

# SurveyResponse Return Per Question Type v5

Last Modified on 03/19/2019 12:41 pm EDT |

This document includes specifics of the SurveyResponse return for all fields in SurveyGizmo that collect data, including all question types, other textboxes, other row headers and comment fields. IDs and reporting values are essential for identifying survey response data; the [Survey Legend](#) will be your friend in this process.

## Jump to a Question Type

- [Text Fields and Single-Select Questions](#)
- [Radio Button Grid](#)
- [Checkboxes and Image Select \(Multi\)](#)
- [Checkbox Grid](#)
- [Star Rating Grid](#)
- [File Upload](#)
- [Ranking \(Drag & Drop and Grid\)](#)
- [Image Heatmap](#)
- [Text Highlighter](#)
- [Signature](#)
- [Dropdown Menu List](#)
- [Textbox List](#)
- [Continuous Sum](#)
- [Piped Question](#)
  
- [Slider List](#)
- [Video / Audio Sentiment](#)
- [Cascading Dropdown Menu](#)
- [Grouping \(Open Card Sort\)](#)
- [Grouping \(Closed Card Sort\) / Quick Sort](#)

- [Dropdown Menu Grid](#)
- [Textbox Grid](#)
- [Max Diff](#)
- [Semantic Differential](#)
- [Contact Form and Custom Group](#)
- [Custom Table](#)
- [Other Specify Options](#)
- [Other Row Headers](#)
- [Comments](#)

## Text Fields and Single-Select Questions

Text fields and single-select questions have the same return, which includes the question ID and the response. The response will be the reporting value, in the case of single-select questions, or the text that was entered by the respondent in the case of text fields.

Text fields and single-select questions include: [Radio Buttons](#), [Dropdown Menu](#), [Textbox](#), [Essay](#), [Email](#), [Date](#), [Slider](#), [Net Promoter Score®](#), [Rating](#), [Number](#), [Percent](#), [Image Select \(select one\)](#). In addition, all of these question types, when part of a [Custom Group](#) or [Contact Form](#), return in the same format; [learn more](#).

json example:

```
"5": {
  "id": 5,
  "type": "RADIO",
  "question": "Radio Button Question Title",
  "section_id": 1,
  "answer": "Answer Option 1",
  "answer_id": 10001,
  "shown": true
}
```

## Radio Button Grid

The [Radio Button Grid](#) is a series of Radio Button questions where each row is a

question. Each row has its own question ID and the response will be the reporting value of the grid column.

json example:

```
"7": {
  "id": 7,
  "type": "RADIO",
  "parent": 6,
  "question": "Row 1",
  "section_id": 1,
  "answer": "Column 1",
  "shown": true
},
"8": {
  "id": 8,
  "type": "RADIO",
  "parent": 6,
  "question": "Row 2",
  "section_id": 1,
  "answer": "Column 2",
  "shown": true
}
```

### Checkboxes and Image Select (multi select)

Multi-select questions, which includes [Checkboxes](#) and [Image Select \(multi select\)](#), will return multiple fields. Think of each answer option as a separate question where the answer can be selected or unselected. The return includes the question ID, the option SKU and the response. The response will be the reporting value.

If the option was not selected the [answer] field will not be shown for that option.

json example:

```
"10": {
  "id": 10,
  "type": "parent",
  "question": "Checkboxes Question Title",
  "section_id": 1,
  "options": {
    "10007": {
      "id": 10007,
      "option": "Answer Option 1",
      "answer": "Answer Option 1"
    }
  },
  "shown": true
}
```

## Checkbox Grid

The **Checkbox Grid** is a series of Checkbox questions where each row is a question. Because checkbox questions are multi-select questions each answer option is treated as a separate question where the answer can be selected or unselected. For this reason, the Checkbox Grid returns rows x columns elements that can be identified by the question ID (row) and the option SKU (column) combination. The response will be the reporting value.

For example, a Checkbox Grid with 2 rows and 2 columns will output 4 fields.

If the option was not selected the response will be empty.

json example:

```

"30": {
  "id": 30,
  "type": "parent",
  "question": "Checkbox Grid Question Title",
  "subquestions": {
    "31": {
      "10062": {
        "id": 10062,
        "type": "CHECKBOX",
        "parent": 30,
        "question": "Row 1 : Column 1",
        "answer": "Column 1",
        "shown": true
      },
      "10063": {
        "id": 10063,
        "type": "CHECKBOX",
        "parent": 30,
        "question": "Row 1 : Column 2",
        "answer": null,
        "shown": true
      }
    },
    "32": {
      "10062": {
        "id": 10062,
        "type": "CHECKBOX",
        "parent": 30,
        "question": "Row 2 : Column 1",
        "answer": null,
        "shown": true
      },
      "10063": {
        "id": 10063,
        "type": "CHECKBOX",
        "parent": 30,
        "question": "Row 2 : Column 2",
        "answer": "Column 2",
        "shown": true
      }
    }
  },
  "section_id": 3,
  "shown": true
}

