Advisors Application Program Interface (API)

The URL generally has the form /api/<resource>/<action>, which could either retrieve a result or perform some action.

When retrieving a result, the API expects a HTTP GET method, and when performing an action it expects a HTTP POST.

The resource /api/advisors refer to all the advisors and brokerages in the database. To address a specific brokerage, use a URL of the form /api/advisors/brokerages/{brokerageCode}/<action>.

Successful results are returned as HTTP 200 or 201 responses, with the body containing JSON data. Defined error responses will be in the HTTP 400 range, and undefined server errors in the HTTP 500 range.

Error responses contain the following JSON format:

```
{
   "errors" : [
      {
         "description" : "<error description>",
         "code" : "<error code>",
         "index" : <integer>
      }
   ]
}
```

where the error description is a detailed description of the error, and the error code is a short reference of the error.

Index is an optional error field and is present when the request data is an array of objects. It indicates the position of the offending object in the request array.

FINWORKS
INFORMATION TECHNOLOGY SOLUTIONS

Supported API calls

1. Brokerage fee statement 3
2. Advisor details 4
3. Brokerage details 5
4. Advisor clients details 5
5. Brokerage clients details 9
6. Advisors 12
7. Create an advisor 10
8. Create a brokerage 11
9. Advisor fee report 15
10. Brokerage fee report 16
11. Assets Under Management (AUM) graph data points - Advisor 17
12. Assets Under Management (AUM) graph data points - Brokerage 19

1. Brokerage fee statement

GET:

/api/advisors/brokerages/{brokerageCode}/fee_statement?start_date={startDate}&end_date={endDate}&add_password={boolean}'

Content-Type: application/pdf

Example request:

/api/advisors/brokerages/BRK123/fee_statement?start_date=2005-01-01&end_date=2005-01-31&add_password=true'

JSON response:

HTTP: 200

Content-Type: application/pdf

Notes:

Response contains pdf binary data.

URLs must be *percent encoded*, eg. space = %20, comma = %2C

All fields are mandatory except for 'add_password'

add_password = 'true' or 'false'

if add_password is not specified, a password will be added to the document based on the value of the "Password protect PDF documents?" configuration setting.

Error examples:

HTTP response	Code	Description
400	Malformed request	Invalid date
500	BrokerageNotFound	Could not find a brokerage with with code: AnInvalidCode

2. Advisor details

GET: /api/advisors/{advisorCode}

Content-Type: application/json

Example request:

/api/advisors/00000001

Example JSON response:


```
{
        "Advisor Name": "AnAdvisor",
        "Advisor Code": "00000001",
        "Advisor Entity Unique Id": 1234567890,
        "Brokerage Code": "10000000",
        "Brokerage Name": "aBrokerage",
        “Brokerage Entity Unique Id": 0987654321
        “Contact Method” : “EMAIL”,
        “Email Address” : “anAdvisor@email.com”,
        “License Categories” :
        [
                {
                        “Category Number” : “1”,
                        “Category Name” : “Category1”,
                        “Can Get Mandate” : “true”,
                        “Sub Category Number” : “1”,
                        “Sub Category Name” : “Insurance Category”
                }
        ]
}
```


Error examples:

HTTP response	Code	Description
404	AdvisorNotFound	Could not find an advisor with code: AnInvalidCode

3. Brokerage details

GET:

Fetch all brokerages:

/api/advisors/brokerages/

Fetch information on one brokerage:

/api/advisors/brokerages/{brokerageCode}

4. Advisor clients details

GET:

Fetch by number of investors:

/api/advisors/{advisorCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}

Fetch using a sort by field:

/api/advisors/{advisorCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}&sort_by={sortByField}

Fetch by using number of records per page and requested page number:

/api/advisors/{advisorCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}&records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

Fetch using sort by, number of records per page and requested page number:

/api/advisors/{advisorCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}&sort_by={sortByField}&records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

As of the April 2026 release, **pagination is enforced** and a maximum of 100 records will be returned per page when number_of_investors=All, despite whether records_per_page and page_number is included or not.

