com",
          "iceneter.biuro2@mailinator.com"
        ]
      }
    }
  ]
}

```

```

    ],
    "id":50,
    "label":"Internet Center"
  },
  "customerAddress":null,
  "customerName":null,
  "description":null,
  "hasReminder":false,
  "id":50,
  "invalidInvoiceDate":null,
  "invalidTransactionNumber":null,
  "invoiceAnnotations":null,
  "invoiceDate":"2010-09-27",
  "issuingPerson":null,
  "lastUpdateDate":"2010-05-29T20:53:30Z",
  "mail":null,
  "mode":"SALE",
  "name":"Sprzedaż DE PC",
  "nipNumber":null,
  "paid":true,
  "paidValue":999,
  "parcelCourierName":null,
  "parcelDeliveryType":null,
  "parcelDescription":null,
  "parcelNumber":null,
  "parcelShipmentDate":null,
  "paymentDate":"2010-09-30",
  "paymentStatus":null,
  "paymentType":"TRANSFER",
  "permissions":{
    "transaction":[
      "create",
      "extended_update",
      "read",
      "update"
    ],
    "transactionentry":[
      "create",
      "read",
      "update"
    ]
  }
]
},
"project":{

```

```

    "id":4,
    "name":"GANT USE CASE2"
  },
  "reminderAdvance":null,
  "restOfPaidValue":0,
  "sellerAddress":null,
  "storageStateUpdate":"NO_UPDATE",
  "tags":[

  ],
  "task":{
    "id":24,
    "name":"GANTT UC2 C",
    "status":{
      "group":"OPEN"
    }
  },
  "transactionBruttoPrice":999,
  "transactionDate":"2010-09-24",
  "transactionGroupBruttoPrice":null,
  "transactionGroupVatPrice":null,
  "transactionNettoPrice":818.85,
  "transactionNumber":"1/1/2010",
  "transactionVatPrice":180.15,
  "type":"INVOICE"
}
],
"dir":"ASC",
"sort":"name",
"totalSize":1
}

```

f) Getting invoices of a company - transactions

GET

[https://system.firmao.net/{organizationID}/svc/v1/transactions?start=0&limit=50&](https://system.firmao.net/{organizationID}/svc/v1/transactions?start=0&limit=50&customer(eq)=10)

customer(eq)=10

Where 10 is the ID of the company

Response:


```
{
  "data": [
    {
      "actualPaymentDate": null,
      "actualPaymentType": null,
      "advanceInvoiceGroupId": null,
      "approved": true,
      "attachmentsCount": 0,
      "bankAccount": {
        "id": 1,
        "label": "test"
      },
      "baseBruttoPrice": 0,
      "baseNettoPrice": 0,
      "basePaidValue": 0,
      "baseRestOfPaidValue": 0,
      "baseVatPrice": 0,
      "calculateFromGross": false,
      "commision": 0,
      "commisionRate": 0,
      "connectedDocDate": null,
      "connectedDocNumber": null,
      "connectedDocType": null,
      "contact": null,
      "correctionCause": null,
      "createdBy": {
        "id": 1,
        "label": "Jacob Smith"
      },
      "creationDate": "2017-09-21T13:38:08Z",
      "currency": "USD",
      "customer": {
        "discount": null,
        "emails": null,
        "id": 1,
        "label": "DELL",
        "salePriceGroup": "A"
      },
      "customerAddress": null,
    }
  ]
}
```

```
"customerBankAccountNumber": null,
"customerName": "DELL",
"customFields": null,
"dateForExchangeRate": "2017-09-21",
"dateLabel": "SELL_DATE",
"description": null,
"documentDate": "2017-09-21",
"documentGenerated": false,
"documentNumber": "2017/9/21/21",
"doNotIncludeInLedgerReport": false,
"exchangeRate": null,
"exchangeRateDate": null,
"factor": 1,
"finalPartialInvoice": false,
"hasReminder": false,
"id": 50034,
"integrationType": null,
"invoiceAnnotations": null,
"invoiceDate": "2017-09-21",
"invoicePatternId": 1,
"invoiceReceptionDate": null,
"issuingPerson": "Jacob Smith",
"issuingPersonObject": {
  "firstName": "Jacob",
  "id": 1,
  "lastName": "Smith"
},
"lastSelectedInvoiceReport": "Invoice",
"lastUpdateDate": "2017-09-21T13:38:08Z",
"mail": null,
"mode": "SALE",
"name": null,
"nipNumber": null,
"paid": false,
"paidValue": 0,
"paidValueInvoice": 0,
"parcelCourierName": null,
"parcelDeliveryType": null,
"parcelDescription": null,
```

