

PUBLICAPI DOCUMENT (V2)

1. Get country List

URL: /api/v2/countrylist

Method: GET

CURL: curl -X GET /api/v2/countrylist
--header 'x-api-key: API_KEY'
--header 'Authorization: Bearer TOKEN'

2. Get Currency List

URL: /api/v2/currencylist

Method: GET

CURL: curl -X GET /api/v2/currencylist
--header 'x-api-key: API_KEY'
--header 'Authorization: Bearer TOKEN'

3. Get Users List (For Dynamic custom fields)

URL: /api/v2/team_members

Method: GET

CURL: curl -X GET /api/v2/team_members
--header 'x-api-key: API_KEY'
--header 'Authorization: Bearer TOKEN'

4. Get Teams List (For Dynamic custom fields)

URL: /api/v2/teams

Method: GET

CURL: curl -X GET /api/v2/teams
--header 'x-api-key: API_KEY'
--header 'Authorization: Bearer TOKEN'

5. Get Firm types

URL: /api/v2/firmtypes

Method: GET

CURL: curl -X GET /api/v2/firmtypes
--header 'x-api-key: API_KEY'
--header 'Authorization: Bearer TOKEN'

6. Get Tags

URL: /api/v2/tags

Method: GET

CURL: curl -X GET /api/v2/tags
--header 'x-api-key: API_KEY'
--header 'Authorization: Bearer TOKEN'

Method: POST

Request Body: [{
 "name":String,
 "type": String,
}]

Description: Create new tags

Tag type - assetallocation, conviction, InvestmentThesis, WatchList, Geography, Investor, Question, Relationship, Contact, Attachments, DocumentGroups

7. Get Strategies

URL: /api/v2/strategies

Method: GET

CURL: curl -X GET /api/v2/strategies
 --header 'x-api-key: API_KEY'
 --header 'Authorization: Bearer TOKEN'

Method: POST

Request Body: {'strategies': [strings]}

Description: Create new strategy/strategies

8. Firms

URL: /api/v2/firms

- **Method:** POST
- **Description:** Create new Firms.
- **Example Request:**

```
{  
  "source_type": 1,  
  "data": [  
    {  
      "source_id": "771234",  
      "name": "name771",  
      "domain": "www.firm772.com",  
      "primary_contact_email": "dv1@name771.com",  
      "metadata": {  
        "firm_type": "administrator",  
        "alternate_name": "firm771-1",  
        "secondary_contact_emails": [],  
        "location": [  
          {  
            "phone_1": "9710346888",  
            "phone_2": "9710346999",  
            "street_2": "st24",  
            "zipcode": "360371",  
            "city": "city",  
            "is_headquarter": true,  
            "country": "norway",  
            "street_1": "st222",
```

```

        "state": "st",
        "fax": "998888"
    }
]
},
"custom_fields": {
    "dept name": "music"
}
}
]
}

```

- **CURL REQUEST:**

```

curl -X POST 'https://DV_API_URL/api/v2/firms' -H 'X-API-KEY:
d659f024d31645b1a1a5e29ed6661aa2' -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9' -H 'Content-Type: application/json'
--data-raw '{
    "source_type": 1,
    "data": [
        {
            "source_id": "771234",
            "name": "name771",
            "domain": "www.firm772.com",
            "metadata": {
                "firm_type": "administrator",
                "alternate_name": "firm771-1",
                "secondary_contact_emails": [],
                "location": [
                    {
                        "phone_1": "9710346888",
                        "street_2": "st24",
                        "zipcode": "360371",
                        "city": "city",
                        "is_headquarter": true,
                        "country": "norway",
                        "street_1": "st222",
                        "state": "st",
                        "fax": "998888"
                    }
                ]
            },
            "custom_fields": {
                "dept name": "music"
            }
        }
    ],
}'

```

- **Request Fields Explained:**

- **mandatory fields:** source_id, name, domain, primary_contact_email
- **optional fields:**
 - every field in `metadata` is optional
 - every field in `location` is optional
 - `metadata`, `location`, `custom_fields` can be empty