If the {recordsPerPage} request exceeds 100, the results returned will be limited to 100.

Including sort_by={sortByField} **does not enforce pagination** and will return all client records if requested. To reduce the number of records requested and time required to complete the call, include the pagination parameters in the call:

records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

If a page_number that does not exist is called, an error message will be returned stating 'Invalid page number requested \ "{requestedPageNumber}\". Total pages is \"{value}\'

Content-Type: application/json

Example request:

/api/advisors/00000001/clientDetails?currency=ZAR&number_of_investors=all

Notes for optional parameters:

- &sort_by= can be one of: InvestorName, MarketValue, IRR. MarketValue is default if not specified.
    - Market Value is sorted by investor's total market value across all contracts

  example:
  /api/advisors/00000001/clientDetails?currency=ZAR&number_of_investors=all&sort_by=InvestorName

- &records_per_page= and &page_number= must be specified together. only if both are present will pagination for the result be performed

  example:

  /api/advisors/AGFSPB/clientDetails?currency=ZAR&number_of_investors=All&records_per_page=2&page_number=2

  The following will be added to the output json if pagination is requested:

  "Pagination":{

  "Total Pages": 2,

  "Total Records": 3,

  "Records Per Page": 2,

  "Current Page": 2}

Sorting and pagination can both be requested, as well as limiting the number of records returned using &number_of_investors parameter, as before. The number of records returned per page is limited to 100.

This also applies for the brokerage client details call.

Example JSON response:


```
{
        "Advisor Name": "AnAdvisor",
        "Advisor Code": "00000001",
        "Advisor Entity Unique Id": 1234567891,
        "Brokerage Entity Unique Id": 456789870,
        "Brokerage Code": "000000",
        "Brokerage Name": "ABrokerage",
        "Number of Active Contracts": 4,
        “Investor Details” :
        [
                {

                        “Entity Unique Id” : 1234567891
                        “Client Number” : “1234567891”,
                        “Investor Name” : “Piet Piet”,
                        "Market Value In Reporting Currency": {
                                "type": "Money",
                                "currency": "ZAR",
                                "value": "100.00"
                                },

                “Contracts” :
                        [
                                {
                                        “Unique Id” : 1234567890,
                                        “Account Number” : “1”,
                                        “Account Name” : “Piet”,
                                        “Account Description” : “A description”,
                                        "Market Value In Reporting Currency": {
                                                                "type": "Money",
                                                                "currency": "ZAR",
                                                                "value": "100.00"
                                                                },
                                        "Market Value In Preferred Currency": {
                                                                "type": "Money",
                                                                "currency": "ZAR",
                                                                "value": "100.00"
                                                                },
                                        "IRR Since Inception": {
                                            "type": "Percentage",
                                            "value": "0.00"}
                                }
                        ]
                }
        ]
}
```

Notes:

The “number_of_investors” field in the get request limits the number of investors returned per call. If this field is set to “all” then it will simply return all of the investors linked to the specified advisor. Setting the field to 10, or any other number, will simply only return 10 investors linked to the specified advisor.

Please refer to sort_by parameters to receive requested data in a specified order.

The IRR Since Inception field may have the string value ‘Unable to calculate’ if the IRR calculation doesn’t converge; eg:

"Contracts": [
{"Market Value In Reporting Currency": {"type": "Money", "currency": "ZAR", "value": "-2945.81"},
"IRR Since Inception": "Unable to calculate",
"Contract Number": "POL2",
"Unique Id": 433675776,
"Account Description": "Discretionary",
"Account Name": "",
"Account Number": "POL2",
"Market Value": "(R 2,945.81)",
"Market Value In Preferred Currency": {"type": "Money", "currency": "ZAR", "value": "-2945.81"}}]

Error examples:

HTTP response	Code	Description
404	AdvisorNotFound	Could not find an advisor with code: AnInvalidCode
400	InvalidCurrency	Could not match currency identifier: ASD with an existing currency in the system
400	MalformedRequest	Invalid page number requested \"39\". Total pages is \"38\"