```
"parcelNumber": null,
"parcelShipmentDate": null,
"paymentDate": "2017-09-21",
"paymentStatus": "NOT_PAID",
"paymentType": "CASH",
"permissions": {
  "transaction": [
    "create",
    "extended_update",
    "read",
    "update"
  ]
},
"phone": "+48 42 356 53 34",
"profit": 0,
"profitCommision": 0,
"profitCommisionRate": 0,
"projects": [],
"purchaseVatCategory": null,
"rateFromCurrency": null,
"recipient": null,
"recipientAddress": null,
"recipientName": null,
"recipientNipNumber": null,
"recipientPhoneNumber": null,
"reminderAdvance": null,
"restOfPaidValue": 0,
"restOfPaidValueInvoice": 0,
"sellerAddress": {},
"sellerBankAccountNumber": "PL83 1010 1023 0000 2613 9510 0000",
"smallTaxPayer": false,
"storageStateUpdate": "NO_UPDATE",
"tags": [],
"task": null,
"transactionBruttoPrice": 0,
"transactionDate": "2017-09-21",
"transactionGroupBruttoPrice": 0,
"transactionGroupVatPrice": 0,
"transactionNettoPrice": 0,
```

```

    "transactionNettoPriceTax": 0,
    "transactionNumber": "2017/9/21/21",
    "transactionOrInvoiceDate": "2017-09-21",
    "transactionVatPeriod": "09.2017",
    "transactionVatPrice": 0,
    "type": "INVOICE",
    "warehouse": {
      "id": 1,
      "name": "Warehouse 1"
    }
  }
],
"dir": "DESC",
"sort": "transactionDate",
"totalSize": 1
}

```

g) Getting invoice items - transactions

GET

[https://system.firmao.net/{organizationID}/svc/v1/transactionentries?transaction\(eq\)=50](https://system.firmao.net/{organizationID}/svc/v1/transactionentries?transaction(eq)=50)

Where 50 is the ID of the invoice

Response:

```

{
  "data": [
    {
      "balance": null,
      "bruttoPrice": 999.00,
      "calculateFromGross": false,
      "classificationNumber": null,
      "description": null,
      "discount": null,
      "entryDate": "2010-09-24",
      "id": 50,
      "invalidEntry": null,
      "mode": "SALE",
    }
  ]
}

```

```

"name":null,
"nettoPrice":818.85,
"offer":null,
"permissions":{
  "transactionentry":[
    "create",
    "read",
    "update"
  ]
},
"product":{
  "id":51,
  "name":"De PC Pentium 2,74 2GB"
},
"productCode":"ST0002",
"project":null,
"quantity":1.000,
"transaction":{
  "customerName":null,
  "id":50,
  "name":"Sprzedaż DE PC",
  "transactionNumber":"1/1/2010"
},
"type":"ACTUAL",
"unit":"szt.",
"unitBruttoPrice":999.00,
"unitNettoPrice":818.85,
"vatPercent":22.00,
"vatPrice":180.15
}
],
"dir":"ASC",
"sort":"transaction",
"totalSize":1
}

```

h) Getting quotes/orders - offers

GET

<https://system.firmao.net/{organizationID}/svc/v1/offers?start=0&limit=50&sort=name&dir=ASC&dataFormat=MEDIUM>

```
{
  "data":[
    {
      "annotations":null,
      "bruttoPrice":999,
      "calculateFromGross":false,
      "contact":null,
      "createdBy":{
        "id":1,
        "label":"Jacob Smith"
      },
      "creationDate":"2012-11-09T12:06:47Z",
      "currency":"PLN",
      "customFields":null,
      "customer":{
        "email":"icenter.firmao@mailinator.com",
        "id":50,
        "label":"Internet Center"
      },
      "customerAddress":{
        "city":"Łódź",
        "country":"Polska",
        "county":null,
        "postCode":"91-477",
        "street":"Piotrkowska 147/8"
      },
      "customerName":"Internet Center dsfds",
      "daysToDueDate":7,
      "description":null,
      "hasReminder":false,
      "id":1,
      "issuingPerson":"Jacob Smith",
      "lastUpdateDate":"2012-11-09T12:06:48Z",
      "mail":null,
      "mode":"SALE",
      "name":null,
      "nettoPrice":818.85,
      "nipNumber":"1060000062",
      "number":"1/11/2012/U",
      "offerDate":"2012-11-09",
      "offerStatus":"NEW",
      "parcelCourierName":null,
    }
  ]
}
```