- **Fields Explained:**

- source_id: client's unique ID (string) (mandatory)
- name: firm name (string) (mandatory)
- domain: firm domain name (string) (mandatory)
- primary_contact_email: primary contact email for firm (string) (mandatory) [either primary_contact_email or domain is required]
- metadata - [details about firm]
- firm_type: type of firm: [Other, Administrator, Prime Broker, Custodian, Auditor, Marketer/Placement Agent, Board of Director, Asset Manager, Investor, Law Firm, Background Check Provider, Compliance Consultant, ODD Consultant]
- alternate_name: alternate name for firm (string)
- secondary_contact_emails: secondary emails for firm
- location - [contains multiple addresses for firm]
 - street_1 - address line 1
 - street_2 - address line 2
 - city - city
 - state - state
 - country - country
 - zipcode - zipcode
 - phone_1 - phone number 1
 - phone_2 - phone number 2
 - fax - fax number
 - is_headquarter - is this address headquarter or not
- custom_fields - [contains `fieldname` and the value of field]
 - { `fieldname`: `value of the field`,
 - `fieldname`: [multiple values]
 - }

- **Output:**

```
[
  {
    "source_type": 1,
    "dv_id": 47731,
    "source_id": "771234",
    "error": null,
    "status": "success",
    "address_ids": [999]
  }
]
```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- Status:
 - success: no error found; `dv_id` would be not null
 - fail: error found,
 - message will be in `error` key, in case of `fail` ;
 - `dv_id` would be null
 - partial_success:
 - error found in `secondary_contact_emails` and
 - `custom_fields` but firm in inserted;
 - `dv_id` would be not null

- **Method: PUT**

- **Description: Update existing firm**

- **Example Request:**

```
{
  "data": [
    {
      "source_id": "771234",
      "dv_id": 47731,
      "metadata": {
        "firm_type": "administrator",
        "alternate_name": "firm771-1",
        "secondary_contact_emails": [],
        "location": [
          {
            "address_id": 999,
            "phone_1": "9710346888",
            "phone_2": "9710346999",
            "street_2": "st24",
            "zipcode": "360371",
            "city": "city",
            "is_headquarter": true,
            "country": "norway",
            "street_1": "st222",
            "state": "st",
            "fax": "998888"
          }
        ]
      },
      "custom_fields": {
        "dept name": "music"
      }
    }
  ],
  "source_type": 1
}
```

- **CURL REQUEST:**

```
curl -X PUT 'https://DV_API_URL/api/v2/firms' -H 'X-API-KEY:
d659f024d31645b1a1a5e29ed6661aa2' -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9' -H 'Content-Type: application/json'
--data-raw '{
  "data": [
    {
      "source_id": "771234",
      "dv_id": 47731,
      "metadata": {
        "firm_type": "administrator",
        "alternate_name": "firm771-1",
        "secondary_contact_emails": [],
        "location": [
          {
            "address_id": 455,
            "phone_1": "9710346888",
            "phone_2": "9710346999",
            "street_2": "st24",
            "zipcode": "360371",
            "city": "city",
            "is_headquarter": true,
            "country": "norway",
            "street_1": "st222",
            "state": "st",
            "fax": "998888"
          }
        ]
      },
      "custom_fields": {
        "dept name": "music"
      }
    }
  ],
  "source_type": 1
}'
```

- **Request Fields Explained:**

- **mandatory fields:** source_id, dv_id
- **Fields Explained:**
 - dv_id: DiligenceVault Database unique ID returned in POST call
 - address_id: unique ID for address returned in POST call
- rest of the fields are same as POST call

- **Output:**

```
[
  {
    "source_type": 1,
    "dv_id": 47731,
    "source_id": "771234",
```

```

    "error": null,
    "status": "success",
    "address_ids": [999]
  }
]

