Onspring

PEOPLE-POWERED AUTOMATION®

v1 Application Programming Interface (API)

Administrator Guide

Revised: June 2024

IMPORTANT: This document is provided for reference only. All new integrations should utilize the v2 API.

Contents

v1 API Introduction

The Onspring Application Programming Interface (API) allows external programs to retrieve and save data within your Onspring instance. All interaction with the API occurs using the following base URL:

`https://api.onspring.com`

In order to successfully interact with the v1 API, the external program must provide an X-ApiKey header in each HTTP request, using the following format:

```
X-ApiKey: 000000fffff000000ffffff/00000000-ffff-0000-ffff-000000000000
```

The value for the X-ApiKey header must be obtained from an Onspring user with permissions to read API keys for your instance, using the following instructions:

1. Within Onspring, navigate to **Administration > Security > API Keys**.
2. On the list page, add a new API key (requires create permissions) or click an existing API key to view its details.
3. On the details page for an API key, expand the Developer Information section. The X-ApiKey header value may be copied from this section.

IMPORTANT: The API key must have a status of "Enabled." Additionally, each API key has an associated role that controls the permissions for requests made using that API key. An Onspring administrator may configure those permissions as they would any other role in Onspring.

v1 Software Development Kit (SDK)

If you are using .NET 4.6.1 or higher, you may want to use our SDK to interact with the Onspring v1 API. The SDK provides helpers that perform such tasks as formatting the calls to the API, returning strongly typed objects, and parsing errors. Because of this, if you are using the SDK, the details covered in the sections that follow will not be as important for you to understand. The home page for the SDK is:

`https://github.com/onspring-technologies/onspring-api-sdk`

The SDK can be included in a Visual Studio project via a Nuget package. The home page for the Nuget package is:

`https://www.nuget.org/packages/onspring.api.sdk`

Sample projects that use the SDK via the Nuget package are available at our Github page:

`https://github.com/onspring-technologies`

v1 API Request Overview

The requests that may be made to the v1 API are listed below. Each request has its own section later in this document. Parameters (shown between < and >) will usually be replaced with integer values in the actual requests. Some requests may have additional query string options, which are described in the section for that request. For requests that return data (except attachment or image files), the response body will be in JSON format.

What Would You Like to Do?	URL or Minimum URL Template
Verify connectivity	`GET /v1/ping`
Get the list of apps and surveys	`GET /v1/apps`
Get the fields for an app/survey	`GET /v1/fields?appId=<appId>`
Get one field	`GET /v1/fields/<fieldId>`
Get records	`GET /v1/records/<appId>`
Get one record	`GET /v1/records/<appId>/<recordId>`
Add a record	`POST /v1/records/<appId>`
Update a record	`PUT /v1/records/<appId>/<recordId>`
Delete a record	`DELETE /v1/records/<appId>/<recordId>`
Add an attachment or image file to a record	`POST /v1/files/<appId>/<recordId>/<fieldId>?fileName=<filename>`
Get an attachment or image file from a record	`GET /v1/files/<appId>/<recordId>/<fieldId>?fileId=<fileId>`
Get the reports for an app/survey	`GET /v1/reports?appId=<appId>`
Get the data from one report	`GET /v1/reports/<reportId>`

Handling Errors

Any request to the API may fail for various reasons related to API key permissions or the format of the request or its data. In these cases, the response will contain the following:

- One of these HTTP Status Codes:
  - 400 (Bad Request)
  - 401 (Unauthorized)
  - 403 (Forbidden)
  - 404 (Not Found)
- A body in this format:

```
{
    "Message":"<message>"
}
```

  Where <message> may be either a simple string value or a serialized JSON value in this form:

```
{\"Errors\":[\"<message>\",\"<message>\"]}
```

Verify Connectivity

This request is used to verify that an external program can successfully connect to the API.

URL:

```
GET /v1/ping
```

Response Data: The response contains no body. A successful call returns HTTP Status 204 (No Content).