```
"parcelDeliveryType":null,
"parcelDescription":null,
"parcelNumber":null,
"parcelShipmentDate":null,
"paymentDate":"2012-11-09",
"paymentType":"CASH",
"permissions":{
  "offer":[
    "create",
    "read",
    "update"
  ],
  "transactionentry":[
    "create",
    "read",
    "update"
  ]
},
"project":null,
"reminderAdvance":null,
"sellerAddress":{

},
"tags":[

],
"task":null,
"type":"AGREEMENT",
"validFrom":"2012-11-09",
"vatPrice":180.15
}
],
"dir":"DESC",
"sort":"offerDate",
"totalSize":1
}
```

i) Getting contacts - contacts

GET

<https://system.firmao.net/{organizationID}/svc/v1/contacts?start=0&limit=50&sort=firstName&dir=ASC&dataFormat=MEDIUM>

```
{
  "data": [
    {
      "address": null,
      "createdBy": null,
      "creationDate": "2002-12-13T11:40:45Z",
      "customer": {
        "emails": [
          "iceneter.biuro1@mailinator.com",
          "iceneter.biuro2@mailinator.com",
          "icenter.firmao@mailinator.com"
        ],
        "id": 50,
        "label": "Internet Center"
      },
      "customFields": null,
      "departament": "Secretariat",
      "description": null,
      "email": "anna.wojewodzka@mailinator.com",
      "email2": null,
      "email3": null,
      "emails": [
        "anna.wojewodzka@mailinator.com"
      ],
      "firstName": "Anna",
      "hasReminder": false,
      "id": 51,
      "instantMessenger": null,
      "label": "Anna Wojewódzka",
      "lastName": "Wojewódzka",
      "lastUpdateDate": "2010-05-29T20:53:30Z",
      "nextContactDate": null,
    }
  ]
}
```



```

"permissions": {
  "contact": [
    "create",
    "read",
    "update"
  ]
},
"phone": "42 688 14 22",
"phoneOther": null,
"phoneOther2": null,
"phones": [
  "42 688 14 22"
],
"position": "Secretary",
"reminderAdvance": null,
"skypeld": null,
"tags": []
},
{
  "address": null,
  "createdBy": null,
  "creationDate": "2002-12-13T11:40:45Z",
  "customer": {
    "emails": [
      "iceneter.biuro1@mailinator.com",
      "iceneter.biuro2@mailinator.com",
      "icenter.firmao@mailinator.com"
    ],
    "id": 50,
    "label": "Internet Center"
  },
  "customFields": null,
  "departament": null,
  "description": null,
  "email": "m.nowak@mailinator.com",
  "email2": null,
  "email3": null,
  "emails": [
    "m.nowak@mailinator.com"
  ]
}

```

```
    ],
    "firstName": "Gary",
    "hasReminder": false,
    "id": 50,
    "instantMessenger": null,
    "label": "Gary Nowak",
    "lastName": "Nowak",
    "lastUpdateDate": "2017-09-21T14:44:57Z",
    "nextContactDate": null,
    "permissions": {
      "contact": [
        "create",
        "read",
        "update"
      ]
    },
    "phone": "48507306295",
    "phoneOther": "641044884",
    "phoneOther2": null,
    "phones": [
      "48507306295",
      "641044884"
    ],
    "position": "Salesman",
    "reminderAdvance": null,
    "skypeId": null,
    "tags": []
  }
],
"dir": "ASC",
"sort": "firstName",
"totalSize": 2
}
```

j) Getting company groups - customergroups

<https://system.firmao.net/{organizationID}/svc/v1/customergroups?start=0&limit=50&sort=firstName&dir=ASC&dataFormat=MEDIUM>

```
{
  "data": [
    {
      "createdBy": {
        "id": 2,
        "label": "Ethan Lam"
      },
      "creationDate": "2010-07-13T07:00:00Z",
      "hasTransactions": [],
      "id": 1,
      "lastUpdateDate": "2010-07-13T07:00:00Z",
      "name": "Regular customers",
      "permissions": {
        "customergroup": [
          "create",
          "read",
          "update"
        ]
      },
      "salesmen": [
        {
          "id": 5,
          "label": "Anthony Wilson"
        }
      ]
    }
  ],
  "dir": "ASC",
  "multiloggingdetected": true,
  "sort": "name",
  "totalSize": 1
}
```

k) Getting notes - salesnotes

<https://system.firmao.net/{organizationID}/svc/v1/salesnotes?start=0&limit=50&sort=firstName&dir=ASC&dataFormat=MEDIUM> {