```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- address_ids: unique ID's for each address provided
- status:
 - success: no error found; `dv_id` would be not null
 - Partial_success:
 - error found in `secondary_contact_emails` and `custom_fields` but firm in inserted;
 - `dv_id` would be not null
 - Fail:
 - error found, message will be in `error` key, in case of `fail` ;
 - `dv_id` would be null

9. PRODUCTS

Endpoint: /v2/products

- **Method:** POST
- **Description:** Create new Products.
- **Example Request:**

```

{
  "data": [
    {
      "source_id": "1114",
      "name": "prod1114",
      "strategy_id": 38101,
      "associated_firm_id": 1,
      "metadata": {
        "alternate_name": "alternate name prod",
        "description": "this product is about something",
        "internal_key": "PROD_11",
        "associated_primary_email": "hank@dcompany-domain.com",
        "secondary_contact_emails":
["hank1@dcompany-domain.com", "hank2@dcompany-domain.com"],
        "tag_id": [11],
        "relationship_status_id": 0
      },
      "custom_fields": {
        "dept name": ["sales"]
      }
    }
  ],
  "source_type": 1
}

```

```
}
```

- **CURL REQUEST:**

```
curl -X POST 'https://DV_API_URL/api/v2/products' -H 'X-API-KEY:
d659f024d31645b1a1a5e29ed6661aa2' -H 'Authorization: Bearer token' -H
'Content-Type: application/json'
--data-raw '{
  "data": [
    {
      "source_id": "1114",
      "name": "prod1114",
      "strategy_id": 38101,
      "associated_firm_id": 1,
      "metadata": {
        "alternate_name": "alternate name prod",
        "description": "this product is about something",
        "internal_key": "PROD_11",
        "associated_primary_email": "hank@dcompany-domain.com",
        "secondary_contact_emails": ["hank1@dcompany-
domain.com", "hank2@dcompany-domain.com"],
        "tag_id": [11],
        "relationship_status_id": 0
      },
      "custom_fields": {
        "dept name": ["sales"]
      }
    }
  ],
  "source_type": 1
},'
```

- **Request Fields Explained:**

- **mandatory fields:**
 - source_id, name, strategy_id, associated_firm_id
- **optional fields:**
 - every field in `metadata` is optional
 - `metadata`, `custom_fields` can be empty

- **Fields Explained:**

- source_id: client's unique ID (string) (mandatory)
- name: product name (string) (mandatory)
- strategy_id: strategy ID for product (string) (mandatory) [strategy id can be fetched from GET STRATEGY API]
- associated_firm_id: firm id with this product is associated with (string) (mandatory)

- metadata - [details about firm]
- alternate_name: alternate name for product (string)
- description: description about product (string)
- associated_primary_email: primary email for product
- secondary_contact_emails: secondary emails for product
- tag_id: tags id [tag id can be fetched from GET TAGS API]
- relationship_status_id:
- custom_fields - [contains `fieldname` and the value of field]
 { `fieldname`: `value of the field`,
 `fieldname`: [multiple values`]}

- **Output:**

```
[
  {
    "source_type": 1,
    "dv_id": 47731,
    "source_id": "771234",
    "error": null,
    "status": "success"
  }
]
```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- status:
 - success: no error found; `dv_id` would be not null
 - Partial_success:
 - error found `custom_fields` but product is inserted;
 - `dv_id` would be not null
 - fail: error found, message will be in `error` key, in case of `fail` ;
 - `dv_id` would be null

- **Method: PUT**

- **Description: Updated the existing product details**

- Only Fields inside `metadata` and `custom_fields` will be updated in PUT
- If `metadata` and `custom_fields` are empty : nothing will be updated in the existing product

- **Example Request:**

```
{
  "data": [
```



```

        "secondary_contact_emails": ["hank1@dcompany-
domain.com", "hank2@dcompany-domain.com"],
        "tag_id": [11],
        "relationship_status_id": 0
    },
    "custom_fields": {
        "dept name": ["sales"]
    }
},
],
"source_type": 1
}'