5. Brokerage clients details

GET:

Fetch by number of investors:

/api/advisors/brokerages/{brokerageCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}

Fetch using a sort by field:

/api/advisors/brokerages/{brokerageCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}&sort_by={sortByField}

Fetch by using number of records per page and requested page number:

/api/advisors/brokerages/{brokerageCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}&records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

Fetch using sort by, number of records per page and requested page number:

/api/advisors/brokerages/{brokerageCode}/clientDetails?currency={currency}&number_of_investors={numberOfInvestors}&sort_by={sortByField}&records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

Content-Type: application/json

Example request:

/api/advisors/brokerages/00000001/clientDetails?currency=ZAR&number_of_investors=all

See details about pagination and sorting on the **Advisor clients details** call above.

Example JSON response:


```
{
        "Brokerage Name": "ABrokerage",
        "Brokerage Code": "00000001",
        "Brokerage Entity Unique Id": 1234567890,
        "Number of Active Contracts": 4,
        “Investor Details” :
        [
                {
                        “Entity Unique Id” : 1234567891,
                        “Client Number” : “1234567891”,
                        “Investor Name” : “Piet Piet”,
                        “Market Value” : “R 100.00”,
                        "Market Value In Reporting Currency": { "type": "Money",
```

```
                "currency": "ZAR",
                "value": "100.00"
                },
        "Market Value In Preferred Currency": {
                "type": "Money",
                "currency": "ZAR",
                "value": "100.00"
                },
        “Contracts” :
        [
                {
                        “Unique Id” : 1234567890,
                        “Account Number” : “1”,
                        “Account Name” : “Piet”,
                        “Account Description” : “A description”,
                        “Market Value” : “R 100.00”,
                        "Market Value In Reporting Currency": {
                                        "type": "Money",
                                        "currency": "ZAR",
                                        "value": "100.00"
                                        },
                        "Market Value In Preferred Currency": {
                                        "type": "Money",
                                        "currency": "ZAR",
                                        "value": "100.00"
                                        },
                        "IRR Since Inception": {"type": "Percentage", "value": "0.00"}
                }
        ]
    }
  ]
}
```


Notes:

As of the April 2026 release, **pagination is enforced** and a maximum of 100 records will be returned per page when **number_of_investors=All**, despite whether records_per_page and page_number is included or not.

If the {recordsPerPage} request exceeds 100, the results returned will be limited to 100.

Including sort_by={sortByField} **does not enforce pagination** and will return all client records if requested. To reduce the number of records requested and time required to complete the call, include the pagination parameters in the call: records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

If a page_number that does not exist is called, an error message will be returned stating ‘Invalid page number requested \ “{requestedPageNumber}\”. Total pages is \”{value}\’

The “number_of_investors” field in the get request limits the number of investors returned per call. If this field is set to “all” then it will simply return all of the investors linked to the specified advisor. Setting the field to 10, or any other number, will simply only return 10 investors linked to the specified advisor.

Please refer to sort_by parameters to receive requested data in a specified order.

Error examples:

HTTP response	Code	Description
404	BrokerageNotFound	Could not find a brokerage with code: AnInvalidCode
400	InvalidCurrency	Could not match currency identifier: ASD with an existing currency in the system

6. Advisors

Answer a list of advisors.

GET:

/api/advisors

Content-Type: application/json

Note: This call returns all the active advisors.

Example JSON response:

HTTP: 200


```
{
	"data": [
		{
			"Code": "AGFSPB",//To be removed
			"Advisor Code": "AGFSPB",
			"Name": "Piet Pompies",//To be removed
			"Advisor Name": "Piet Pompies",
			"Entity": "/api/entities/15555072",//To be removed
			"Advisor Entity Unique Id": 15555072,
			"Brokerage Code": "000017",
			"Brokerage Name": "Good Advice Brokers",
			"Brokerage Entity Unique Id": 738597888,
			"Brokerage": "/api/entities/738597888" //To be removed
		}
	]
}
```

7. Create an advisor

Create an advisor.