```
"data": [  
  {  
    "agreement": null,  
    "attachmentsCount": 0,  
    "contact": null,  
    "createdBy": {  
      "id": 1,  
      "label": "Jacob Smith"  
    },  
    "creationDate": "2017-09-12T12:06:19Z",  
    "customer": null,  
    "customFields": null,  
    "date": "2017-09-12T12:06:00Z",  
    "description": "meeting",  
    "firstForCustomer": false,  
    "id": 50006,  
    "lastUpdateDate": "2017-09-12T12:06:18Z",  
    "mail": null,  
    "offer": null,  
    "permissions": {  
      "salesnote": [  
        "create",  
        "read",  
        "update"  
      ]  
    },  
    "product": null,  
    "project": null,  
    "tags": [],  
    "task": null,  
    "transaction": null,  
    "type": "MEETING"  
  },  
],
```

```
"dir": "ASC",
"multiloggingdetected": true,
"sort": "date",
"totalSize": 1
}
```

I) Getting stock documents - storedoccs

<https://system.firmao.net/{organizationID}/svc/v1/storedoccs?start=0&limit=50&sort=firstName&dir=ASC&dataFormat=MEDIUM> {

```
"data": [
  {
    "attachmentsCount": 0,
    "bankAccount": null,
    "baseBruttoPrice": 29,
    "baseNettoPrice": 23.58,
    "basePaidValue": 0,
    "baseRestOfPaidValue": 29,
    "baseVatPrice": 5.42,
    "calculateFromGross": false,
    "contact": null,
    "createdBy": {
      "id": 1,
      "label": "Jacob Smith"
    },
    "creationDate": "2017-09-13T08:26:34Z",
    "currency": "USD",
    "customer": {
      "emails": [
        "compservice.firmao@mailinator.com"
      ],
      "id": 51,
      "label": "COMP-SERVICE"
    },
    "customerAddress": {
      "city": "New York",
      "country": "USA",
      "county": null,

```

```
"postCode": "10001",
"street": "143 Parker Ave"
},
"customerBankAccountNumber": "50 1020 5558 1111 1702 0600 0067",
"customerName": "COMP-SERVICE John Smith",
"customFields": null,
"dateForExchangeRate": "2017-09-13",
"description": null,
"documentDate": "2017-09-13",
"documentNumber": "2017/09/001/PZ",
"exchangeRate": null,
"exchangeRateDate": null,
"factor": 1,
"hasReminder": false,
"id": 1,
"invoiceAnnotations": null,
"invoiceDate": "2017-09-13",
"issuingPerson": "Jacob Smith",
"issuingPersonObject": {
  "firstName": "Jacob",
  "id": 1,
  "lastName": "Smith"
},
"lastSelectedInvoiceReport": "Goods received note",
"lastUpdateDate": "2017-09-13T08:26:33Z",
"name": null,
"nipNumber": "7291077843",
"paid": false,
"paidValue": 0,
"parcelCourierName": null,
"parcelDeliveryType": null,
"parcelDescription": null,
"parcelNumber": null,
"parcelShipmentDate": null,
"paymentDate": "2017-09-13",
"paymentType": "CASH",
"permissions": {
  "storedoc": [
    "create",
```

```

        "read",
        "update"
    ]
},
"phone": "22 345 74 26",
"rateFromCurrency": null,
"receivingPerson": null,
"receivingPersonObject": null,
"recipient": null,
"recipientAddress": null,
"recipientName": null,
"recipientNipNumber": null,
"recipientPhoneNumber": null,
"reminderAdvance": null,
"restOfPaidValue": 29,
"sellerAddress": {},
"sellerBankAccountNumber": null,
"sourceWarehouse": null,
"storageDocBruttoPrice": 29,
"storagedocDate": "2017-09-13",
"storageDocNettoPrice": 23.58,
"storageDocNumber": "2017/09/001/GR",
"storageDocVatPrice": 5.42,
"storageStateUpdate": "AUTO_UPDATE",
"tags": [],
"task": null,
"type": "OUTSIDE_INCOME",
"warehouse": {
    "id": 1,
    "name": "Warehouse 1"
}
}
],
"dir": "ASC",
"multiloggingdetected": true,
"sort": "name",
"totalSize": 1
}

```

m) Getting files list - documents

<https://system.firmao.net/{organizationID}/svc/v1/documents?start=0&limit=50&dataFormat=MEDIUM>