Get the List of Apps and Surveys

This request returns a list of all apps and surveys accessible to the API key.

URL:

```
GET /v1/apps
```

Response Data: The response is comprised of an array of app objects, with each app object containing the following information:

- Id – the integer id of the app/survey
- Name – the name of the app/survey

Example Response:

```
[
    {
        "Id": 1,
        "Name": "Roles"
    },
    {
        "Id": 2,
        "Name": "Users"
    }
]
```

Get the Fields for an App/Survey

This request returns a list of fields accessible to the API key for the selected app/survey.

URL Template:

```
GET /v1/fields/appId=<appId>
```

Response Data: The response is comprised of an array of field objects. Certain information is returned for each field, regardless of type. Some field types also include additional information.

All Field Types:

- Id – the integer id of the field
- AppId – the integer id of the app/survey
- Name – the name of the field
- Type –
    - 100 for Text fields
    - 200 for Number fields
    - 204 for AutoNumber fields (Record Id)
    - 300 for Date fields
    - 307 for Time Span fields
    - 400 for List fields
    - 500 for Reference and Parallel Reference fields
    - 600 for Survey Group Scoring fields
    - 601 for Survey Campaign fields
    - 800 for Attachment fields
    - 801 for Image fields
    - 900 for Formula fields
- Status – 0 for Enabled, 1 for Disabled
- IsRequired – true or false
- IsUnique – true or false

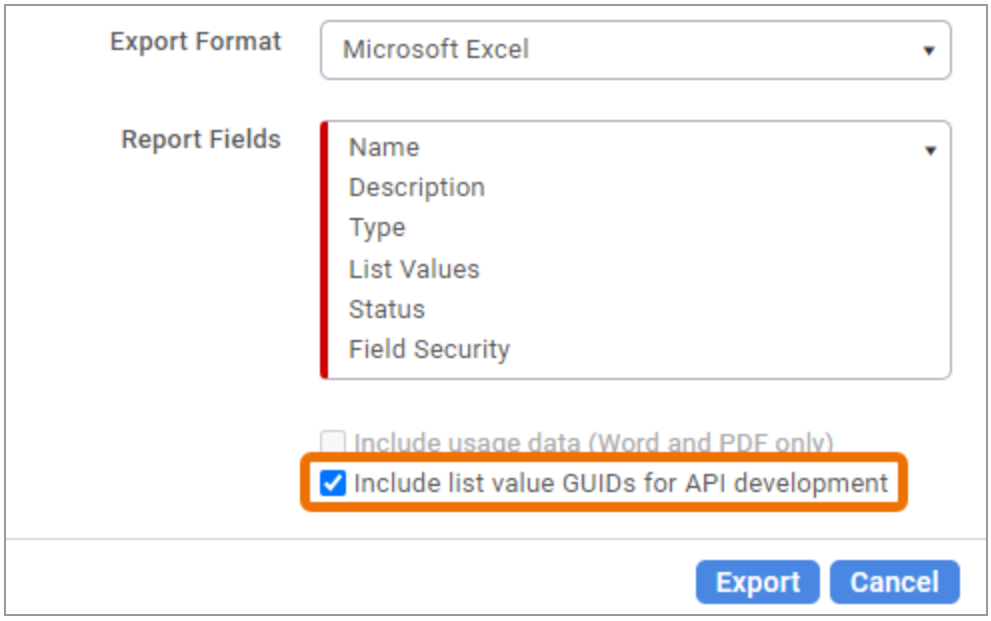
Reference/Parallel Reference Fields – Additional Information:

- Multiplicity – 0 for Single Select, 1 for Multi-Select

List Fields – Additional Information:

- Multiplicity – 0 for Single Select, 1 for Multi-Select
- Values – an array of list value objects, where each object contains:
  - Id – a GUID that identifies the list value *
  - Name – the text representation of the list value
  - SortOrder – an integer used to sort the list values
  - NumericValue – the numeric value configured for the list value
  - Color – the color configured for the list value (in RGB format: "#rrggbb")

***NOTE:** GUIDs for list values may also be obtained by exporting the field report for an app in Onspring.



Formula Fields – Additional Information:

- OutputType – 0 for Text, 1 for Numeric, 2 for Date, 3 for List
- Values – an array of list value objects, where each object contains the information described above for List fields. If OutputType is not 3 (List), the array will be empty.