```

- **Request Fields Explained:**
 - **mandatory fields:**
 - dv_id, source_id, name, strategy_id, associated_firm_id
 - optional fields:
 - every field in `metadata` is optional
 - `metadata`, `custom_fields` can be empty
- **Fields Explained:**
 - dv_id: DiligenceVault Database unique ID returned in POST call
 - rest of the fields are same as POST call
- **Output Explained:**
 - dv_id: unique ID of DiligenceVault Database
 - source_id: unique ID of client's DB
 - error: if any error found, it will be returned here
 - status:
 -
 - success: no error found; `dv_id` would be not null
 - Fail:
 - error found, message will be in `error` key, in case of `fail` ;
 - `dv_id` would be null
 - Partial_success:
 - error found `custom_fields` but product in inserted;
 - `dv_id` would be not null

10. Contacts

- **Endpoint: /v2/contacts**
 - **Method: POST**
 - **Description: Create new contacts.**
 - **Example Request:**

```

{
  "source_type": 1,
  "data": [
    {
      "source_id": "8801",

```



```

        "street_2": "6th Floor",
        "city": "Boston",
        "state": "MA",
        "country": "United States",
        "zipcode": "02109",
        "phone_1": "123-4567-890",
        "phone_2": "123-456-7890",
        "fax": "123-456-7890"
    }
]
},
"custom_fields": { `fieldname`: `value of the field`,
                    `fieldname`: [multiple values`]}
}]
}'

```

- **Request Fields Explained:**

- **mandatory fields:**
 - source_id, email, associated_firm_id
- **optional fields:**
 - every field in `metadata` is optional
 - `metadata`, `custom_fields` can be empty
 -

- **Fields Explained:**

- source_id: client's unique ID (string) (mandatory)
- email: email ID of the contact (string) (mandatory)
- associated_firm_id: firm id this contact is associated with (string) (mandatory)
- custom_fields - [contains `fieldname` and the value of field]


```

      { `fieldname`: `value of the field`,
        `fieldname`: [multiple values`]}
      
```
- metadata - [details about firm]
 - first_name: first name of contact
 - last_name: last name of contact
 - title: title for contact [Ms,Mr,Mrs]
 - address: [multiple/single addresses for this contact]
 - street_1 - street address line 1
 - street_2 - street address line 2
 - city - city
 - state - state
 - country - country
 - zipcode - zipcode
 - phone_1 - phone number 1
 - phone_2 - phone number 2

- fax - fax

- **Output:**

```
[
  {
    "source_type": 1,
    "dv_id": 70520,
    "source_id": "8801",
    "error": null,
    "status": "success",
    "address_ids": []
  }
]
```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- address_ids: unique ID's for each address provided
- status:
 - success: no error found; `dv_id` would be not null
 - Partial_success:
 - `custom_fields` but contact in inserted;
 - `dv_id` would be not null
 - fail:
 - error found, message will be in `error` key, in case of `fail` ;
 - `dv_id` would be null

- **Method: PUT**

- **Description: Update existing contacts.**

- Only Fields inside `metadata` and `custom_fields` will be updated in PUT
- If `metadata` and `custom_fields` are empty : nothing will be updated in the existing product
- IF metadata and custom_fields keys are not present: OUTPUT will return with `fail` status
- **Email address cannot be updated once contacts is created successfully**

- **Example Request:**

```
{
  "source_type":1,
  "data": [
    {
      "source_id": "8801",
      "dv_id": 70520,
      "associated_firm_id": 1,
      "metadata": {
```



```

        "country": "United States",
        "zipcode": "02109",
        "phone_1": "123-4567-890",
        "phone_2": "123-456-7890",
        "fax": "123-456-7890"
    }
]
},
"custom_fields": { `fieldname`: `value of the field`,
                    `fieldname`: [ `multiple values` ]
}
}]
}'

```

- **Request Fields Explained:**

- **mandatory fields:**
 - dv_id, source_id, associated_firm_id, metadata, custom_fields
- **Fields Explained:**
 - dv_id: DiligenceVault Database unique ID returned in POST call
 - rest of the fields are same as POST call