```
{
  "data": [
    {
      "contentType": "image/png",
      "createdBy": null,
      "creationDate": "2017-09-08T07:27:05Z",
      "description": null,
      "fileSize": 15861,
      "isFile": true,
      "name": "specialist.png",
      "objectClass": "mail",
      "objectDeleted": false,
      "objectId": 23,
      "objectLabel": "Training Calendar - September 2017",
      "tags": [],
      "uploaded": true,
      "url": null
    },
    {
      "contentType": "text/plain",
      "createdBy": null,
      "creationDate": "2017-09-08T07:27:08Z",
      "description": null,
      "fileSize": 117,
      "isFile": true,
      "name": "Delivery_raport.txt",
      "objectClass": "mail",
      "objectDeleted": false,
      "objectId": 27,
      "objectLabel": "Mail delivery failed: returning message to sender",
      "tags": [],
      "uploaded": true,
      "url": null
    }
  ]
}
```



```
    }  
  ],  
  "dir": "ASC",  
  "sort": "name",  
  "totalSize": 32  
}
```

n) Getting mails - mails

[https://system.firmao.net/{organizationID}/svc/v1/mails?start=0&limit=50&sort=sendDate&dir=DESC&box\(eq\)=INBOX&dataFormat=MEDIUM](https://system.firmao.net/{organizationID}/svc/v1/mails?start=0&limit=50&sort=sendDate&dir=DESC&box(eq)=INBOX&dataFormat=MEDIUM)

```
{  
  "changelog": [],  
  "data": [  
    {  
      "answerFor": null,  
      "attachment": false,  
      "bcc": [],  
      "box": "INBOX",  
      "cc": [  
        "mailaddress@mailinator.com"  
      ],  
      "contact": null,  
      "createdBy": null,  
      "customer": {  
        "email": "dell@gmail.com",  
        "id": 1,  
        "label": "DELL"  
      },  
      "customFields": null,  
      "done": false,  
      "folder": null,  
      "id": 58,  
      "mailAccount": {  
        "id": 1,  
        "incomingServerProtocol": "IMAP",  
        "label": "test@firmao.net"  
      },  
    },  
  ],  
}
```

```

"mailInvitations": [],
"mailSendingTime": null,
"project": null,
"read": true,
"replyTo": "\"dell@gmail.com\" <dell@gmail.com>",
"sender": "dell@gmail.com",
"senderLabel": "dell@gmail.com",
"sentDate": "2017-09-08T07:56:50Z",
"status": "RECEIVE_SUCCEED",
"subject": "as",
"tags": [],
"task": null,
"to": [
  "test@firmao.net"
]
},
{
"answerFor": null,
"attachment": false,
"bcc": [],
"box": "INBOX",
"cc": [],
"contact": null,
"createdBy": null,
"customer": {
  "email": "dell@wp.net",
  "id": 50002,
  "label": "DELL"
},
"customFields": null,
"done": false,
"folder": null,
"id": 24,
"mailAccount": {
  "id": 2,
  "incomingServerProtocol": "IMAP",
  "label": "dell@gmail.com"
},
"mailInvitations": [],

```

```

    "mailSendingTime": null,
    "project": null,
    "read": true,
    "replyTo": "\"DELL\" <dell@wp.net>",
    "sender": "dell@wp.pl",
    "senderLabel": "DELL",
    "sentDate": "2017-09-08T07:26:27Z",
    "status": "REPLIED",
    "subject": "Test Firmao",
    "tags": [],
    "task": null,
    "to": [
      "mailadress@firmao.net",
      "dell@gmail.com"
    ]
  },
],
"dir": "DESC",
"sort": "sentDate",
"totalSize": 51
}

```

o) Getting invoice numbering series

GET:

<https://system.firmao.net/{identyfikacjaOrganizacji}/svc/v1/invoicenumbers?dataFormat=MEDIUM>

Request gets a list of numbering series - their numbers, IDs, names and numbering templates for each document type (regular invoice / proforma/ quote/ order etc.)

ID that has been gotten this way can be used e.g. to get the next invoice number in order:

GET:

<https://system.firmao.net/{OrganizationID}/svc/v1/invoicenumbers/next?objectClass={objectName}&type={documentType}&mode={saleMode}&id={numberingSeriesID}>

ID: we receive from previous request

mode:

- SALE - sale document
- PURCHASE - purchase document

objectClass together with type dictionary:

- transaction (Transaction)
 - invoice (Invoice)
 - receipt (Receipt)
 - proforma (Proforma invoice)
 - correction (Corrective invoice)
 - bill (Bill)
 - invoicemargin (Margin invoice)
 - advance_invoice (Advance invoice)
- agreement (Agreement)
 - agreement (Agreement)
- offer (Quote/order)
 - offer (Quote)
 - order (Order)
- storedoc (Stock document)
 - outside_income (Goods Received Note)
 - outside_release (Goods Issued Note),
 - inside_release (Internal Goods Issued Note),
 - inside_income (Internal Goods Received Note),
 - warehouse_transfer (Internal Goods Transfer Note)

For example in order to get the next number of a sale proforma invoice for organization Stellis and numbering series with ID 2 you have to send following request:

GET:

<https://system.firmao.net/stellis/svc/v1/invoicenumbers/next?objectClass=transaction&type=proforma&mode=SALE&id=2>

Getting next sale order number for the same numbering series and organization is:

GET:

<https://system.firmao.net/stellis/svc/v1/invoicenumbers/next?objectClass=offer&type=order&mode=SALE&id=2>

Getting next agreement number:

GET:

<https://system.firmao.net/stellis/svc/v1/invoicenumbers/next?objectClass=agreement&type=agreement&mode=SALE&id=2>

5. Removing objects (PUT, DELETE)

In Firmao most objects aren't removed but just marked as removed and can be restored. Such objects are removed by modifying field 'deleted' to 'true' as it is described in chapter 3.

Other objects (that don't have field 'deleted') are removed with DELETE method. Among others these are:

- Reminders
- Attachments
- Invoice entries

These objects cannot be restored.

6. Detailed description of how to create, edit, finish and start work times

Creating new work time **just with a start time:**

<https://system.firmao.net/{organizationID}/svc/v1/timeentries>

POST