Example Request:

```
GET /v1/fields?appId=2
```

Example Response:


```
[
    {
        "Id": 31,
        "AppId": 2,
        "Name": "Updated Date",
        "Type": 300,
        "Status": 0,
        "IsRequired": false,
        "IsUnique": false
    },
    {
        "Multiplicity": 0,
        "Values": [
            {
                "Id": "61dd5482-5293-4960-9dac-da6dc71c26eb",
                "Name": "Missouri",
                "SortOrder": 0,
                "NumericValue": null,
                "Color": "#0b0b0b"
            },
            {
                "Id": "93471483-f022-45d2-8ba2-a57d8a3c051e",
                "Name": "Kansas",
                "SortOrder": 1,
                "NumericValue": null,
                "Color": "#1636ee"
            },
            {
                "Id": "fe997fe7-3f5f-4aae-9e8e-aa5960badac3",
                "Name": "Iowa",
                "SortOrder": 2,
                "NumericValue": 1,
                "Color": "#f5c817"
            },
            {
                "Id": "286cc321-18ca-4bad-8fb6-85f9a7758d46",
                "Name": "Nebraska",
                "SortOrder": 3,
                "NumericValue": null,
                "Color": "#f90505"
```

```
            }
        ],
        "Id": 48,
        "AppId": 2,
        "Name": "State",
        "Type": 400,
        "Status": 0,
        "IsRequired": false,
        "IsUnique": false
    },
    {
        "Multiplicity": 0,
        "Id": 161,
        "AppId": 2,
        "Name": "Created By",
        "Type": 500,
        "Status": 0,
        "IsRequired": false,
        "IsUnique": false
    },
    {
        "OutputType": 1,
        "Values": [],
        "Id": 724,
        "AppId": 2,
        "Name": "Average Score",
        "Type": 900,
        "Status": 0,
        "IsRequired": false,
        "IsUnique": false
    }
]
```

Get One Field

This request returns information for the selected field (if the field is accessible to the API key).

URL Template:

```
GET /v1/fields/<fieldId>
```

Response Data: The response is comprised of a field object containing the same information described for field objects in the Get the Fields for an App/Survey section (which may vary based on the field type).

Example Request:

```
GET /v1/fields/31
Example Response:
{
    "Id": 31,
    "AppId": 2,
    "Name": "Updated Date",
    "Type": 300,
    "Status": 0,
    "IsRequired": false,
    "IsUnique": false
}
```

Get Records

This request returns a list of record information for the selected app/survey. (Only records accessible to the API key will be included.)

Minimum URL Template:

```
GET /v1/records/<appId>
```

Optional Parameters: There are several optional parameters that may be provided via the query string, each of which controls the data that is returned:

- **fieldIds** – a comma-separated list of integers that controls which fields are included in the response for each record (if not provided, all accessible fields are included)
- **recordIds** – a comma-separated list of integers that controls which records are included in the response. This parameter is intersected with $filter. If it is not provided, all accessible records (within the constraints of the $filter) are returned.
- **$filter** – a string representing a set of rules used to control which records are included in the response. This parameter is intersected with recordIds. If it is not provided, all accessible

records (within the constraints of recordIds) are returned. The filter expression is similar to that used with OData, with this specific implementation for Onspring:

- Each rule uses this format: <fieldId> <operator> <value>
- Only these field type/operator combinations are supported:
  - Text fields (including Formula fields with Text output) can be used with:
    - eq – Equals
    - ne – Not Equals
  - AutoNumber (Record Id), Number, and Date fields (including Formula fields with Number or Date output) can be used with:
    - eq – Equals
    - ne – Not Equals
    - lt – Less Than
    - gt – Greater Than
  - Value delimiters:
    - String values must be delimited with single quotation marks
    - Number values require no delimiters
    - Date values must be delimited with single quotation marks and preceded by the word datetime
  - and, not, or and parenthesis may be used to create compound expressions.
  - Examples:
    - not (38 gt 10 or 36 eq 'Smith')
    - 37 gt datetime'2014-03-01T00:00:00.0000000'
  - dataFormat – certain field types return different formats of data depending on which of the two values below is provided. (For other field types, this value makes no difference. See Response Data below for more details.)
    - Raw – (the default if no value is provided)
      - AutoNumber fields (Record Id) return an integer value
      - Date fields return a date/time value
      - List fields return a Guid (if single-select) or an array of Guids (if multi-select) representing the selected list value ids
      - Number fields return a decimal value
      - Multi-line Text fields return a string value that includes any html tags
      - Time Span fields return an object with the following members:
        - Quantity – an integer representing the number of increments
        - Increment –
          - 2 for Seconds
          - 4 for Minutes
          - 8 for Hours
          - 16 for Days

        - 32 for Weeks
        - 64 for Months
        - 128 for Years
    - Recurrence –
        - 0 for None
        - 1 for EndByDate
        - 2 for EndAfterOccurrences
    - EndByDate – a date/time value used when Recurrence is 1
    - EndAfterOccurrences – an integer used when Recurrence is 2
- Formatted – (intended to be easier for people to read) For the field types described in Raw, returns a string value as follows:
    - AutoNumber fields (Record Id) apply the configured formatting
    - Date fields apply the configured formatting
    - List fields return the name (if single select) or an array of names (if multi-select) for each selected list value
    - Number fields apply the configured formatting
    - Multi-line Text fields return a string value from which html tags have been removed
    - Time Span fields return the string representation of the time span members (e.g., "Every 10 Day(s) End By 1/1/2014 6:00 AM")

Response Data: The response is comprised of an array of record objects, with the following information included for each record object:

- AppId – the integer id of the app/survey
- RecordId – the internal record id of the record (Note: this may not be the same as the value of the Record Id AutoNumber field)
- FieldData – an array of objects for each field that has a value, with each object containing the following members:
    - FieldId – integer that identifies the field
    - Value – the value of the field on the record as defined by the Type member
    - Type – indicates the type of data contained in the Value member
        - 0 - string (including those resulting from dataFormat=Formatted)
        - 1 - integer value (used for AutoNumber, Survey Campaign, and single select Reference fields)
        - 2 - decimal value
        - 3 - date value
        - 4 - time span object (as described for dataFormat=Raw)
        - 5 - Guid value (used for single select List fields)
        - 10 - an array of strings (used for multi-select List fields)

- 11 - an array of integers (used for Image fields and multi-select Reference fields)
- 15 - an array of Guids (used for multi-select List fields)
- 16 - an array of attachment objects as follows:
  - FileId – integer that identifies the file
  - FileName – the name of the file
  - Notes – notes entered for the attachment
- 17 - an array of scoring group objects as follows:
  - ListValueId – Guid of the list value used for grouping
  - Name – name of the list value used for grouping
  - Score – decimal representing the score for the group
  - MaximumScore – decimal representing the maximum possible score for the group

Example Request:

```
GET /v1/records/2?recordIds=1,3&fieldIds=31,48&dataFormat=Raw
```

Example Response:

```
[
    {
        "AppId": 2,
        "RecordId": 1,
        "FieldData": [
            {
                "Type": 3,
                "FieldId": 31,
                "Value": "2016-02-09T16:23:26.482Z"
            },
            {
                "Type": 5,
                "FieldId": 48,
                "Value": "93471483-f022-45d2-8ba2-a57d8a3c051e"
            }
        ]
    },
    {
        "AppId": 2,
        "RecordId": 3,
        "FieldData": [
            {
                "Type": 3,
                "FieldId": 31,
```

```
                "Value": "2016-02-04T08:32:40.127Z"
            }
        ]
    }
]
```

Get One Record

This request returns data from the selected record in the selected app (if the record is accessible to the API key).