- **Output:**

```

[
  {
    "source_type": 1,
    "dv_id": 70520,
    "source_id": "8801",
    "error": null,
    "status": "success",
    "address_ids": []
  }
]

```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- address_ids: unique ID's for each address provided
- status:
 - success: no error found; `dv_id` would be not null
 - fail: error found, message will be in `error` key, in case of `fail` ;
 - `dv_id` would be null
 - partial_success: `custom_fields` but contact in inserted;
 - `dv_id` would be not null

11. VEHICLE

- **Endpoint:** /v2/vehicles

- **Method:** POST
- **Description:** Create new vehicle
- **Example Request:**
 - {
"data": [
 {
 "source_id": "67890",
 "name": "vehi67890",
 "associated_fund_id": 4,
 "currency": "USD",
 "metadata": {
 "alternate_name": "vehic11",
 "inception_date": "2018-01-01",
 "internal_key": "VEHIC11"
 },
 "custom_fields": { `fieldname`: }
 },
],
"source_type": 1
}

- **CURL REQUEST:**

-

```
curl -X POST 'https://DV_API_URL/api/v2/vehicles' -H 'X-API-KEY:
d659f024d31645b1a1a5e29ed6661aa2' -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiE2MDMzODg5MjksIm5iZiI6
MTYwMzY0ODkyOSwianRpIjoibmMmYxZDktODNlYy00YmMzLTgyMTYtZW
M5NzA5MzYyMmZjIiwiaXNjaXZlbnNpdjkiOiJmZjY2VzcyIsInVzZXJfY2xhaW1zIjpw
NoIjpmYXZlbnNpdjkiOiJmZjY2VzcyIsInVzZXJfY2xhaW1zIjpw
MyLCJmaXZlbnNpdjkiOiJmZjY2VzcyIsInVzZXJfY2xhaW1zIjpw
fdHlwZSI6OTM2LCJhY2Nlc3N0eXBlaWQibGciOiJmZjY2VzcyIsInVzZXJfY2xhaW1zIjpw
OnJlYWQibGciOiJmZjY2VzcyIsInVzZXJfY2xhaW1zIjpw' -H 'Content-
Type: application/json'
--data-raw '{
  "data": [
    {
      "source_id": "67890",
      "name": "vehi67890",
      "associated_fund_id": 4,
      "currency": "USD",
      "metadata": {
        "alternate_name": "vehic11",
        "inception_date": "2018-01-01",
```

```

        "internal_key": "VEHIC11"
    },
    "custom_fields": { `fieldname`: }
}
],
"source_type": 1
}'

```

- **Request Fields Explained:**

- **mandatory fields:**
 - source_id, name, associated_fund_id, currency
- **optional fields:**
 - every field in `metadata` is optional
 - `metadata`,`custom_fields` can be empty

- **Fields Explained:**

- source_id: client's unique ID (string) (mandatory)
- name: name of the vehicle (string) (mandatory)
- associated_fund_id: product id this vehicle is associated with (string) (mandatory)
- currency: Currency Name
- custom_fields - [contains `fieldname` and the value of field]
{ `fieldname`: `value of the field`, `fieldname`: [multiple values`]}
- metadata - [details about firm]
 - alternate_name: alternate name for this vehicle
 - inception_date: date of inception for this vehicle
 - internal_key: any unique key you want to provide

- **Output:**

```

[
  {
    "source_type": 1,
    "dv_id": 8502,
    "source_id": "67890",
    "error": null,
    "status": "success"
  }
]

```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- status:
 - success: no error found; `dv_id` would be not null
 - partial_success: `custom_fields` but vehicle in inserted; `dv_id` would be not null
 - fail: error found, message will be in `error` key, in case of `fail` ;