```
{
  "type": "TASK",
  "user": 1,
  "dateTimeFrom": "2017-09-15T08:30:00+01:00"
}
```

Response:

```
{ "changelog" :  
[{"deletedWith":"","modification":"ATTACH","newValueClass":"timeentry","newValueDisplay":"Jacob Smith, CTE\nod  
[[2017-09-15T07:30:00.000Z]]\n","newValued":"50007","newValuesDictionary":false,"  
objectClass":"user","objectId":1,"objectLabel":"Jacob  
Smith","oldValuesDictionary":false,"property":"timeEntry"}]  
}
```

In response a **"newValued":"50007"** is the ID of a newly created work time

Creating new work time **just with a finish time:**

<https://system.firmao.net/{organizationID}/svc/v1/timeentries>

POST

```
{  
  "type":"TASK",  
  "user":1,  
  "dateTimeTo":"2017-09-15T16:30:00+01:00"  
}
```

Response:

```
{ "changelog" :  
[{"deletedWith":"","modification":"ATTACH","newValueClass":"timeentry","newValueDisplay":"Jacob Smith,  
CTE\ndo[[2017-09-15T16:30:00.000Z]]\n","newValued":"50008","newValuesDictionary":false,"objectClass":"user","objectId":1,"objectLabel":"Jacob  
Smith","oldValuesDictionary":false,"property":"timeEntry"}]  
}
```

In response a **"newValued":"50008"** is the ID of a newly created work time

Finishing **existing** work time

<https://system.firmao.net/{organizationID}/svc/v1/timeentries/50007> PUT

```
{  
  "dateTimeTo":"2017-09-15T11:30:00+01:00"  
}
```

```
}
```

Response:

```
{"changelog":[{"deletedWith":"","modification":"CHANGE","newValueClass":"timeentry","newValueDisplay":"Emma Clark, LEAVE {[2017-09-14T22:00:00.000Z]} - {[2017-09-15T10:30:00.000Z]}", "newValueId":"-1","newValuesDictionary":false,"objectClass":"user","objectId":6,"objectLabel":"Emma Clark","oldValueClass":"timeentry","oldValueDisplay":"Emma Clark, LEAVE {[2017-09-14T22:00:00.000Z]} - {[2017-09-15T22:00:00.000Z]}", "oldValueId":"-1","oldValuesDictionary":false,"property":"timeEntry"}]}
```

Creating a new work time with **a start and finish time:**

<https://system.firmao.net/{organizationID}/svc/v1/timeentries>

POST

```
{
  "type":"TASK",
  "user":1,
  "dateTimeFrom":"2017-09-15T08:30:00+01:00",
  "dateTimeTo":"2017-09-15T12:00:00+01:00"
}
```

user - ID of the user in Firmao system (number) for which we create a work time

Response:

```
{"changelog":[{"deletedWith":"","modification":"ATTACH","newValueClass":"timeentry","newValueDisplay":"Jacob Smith, TASK {[2017-09-15T07:30:00.000Z]} - {[2017-09-15T11:00:00.000Z]}", "newValueId":"50037","newValuesDictionary":false,"objectClass":"user","objectId":1,"objectLabel":"Jacob Smith","oldValuesDictionary":false,"property":"timeEntry"}]}
```

7. Getting PDF files

a) Invoices

[https://system.firmao.net/{organizationID}/svc/v1/reports/invoices/transaction/Invoice{file_name}.pdf?original=true&id=1](https://system.firmao.net/{organizationID}/svc/v1/reports/invoices/transaction/<u>Invoice</u>{file_name}.pdf?original=true&id=1)

Where Invoice is the name of the used template and file name is a name that the generated file will have

b) Quote/Orders/Agreements

[https://system.firmao.net/{organizationID}/svc/v1/reports/invoices/offer/Quote{file_name}.pdf?id=1](https://system.firmao.net/{organizationID}/svc/v1/reports/invoices/offer/<u>Quote</u>{file_name}.pdf?id=1)

Where Quote is the name of the used template and file name is a name that the generated file will have.

8. Response codes

API uses default HTTP error codes:

2xx - correct operation

4xx - error occurred

5xx - error occurred

Detailed description of some errors:

Correct request:

200 - correctly gotten or edited errors

201 - correct object creation

Wrong request:

400 - server internal error
401 - request requires authentication
403 - no access permissions
404 - resource not found
409 - conflict, operation cancelled
500 - unexpected error
501 - function not implemented
502 - proxy server error

In case of errors, by default an intelligible description of the error appears.
For example:

```
{"errorCode":"entityNotFoundForId","field":"project","rejectedValue":"50","additionalInfo":{"id":"50","cls":"project"}}
```

9. Sample PHP scripts

WARNING: When copying code from PDF, it often cuts off hyphens, or adds unnecessary newline characters.

a) Adding a company

```
<?php
```

```
$usr = base64_encode("login:password");  
$url = 'https://system.firmao.net/{organizationID}/svc/v1/customers';
```

```
$content = array(  
"bankAccountNumber" => "50 1020 5558 1111 1702 0600 0067",  
"correspondenceAddress.city" => "New York",  
"correspondenceAddress.country" => "USA",  
"correspondenceAddress.postCode" => "10001",  
"correspondenceAddress.street" => "143 Parker Ave",  
"customerGroups" => array(),  
"customerType" => "PARTNER",  
"description" => "Company specializing in computer service",  
"emails" => array("compervice.firmao@mailinator.com",  
"computers@mailinator.com"),  
"employeesNumber" => "1",  
"faxNumber" => "22 398 78 87",
```

```

"label" => "COMP-SERVICE",
"name" => "COMP-SERVICE John Smith",
"nipNumber" => "7291077843", "officeAddress.city"
=> "New York", "officeAddress.country" => "USA",
"officeAddress.postCode" => "10001",
"officeAddress.street" => "143 Parker Ave",
"ownership" => "PRIVATE", "phones" =>
"223457426",
"status" => null,
"tags" => array(),
"voivodeship" => null,
"website" => "www.compservice.bis");

$json = json_encode($content);
echo $json.'<br /><nr />';
$curl = curl_init($url);

curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($curl, CURLOPT_HEADER, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array("Content-type: application/json;
charset=UTF-8", "Authorization: Basic $usr")); curl_setopt($curl, CURLOPT_POSTFIELDS,
$json);

//accepting certificates
curl_setopt ($curl, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt ($curl, CURLOPT_SSL_VERIFYPEER, 0);

$json_response = curl_exec($curl);
$status = curl_getinfo($curl, CURLINFO_HTTP_CODE); if (
$status != 201 && $status != 200) {
    die("Error: call to URL $url failed with status $status, response $json_response,
curl_error " . curl_error($curl) . ", curl_errno " . curl_errno($curl)); }

curl_close($curl);
$response = json_decode($json_response, true); ?>

```

b) Getting company data

```
<?php
$usr = base64_encode("login:password");
$url = 'https://system.firmao.net/{organizationID}/svc/v1/customers';

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "GET");
curl_setopt($curl, CURLOPT_HEADER, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array("Content-type: application/json;
charset=UTF-8", "Authorization: Basic $usr")); curl_setopt($curl, CURLOPT_POSTFIELDS,
$json); //accepting certificates

curl_setopt ($curl, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt ($curl, CURLOPT_SSL_VERIFYPEER, 0);
$json_response = curl_exec($curl);
$status = curl_getinfo($curl, CURLINFO_HTTP_CODE); if (
$status != 201 && $status != 200) {
die("Error: call to URL $url failed with status $status, response $json_response,
curl_error " . curl_error($curl) . ", curl_errno " . curl_errno($curl)); }
curl_close($curl);
echo $json_response;
$response = json_decode($json_response, true); ?>
```

c) Adding invoice