Minimum URL Template:

```
GET /v1/records/<appId>/<recordId>
```

Optional Parameters: There are two optional parameters that may be provided via the query string, each of which controls the data that is returned:

- fieldIds – a comma-separated list of integers that controls which fields are included in the response for the record (if not provided, all accessible fields are included)
- dataFormat – certain field types return different formats of data depending on which of the two values below is provided. See the description under the Get Records section for more specifics.
    - Raw – (the default if no value is provided)
    - Formatted

Response Data: The response is comprised of a record object containing the same information described for record objects in the Get Records section.

Example Request:

```
GET /v1/records/2/1?fieldIds=31,48&dataFormat=Formatted
```

Example Response:

```
{
    "AppId": 2,
    "RecordId": 1,
    "FieldData": [
        {
            "Type": 0,
            "FieldId": 31,
            "Value": "2/9/2016 4:23 PM"
        },
        {
            "Type": 0,
```

```
            "FieldId": 48,
            "Value": "Kansas"
        }
    ]
}
```

Add a Record

This request adds a record to the selected app/survey (if the API key has permission to create records in the app/survey).

URL Template:

```
POST /v1/records/<appId>
```

Additional Request Headers:

```
Content-Type: application/json;charset=utf-8
```

Request Body: The request body must be comprised of a JSON object with a FieldData object within it. The name of each member within FieldData should be a fieldId (delimited with double quotes).

The format should be as follows:

```
{
    "FieldData": {
        "<fieldId>": <value>,
           …
        "<fieldId>": <value>
    }
}
```

NOTE: Onspring stores date/time information in UTC (Coordinated Universal Time), and the API expects to pass UTC data back and forth. The Onspring GUI (graphical user interface) converts the UTC value to the appropriate display value depending on your instance configuration and user profile settings. If your source data is stored in a local time zone, you should convert it to UTC before passing the data to Onspring.

Response Data: If the request is successful, the response will contain the following (where <recordId> will be replaced with an integer value indicating the internal record id of the new record):

- HTTP Status 201 (Created)
- Additional Headers:
    - Location: /v1/records/<appId>/<recordId>

    - OnspringCreatedId: <recordId>
- If there are no warnings, a body in this format:

```
{
    "recordId":<recordId>
}
```

- If there are warnings, a body in this format:

```
{
    "recordId":<recordId>,
    "Warnings":[
        "<message>",
        …
        "<message>"
    ]
}
```

Example Request:

```
POST /v1/records/2
{
    "FieldData": {
        "33": "API add test",
        "46": "123"
    }
}
```

Example Response Additional Headers:

```
Location: /v1/records/2/417
OnspringCreatedId: 417
```

Example Response Body:

```
{
    "recordId":417
}
```

Update a Record

This request updates an existing record (if the API Key has permission to update the record).

URL Template:

```
PUT /v1/records/<appId>/<recordId>
```

Additional Request Headers:

```
Content-Type: application/json;charset=utf-8
```

Request Body: The request body must be comprised of a JSON object with a FieldData object within it. The name of each member within FieldData should be a fieldId (delimited with double quotes).

The format should be as follows:

```
{
    "FieldData": {
        "<fieldId>": <value>,
           …
        "<fieldId>": <value>
    }
}
```

NOTE: Onspring stores date/time information in UTC (Coordinated Universal Time), and the API expects to pass UTC data back and forth. The Onspring GUI (graphical user interface) converts the UTC value to the appropriate display value depending on your instance configuration and user profile settings. If your source data is stored in a local time zone, you should convert it to UTC before passing the data to Onspring.