`dv_id` would be null

- **Method: PUT**
- **Description: Update existing vehicle**
 - Only Fields inside `metadata` and `custom_fields` will be updated in PUT
 - If `metadata` and `custom_fields` are empty : nothing will be updated in the existing product
 - **Example Request:**

```

    {
      "data": [
        {
          "dv_id": 8502,
          "source_id": "67890",
          "name": "vehi67890",
          "currency": "USD",
          "metadata": {
            "alternate_name": "vehic11",
            "inception_date": "2018-01-01",
            "internal_key": "VEHIC11"
          },
          "custom_fields": { `fieldname`: }
        }
      ],
      "source_type": 1
    }
  ]
}

```

- **CURL REQUEST:**

```
curl -X PUT 'https://DV_API_URL/api/v2/vehicles' -H 'X-API-KEY:  
d659f024d31645b1a1a5e29ed6661aa2' -H 'Authorization: Bearer  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQoOjE2MDMzMzODg5MmksIm5iZil6MTYWw  
MzM4ODkyOSwianRpIjoibWxkZWktODNlYy00YmMzLTgyMTYtZWMyNTA5MAZYYy  
MmZjliliZXhwLjoxNmAzMzkwbzI5LCJpcGVudGl0eSI6MzIsImZyZXNoIjpmaXlwczZSwidHI  
wZSI6ImFjY2VzcylsbnVzZXJfY2haW1zlj7InVzZXJfaWQiOiJMaXJtX2lkIjoyLCJmaX  
JtX25lbHUoiOiJBcnRoYW5jZSBMTEMgaGMyISnVzZXJfdHlwZSI6OTM2LCJhY2Nlc3NOeXB1  
aWQioQjkyNSwicGVybWlzc2lvbnMiOiJMdmW5kOnJJYWQifX0.S5MUssccpKczdD2bfnyLORY-  
j-l5qwDEG6g43usf5ug' -H 'Content-Type: application/json'  
--data-raw '{
```

```
"data": [
  {
    "dv_id": 8502,
    "source_id": "67890",
    "name": "vehi67890",
    "currency": "USD",
    "metadata": {
      "alternate_name": "vehic11",
      "inception_date": "2018-01-01",
      "internal_key": "VEHIC11"
    },

```

```

        "custom_fields": { `fieldname`: 'value', `fieldname`: [ `multiple values` ] }
    },
    "source_type": 1
},

```

- **Fields Explained:**

- dv_id: DiligenceVault Database unique ID returned in POST call
- rest of the fields are same as POST call

- **Output:**

```

[
  {
    "source_type": 1,
    "dv_id": 8502,
    "source_id": "67890",
    "error": null,
    "status": "success"
  }
]

```

- **Output Explained:**

- dv_id: unique ID of DiligenceVault Database
- source_id: unique ID of client's DB
- error: if any error found, it will be returned here
- status:
 - success: no error found; `dv_id` would be not null
 - fail: error found, message will be in `error` key, in case of `fail` ;
`dv_id` would be null
 - partial_success: `custom_fields` but vehicle is inserted;
`dv_id` would be not null

CUSTOM FIELDS

- **Custom Fields:**

- **Description: Input/Output of custom fields in firm/product/contact/vehicle POST/PUT endpoints**
-

- **Types of custom fields and examples:**

Notes:

- 1 - If `Make this field mandatory` checkbox is ticked(on web):

Must provide that field in POST/PUT.

2 - If `Allow Multiple fields to be added` is checked values can be provided in an array.

- **Integer(int)**: for whole numbers without decimal
 - Eg for single value: {"custom_fields": {"fieldname":23}}
 - Multiple:{"custom_fields": {"fieldname":[23, 30]}}
- **Numeric(numeric)**: for numbers with decimal
 - single: {"custom_fields": {"fieldname":23.40}}
 - Multiple:{"custom_fields": {"fieldname":[23.40, 30.50]}}
- **Text(text)**: for text line
 - Eg for single value: {"custom_fields": {"fieldname":"random text"}}
 - Multiple:{"custom_fields": {"fieldname":["text1 ", "text 2"]}}
- **Paragraph(textmultiline)**: for text paragraph
 - Eg for single value: {"custom_fields": {"fieldname":"start of random paragraph just for the example..."}}
 - Multiple:{"custom_fields": {"fieldname":["start of a random paragraph just for the example...", "start of a random paragraph just for the example... again.."]}}
- **Dropdown(dropdown)**: for single selection of value
 - single: {"custom_fields": {"fieldname":"Option 1"}}
 - Multiple: Multiples are not allowed in dropdown
- **Link(link)**: for saving URL
 - single: {"custom_fields": {"fieldname":"<https://www.random.org>"}}
 - Multiple:{"custom_fields": {"fieldname":["<https://www.random.org>", "<https://www.something.com>"]}}
- **Checkbox(checkbox)**: For multiple selection (same as dropdown)
 - single: {"custom_fields": {"fieldname":"Option 1"}}
 - Multiple:{"custom_fields": {"fieldname":["Option 1", "Option 1"]}}