```
<?php

$usr = base64_encode("login:password");
$url = 'https://system.firmao.net/{organizationID}/svc/v1/transactions';
```

```

$content = array(
"transactionEntries"=>array(array(
  "classificationNumber"=>"",
  "discount"=>null,
  "productCode"=>"ST0001",
  "quantity"=>1,
  "unit"=>null,
  "vatPercent"=>0,
  "product"=>51,
  "unitNettoPrice"=>"100.00"
)),
"transactionNettoPrice"=>"1000.00",
"transactionBruttoPrice"=>"1000.00",
"transactionVatPrice"=>"0.00",
"invoiceAnnotations"=>null,
"paid"=>false,
"currency"=>"USD",
"paymentType"=>"CASH",
"restOfPaidValue"=>1000,
"paidValue"=>0,
"invoiceDate"=>"2012-08-08",
"paymentDate"=>"2012-08-08",
"transactionDate"=>"2012-08-08",
"type"=>"INVOICE",
"mode"=>"SALE",
"customerAddress.postCode"=>"91477",
"customer"=>50,
"customerAddress.city"=>"New York",
"customerAddress.country"=>"USA",
"customerAddress.street"=>"143 Island Ave",
"nipNumber"=>"1060000062",
"issuingPerson"=>"Jacob Smith",
"transactionNumber"=>"12/8/2012"
);
$json = json_encode($content);
echo $json.'  
<br /><nr />';
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($curl, CURLOPT_HEADER, false);

```

```

curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER,array("Content-type:application/json;
charset=UTF8","Authorization: Basic $usr"));
curl_setopt($curl, CURLOPT_POSTFIELDS, $json);
//akceptowanie certyfikatów
curl_setopt ($curl, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt ($curl, CURLOPT_SSL_VERIFYPEER, 0);
$json_response = curl_exec($curl);
$status = curl_getinfo($curl, CURLINFO_HTTP_CODE); if (
$status != 201 && $status != 200) {
die("Error: call to URL $url failed with status $status, response $json_response,
curl_error " . curl_error($curl) . ", curl_errno " . curl_errno($curl)); }
curl_close($curl);
$response = json_decode($json_response, true); ?>

```

d) Uploading PHP file < 7.0

```
<?php
```

```

$usr = base64_encode("login:haslo");
$url =
'https://system.firmao.net/{organizationID}/svc/v1/{object}/{id}/documentsUpload' ;

    $curl = curl_init($url);
    $parameters = array(
        'description'=>date('Y-m-d'),
        'file'=>'@'. 'c:\test.txt');

    curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "POST");
    curl_setopt($curl, CURLOPT_POSTFIELDS, $parameters);
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl,
CURLOPT_HTTPHEADER,array("Content-type:multipart/form-data;
charset=UTF8","Authorization: Basic $usr"));

```

```

    curl_setopt($curl,  CURLOPT_SSL_VERIFYHOST,  0);
    curl_setopt($curl,  CURLOPT_SSL_VERIFYPEER,  0);
    $json_response = curl_exec($curl);
    $status = curl_getinfo($curl, CURLINFO_HTTP_CODE); if
    ($status != 201 && $status != 200) {
        die("Error: call to URL $url failed with status $status, response $json_response,
curl_error " . curl_error(
            $curl
        ) . ", curl_errno " . curl_errno($curl));
    }
    curl_close($curl);
    $response = json_decode($json_response, true);
?>

```

e) Uploading PHP file >= 7.0

```

<?php

$usr = base64_encode("login:haslo");
$url =
'https://system.firmao.net/{organizationID}/svc/v1/{object}/{id}/documentsUpload" ;

    $curl = curl_init($url);

    $parameters = array(
        'file' => new CURLFile("$filePath")
    );

    curl_setopt($curl, CURLOPT_CUSTOMREQUEST, 'POST');
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_HTTPHEADER, array("Authorization: Basic $usr"));
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
    curl_setopt($curl,  CURLOPT_SSL_VERIFYHOST,  0);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);

    $json_response = curl_exec($curl);
    $status = curl_getinfo($curl, CURLINFO_HTTP_CODE);

```

```
if ($status != 201 && $status != 200) {
    die("Error: call to URL $url failed with status $status, response $json_response,
curl_error " . curl_error(
    $curl
    ) . ", curl_errno " . curl_errno($curl));
}
curl_close($curl);
$response = json_decode($json_response, true);
?>
```

10. More API options

If you need more API options or have any issues with the configuration email us at contact@firmao.net