Response Data: If the request is successful and has no warnings, the response will contain the following:

- HTTP Status 204 (No Content)
- No body

If the request is successful and contains warnings, the response will contain the following:

- HTTP Status 200 (OK)
- A body in this format:

```
{
    "Warnings":[
        "<message>",
        …
        "<message>"
    ]
}
```

Example Request:

```
PUT /v1/records/2/417
{
    "FieldData": {
```

```
            "33": "API update test",
            "46": "456"
        }
    }
```

Delete a Record

This request deletes an existing record (if the API key has permission to delete the record).

URL Template:

```
DELETE /v1/records/<appId>/<recordId>
```

Response Data: The response contains no body. A successful call returns HTTP Status 204 (No Content).

Example Request:

```
DELETE /v1/records/2/417
```

Add an Attachment or Image File to a Record

This request adds an attachment or image file to the selected field in the selected record (if the API Key has permission to update that field in the app/survey).

Minimum URL Template:
POST /v1/files/<appId>/<recordId>/<fieldId>?fileName=<fileName>

Optional Parameters: There are two optional parameters that may be provided via the query string:

- modifiedTime – the modified time (in UTC) to be stored for the file. The value should be provided in this format: `yyyy-mm-dd hh:mm:ssZ`

- fileNotes – the notes to be stored for an attachment. This parameter is ignored for image fields.

Additional Request Headers:

```
Content-Type: <the content type for the file>
Content-Length: <the number of bytes in the file>
```

Request Body: The request body must contain only the bytes representing the file contents.

Response Data: If the request is successful, the response will contain the following (where <fileId> will be replaced with an integer value indicating the internal file id of the new file):\

- HTTP Status 201 (Created)
- Additional Headers:
  - Location: /v1/files/<appId>/<recordId>/<fieldId>?fileId=<fileId>
  - OnspringCreatedId: <fileId>
- A body in this format:

```
{
    "fileId":<fileId>
}
```

Example Request:

```
POST /v1/files/2/417/50?fileName=Contract.pdf
&modifiedTime=2016-04-21%2009:19:45Z&fileNotes=Initial%20revision
<file content bytes>
```

Example Response Additional Headers:

```
Location: /v1/files/2/417/50?fileId=1234
OnspringCreatedId: 1234
```

Example Response Body:

```
{
    "fileId":1234
}
```

Get an Attachment or Image File from a Record

This request gets the contents of an attachment or image file from the selected field in the selected record (if the API key has permission to read that field in the app/survey).

URL Template:

```
GET /v1/files/<appId>/<recordId>/<fieldId>?fileId=<fileId>
```

Response Data: If the request is successful, the response will contain the following:

- HTTP Status 200 (OK)
- Additional Headers:
  - Content-Length: <the number of bytes in the file>
  - Content-Type: <the content type for the file>
  - Content-Disposition: attachment; filename=<fileName>
- A body containing the bytes representing the file contents

Example Request:

```
GET /v1/files/2/417/50?fileId=1234
```

Example Response Additional Headers:

```
Content-Length: 2858183
Content-Type: application/pdf
Content-Disposition: attachment; filename=Contract.pdf
```

Example Response Body:

```
<file content bytes>
```

Get the Reports for an App/Survey

This request returns a list of reports accessible to the API Key for the selected app/survey.

URL Template:

```
GET /v1/reports/appId=<appId>
```

Response Data: The response is comprised of an array of report objects, with each report object containing the following information:

- Id – the integer id of the report
- Name – the name of the report
- AppId – the integer id of the app/surve

Example Request:

```
GET /v1/reports?appId=2
```

Example Response:

```
[
    {
        "Id": 411,
        "Name": "Calendar",
        "AppId": 2
    },
    {
        "Id": 38,
        "Name": "Record Count",
        "AppId": 2
    }
]
```

Get the Data from One Report

This request returns data from the selected report, if the report is accessible to the API Key.