- **Dynamic:**

Note:

- 1 - In dynamic: values are the id's of corresponding types.
- 2 - Id's can be fetched from their GET API's.
- 3 - `multiple`, `required` field rules are applied as explained above.

Let's assume: ids are 23,45

- **Type of dynamic fields:**
 - **Contact(contact)**: GET /v2/contacts.
 - **Product(fund)**: GET /v2/products
 - **Project(diligence)**: GET /v1/diligence
 - **Vehicles(vehicles)**: GET /v2/vehicles
 - **Templates(templates)**: GET /v2/templates
 - **Team members(team_members)**:GET /v2/team_members

- **Teams (teams): GET** /v2/teams
- **Firms (firm): GET** /v2/firms (for investor)
- **Investors (investors): GET** /v2/firms (for manager)

- Example:
 - single: {"custom_fields": {"fieldname":23}}
 - Multiple:{"custom_fields": {"fieldname":[23,45]}}

- **OUTPUT for int, numeric, link, text, paragraph:**

```
{
  "custom_fields": {
    "f-text": {
      "type": "int",
      "has_multiple": false,
      "is_mandatory": false,
      "has_href": false,
      "href": "",
      "data": [{
        "value": 90,
        "dynamic_url":
"https://app.diligencevault.com/#/app/firms/1/funds/90/profile"
      }]
    }
  }
}
```

- **OUTPUT explained:**

- **fieldname** - name given by user while creating the field
- **has_multiple** - boolean value whether multiple values are allowed or not in the field
- **is_mandatory** - boolean value whether if field is marked mandatory or not
- **has_href** - boolean value whether **URL** is added in field
- **href** - URL added to the field, **`has_href`** would be **True** in this Case
- **data:** (array of values)
 - value - values you have entered.
 - Eg: 24, "text line", "paragraph", 34.5
 - **dynamic_url** - If user have ticked **`Allow users to add a URL`** while creating custom field
 - url automatically created based on the value provided
- **type:**
 - int - (integer)
 - numeric - (numeric)
 - text - (text)
 - textmultiline - (paragraph)
 - link - (link)

- **OUTPUT for Dynamic fields and dropdown/checkbox::**

```
{
  "custom_fields": {
    "fieldname": {
      "type": "contact",
      "has_multiple": false,
      "is_mandatory": false,
      "has_href": false,
      "href": "",
      "data": [
        {
          "id": 57,
          "value": "MollyPan"
        }
      ]
    }
  }
}
```

- **Output explained:**

- o fieldname - name given by user while creating the field
- o has_multiple - boolean value whether multiple values are allowed or not in the field
- o is_mandatory - boolean value whether if field is marked mandatory or not
- o has_href - boolean value whether **URL** is added in field
- o href - URL added to the field, **`has_href`** would be **True** in this case
- o type -
 - dropdown (Dropdown)
 - checkbox (Checkbox)
 - contact - Dynamic contact
 - product - Dynamic product
 - diligence - Dynamic project
 - templates - Dynamic templates
 - vehicles - Dynamic vehicle
 - team_members - Dynamic team members
 - teams - Dynamic teams
 - firm - Dynamic firms
 - investors - Dynamic investor
- o data - array of data:
 - id - id of the value you selected
 - value - name of the selected id