POST:

/api/advisors/

Example JSON request with Id number:

```
{
        "idNumber" : "1234567890",
        "advisorCode": ABC",
        "brokerageCode" : "BROKERAGE01"
}
```

Example JSON request with entity unique id:

```
{
        "entityUniqueId" : "234602742",
        "advisorCode": ABC",
        "brokerageCode" : "BROKERAGE01"
}
```

Example JSON request where we need to create the person as well:

```
{
        "surname" : "Nel",
        “idNumber” : “xxx”,
        "identificationDocumentType": "Identity Document",
        "identificationDocumentCountry":"NA",
        "nationality":"NA",
        “title” : “Mr”,
        “firstNames” : “Jan”,
        "advisorCode": ABC",
        "brokerageCode" : "BROKERAGE01"
}
```

Note the following regarding this call:

1. If an ID number provided already exists on Finworks as a person, then the API will add an advisor role to the existing person. It will match on ID number only and won’t use any of the other personal details provided (name, surname, title) since the person already exists.
2. The advisor code provided has to be unique else an error will be returned “Advisor already exists”
3. The brokerage code has to be an existing brokerage
4. identificationDocumentCountry and nationality requires the ISO code as set up in System -> Countries
5. We will check ID number validity for South African ID’s and will not create the person if not a valid South African ID in the following scenarios:
    a. If identificationDocumentType is Identity Document and

identificationDocumentCountry is ZA

b. If both identificationDocumentType and identificationDocumentCountry has not been provided and the System -> Configuration -> Default Country is South Africa

6. If identificationDocumentType and identificationDocumentCountry and Nationality is not provided, then defaults are as follows:
    a. identificationDocumentType is Identity Document
    b. identificationDocumentCountry is the Default Country on System -> Config
    c. Nationality is the Default Country on System -> Config

8. Create a brokerage

Create a brokerage.

POST:

/api/advisors/brokerages

Example JSON request with registration number:

```
{
	"registrationNumber" : "1234567890",
	"brokerageCode" : "BROKERAGE01"
}
```

Example JSON request with entity unique id:

```
{
	"entityUniqueId" : "234602742",
	"brokerageCode" : "BROKERAGE01"
}
```

Example JSON request where we need to create the company as well:

```
{
	"companyName" : "Test Company",
	“companyType” : “Listed company”,
	“registrationNumber” : “123553246”,
	"brokerageCode" : "BROKERAGE01"
}
```

Note the following regarding this call:

1. If a registration number provided already exists on Finworks as a company, then the API will add a brokerage role to the existing company. It will match on registration number only and won’t use any of the other personal details provided (company name, company type) since the company already exists.
2. The brokerage code provided has to be unique else an error will be returned “Brokerage already exists”

9. Advisor fee report

GET:

/api/advisors/{advisorCode}/fee_report?from_date={fromDate}&to_date={toDate}

Content-Type: application/json

Example request:

/api/advisors/000001/fee_report?from_date=2025/01/01&to_date=2025/02/01

Example JSON response:


```
[{
        "Account Number" : "ABC123",
        "Adviser Fee Period" : "2025/01/01 to 2025/01/30",
        "Adviser code" : "000001",
        "Adviser name" : "Doe J",
        "Amount" : "R 0.08",
        "Fee Type" : "Ongoing Adviser Fee",
        "Instrument" : "A Unitised Fund Name",
        "Instrument Code" : "AUFN",
        "Investor ID" : "11223344556677",
        "Investor Name" : "Pompies, P",
        "Platform Provider" : "APlatform Provider",
        "Price" : "1.0000",
        "Product" : "AProduct",
        "Transaction Date" : "2025-01-10",
        "Units" : "0.0800"
}]
```


Error examples:

HTTP response	Code	Description
404	AdvisorNotFound	Could not find an advisor with code: AnInvalidCode
400	Malformed request	Invalid date

10. Brokerage fee report

GET:

/api/advisors/brokerages/{advisorCode}/fee_report?from_date={fromDate}&to_date={toDate}

Content-Type: application/json

Example request:

/api/advisors/brokerages/12345678/fee_report?from_date=2025/01/01&to_date=2025/02/01

Example JSON response:


```
[{
        "Account Number" : "ABC123",
        "Adviser Fee Period" : "2025/01/01 to 2025/01/30",
        "Adviser code" : "000001",
        "Adviser name" : "Doe J",
        "Amount" : "R 0.08",
        "Fee Type" : "Ongoing Adviser Fee",
        "Instrument" : "A Unitised Fund Name",
        "Instrument Code" : "AUFN",
        "Investor ID" : "11223344556677",
        "Investor Name" : "Pompies, P",
        "Platform Provider" : "APlatform Provider",
        "Price" : "1.0000",
        "Product" : "AProduct",
        "Transaction Date" : "2025-01-10",
        "Units" : "0.0800"
}]
```


Error examples:

HTTP response	Code	Description
404	BrokerageNotFound	Could not find a brokerage with code: AnInvalidCode
400	Malformed request	Invalid date

11. Assets Under Management (AUM) graph data points - Advisor

Returns the datapoint for the Assets Under Management (AUM) graph linked to the set advisor code

GET:

api/advisors/{advisorCode}/aum?from_date={fromDate}&to_date={toDate}

With pagination

GET:

api/advisors/{advisorCode}/aum?from_date={fromDate}&to_date=2026-02-18&records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

Content-Type: application/json

Example request:

api/advisors/20123%2F18547/aum?from_date=2025-01-01&to_date=2026-02-18

With pagination

api/advisors/20123%2F18547/aum?from_date=2025-01-01&to_date={toDate}&records_per_page=100&page_number=1

Example JSON response:


```
{
   "Brokerage Entity Unique Id": 2103210432310,
   "Advisor Name": "The Advisor",
   "AUM data": [
      {
         "Date": "2025-01-01",
         "AUM": 5991963.77
      },
      ..............
      {
         "Date": "2026-02-18",
         "AUM": 7575479.65
      }
   ],
   "Advisor Entity Unique Id": 4323102103210,
   "Brokerage Code": "112233",
   "Advisor Code": "20123/18547",
   "Brokerage Name": "The Brokerage"
}
```

Notes:

To escape a forward slash "/" in a URL, you should replace it with its percent-encoded equivalent: %2F.

To represent Advisor Code "20123/18547" use "20123%2F18547"

Error examples:

HTTP response	Code	Description
404	AdvisorNotFound	Could not find an advisor with code: AnInvalidCode
400	Malformed request	Invalid date

12. Assets Under Management (AUM) graph data points - Brokerage

Returns the datapoint for the Assets Under Management (AUM) graph linked to the set brokerage code

GET:

api/advisors/brokerages/{brokerageCode}/aum?from_date={fromDate}&to_date={toDate}

With pagination

GET:

api/advisors/brokerages/{brokerageCode}/aum?from_date={fromDate}&to_date={toDate}&records_per_page={numberOfRecordsPerPage}&page_number={requestedPageNumber}

Content-Type: application/json

Example request:

api/advisors/brokerages/112233/aum?from_date=2025-01-01&to_date=2026-02-18

With pagination

api/advisors/brokerages/112233/aum?from_date=2025-01-01&to_date=2026-02-18&records_per_page=100&page_number=1

Example JSON response:


```
{
   "Brokerage Entity Unique Id": 2103210432310,
   "AUM data": [
      {
         "Date": "2025-01-01",
         "AUM": 5991963.77
      },
      .....
      {
         "Date": "2026-02-18",
         "AUM": 7575479.65
      }
   ],
   "Brokerage Code": "112233",
   "Brokerage Name": "The Brokerage"
}
```

Error examples:

HTTP response	Code	Description
404	BrokerageNotFound	Could not find a brokerage with code: AnInvalidCode
400	Malformed request	Invalid date