Minimum URL Template:

```
GET /v1/reports/<reportId>
```

Optional Parameters: There are two optional parameters that may be provided via the query string, each of which controls the data that is returned:

- dataType – certain reports can produce two types of data depending on how they are configured. The type of data returned is determined by which of the following two values is specified.
    - ReportData – (the default if no value is provided). This option returns one row for each top-level record that matches the report filter.
    - ChartData – This option is only valid for reports that are configured to produce a chart or chart data. It returns one row for each summary aggregation row that would be displayed if the report were configured to display chart data.
- dataFormat – Field and question types return different formats of data depending on which of the two values below is provided.
    - Raw – (the default if no value is provided)
        - Attachment fields and questions return a list of file names separated by "|" (pipe)
        - AutoNumber fields (Record Id) return an integer value
        - Date fields and questions return a string in the format used for editing the value in Onspring (e.g., m/d/yyyy h:mm tt)
        - Image fields return a list of file names separated by "|" (pipe)
        - List fields and questions return a list of list value names separated by "|" (pipe)
        - Matrix questions return a list of strings separated by "|" (pipe), where each string is in the format: `<row list value name> / <column list value name>`
        - Number fields and questions return a decimal value
        - Reference fields return a list of record link field strings separated by "|" (pipe)
        - Survey Campaign fields return the campaign name
        - Survey Group Scoring fields return a list of strings separated by "|" (pipe), where each string is in the format: <name>: <score>
        - Multi-line Text fields and questions return a string value that includes any html tags
        - Single line Text fields and questions return their text value
        - Time Span fields return the string representation of the time span value (e.g., "Every 10 Day(s) End By 1/1/2014 6:00 AM")
        - Chart Group Data and Series Data return the string value that would be displayed in the chart in Onspring
        - Chart Summary Data returns a decimal value

- Formatted (intended to be easier for people to read)
  - Attachment fields and questions return a list of file names separated by "\r\n" (CRLF)
  - AutoNumber fields (Record Id) apply the configured formatting
  - Date fields and questions apply the configured formatting
  - Image fields return a list of file names separated by "\r\n" (CRLF)
  - List fields and questions return a list of list value names separated by "\r\n" (CRLF)
  - Matrix questions return a list of strings separated by "\r\n" (CRLF), where each string is in the format: `<row list value name> / <column list value name>`
  - Number fields and questions apply the configured formatting
  - Reference fields return a list of record link field strings separated by "\r\n" (CRLF)
  - Survey Campaign fields return the campaign name
  - Survey Group Scoring fields return a list of strings separated by "\r\n" (CRLF), where each group string is in the format: `<name>: <score>`
  - Multi-line Text fields and questions return a string value from which html tags have been removed
  - Single line Text fields and questions return their text value
  - Time Span fields return the string representation of the time span value (e.g., "Every 10 Day(s) End By 1/1/2014 6:00 AM")
  - Chart Group Data and Series Data return the string value that would be displayed in the chart in Onspring
  - Chart Summary Data applies the configured formatting

Response Data: The response is comprised of an object that contains a Columns member that is an array of column names and a Rows member that is an array of row value arrays. The number of values in each row value array will match the number of column names.

The format is as follows:

```
{
    "Columns": [
        "<columnName>",
…
        "<columnName>"
    ],
    "Rows": [
        [
            <value>,
…
            <value>
        ],
```

```
…
        [
            <value>,
…
            <value>
        ]
    ]
}
```

Example Request:

```
GET /v1/reports/38?dataType=ReportData&dataFormat=Raw
```

Example Response:

```
{
    "Columns": [
        "AutoNumber Field",
        "Attachment Field",
        “Number Field",
        "Reference Field",
        "Date Field",
        "Image Field",
        "Multi-line Text Field",
        "Time Span Field"
    ],
    "Rows": [
        [
            1,
            "Estimate 1.xlsx|Proposal 1.docx",
            987654,
            "bsmith|mjones",
            "9/3/2014 5:01 PM",
            "Front View.png|Back View.jpg",
            "<p><strong>Notes 1</strong></p>",
            "Every 10 Day(s) End By 1/1/2014 6:00 AM"
        ],
        [
            2,
            null,
            null,
            null,
            null,
```

```
                null,
                null,
                null
            ]
        ]
    }
```