```

## Star Rating Grid

The [Star Rating Grid](#) returns rows x columns elements that can be identified by the question ID (row) and the option SKU (column) combination. The response will be the count of stars selected.

json example:

```
"16": {
  "id": 16,
  "type": "parent",
  "question": "Star Rating Grid Question Title",
  "subquestions": {
    "17": {
      "10019": {
        "id": 10019,
        "type": "MULTI_TEXTBOX",
        "parent": 16,
        "question": "Row 1 : Column 1",
        "answer": "2",
        "shown": true
      },
      "10020": {
        "id": 10020,
        "type": "MULTI_TEXTBOX",
        "parent": 16,
        "question": "Row 1 : Column 2",
        "answer": "3",
        "shown": true
      }
    },
    "18": {
      "10019": {
        "id": 10019,
        "type": "MULTI_TEXTBOX",
        "parent": 16,
        "question": "Row 2 : Column 1",
        "answer": "4",
        "shown": true
      },
      "10020": {
        "id": 10020,
        "type": "MULTI_TEXTBOX",
        "parent": 16,
        "question": "Row 2 : Column 2",
        "answer": "1",
        "shown": true
      }
    }
  },
  "section_id": 3,
  "shown": true
}
```

## File Upload

The [File Upload](#) question returns an array per file.

json example:

```
"20": {
  "id": 20,
  "type": "FILE",
  "question": "File Upload Question Title",
  "section_id": 3,
  "answer": "133-5960e881127b5f4fff11e4c8b481229d_file.jpeg",
  "answer_id": 0,
  "shown": true
}
```

You can construct the link to the file with this information. All file upload links are a specific file page with this base file path:

```
http://surveygizmoreponseuploads.s3.amazonaws.com/fileuploads/
```

In addition to the base, you'll need to add the following:

- Your account ID
- Survey ID
- Name of the file

The complete link contains these elements:

```
/[account id]/[survey id ]/[filename]
```

## Drag & Drop Ranking and Ranking Grid

Ranking questions, both the [Drag & Drop Ranking](#) and the [Ranking Grid](#), will return an element per row with the question ID, the option SKU, and the response. The response is the numeric rating selected for that option.

json example:

```
"21": {
  "id": 21,
  "type": "RANK",
  "question": "Drag & Drop Ranking Question Title",
  "section_id": 3,
  "answer": {
    "10036": {
      "id": 10036,
      "option": "Answer Option 1",
      "rank": "2"
    },
    "10037": {
      "id": 10037,
      "option": "Answer Option 2",
      "rank": "1"
    }
  },
  "shown": true
}
```

## Image Heatmap

The [Image Heatmap](#) returns an array of values that include grid coordinates for each click as well as the answer option associated with the click.

json example:



```

"28": {
  "id": 28,
  "type": "HEATMAP",
  "question": "Image Heatmap Question Title",
  "section_id": 3,
  "answer": {
    "response": {
      "dots": [
        {
          "x": 309,
          "y": 514,
          "xPercent": "32.56",
          "yPercent": "93.12",
          "gridX": 32,
          "gridY": 93,
          "answer": "10058",
          "color": "#94c826",
          "comment": "",
          "id": "a0141af6-858b-59cd-e964-c8c5c2cc2352"
        },
        {
          "x": 635,
          "y": 515,
          "xPercent": "66.91",
          "yPercent": "93.30",
          "gridX": 66,
          "gridY": 93,
          "answer": "10059",
          "color": "#f06586",
          "comment": "",
          "id": "c8fc2d54-6381-5c1c-7d3b-c1a7d49ad660"
        }
      ],
      "imageW": 949,
      "imageH": 552
    }
  },
  "answer_id": 0,
  "shown": true
}

```

## Text Highlighter

The [Text Highlighter](#) element returns each piece of highlighted text as well as the associated answer.

json example:

```

"29": {
  "id": 29,
  "type": "parent",
  "question": "Text Highlighter Question Title",
  "section_id": 3,
  "options": {
    "10060": {
      "id": 10060,
      "option": "Like",
      "answer": "\"Survey platform\": \"CREATE DATA DRIVEN TEAM\""
    },
    "10061": {
      "id": 10061,
      "option": "Dislike",
      "answer": "\"collection & reporting.\": \""
    }
  },
  "shown": true
}

```

## Signature

The typed text from the [Signature](#) question is available via the API. The response will be the text that was entered by the respondent.

Sign here:

Clear

John Hancock

Sign name using mouse or touch pad

Signature of

json example:

```

"33": {
  "id": 33,
  "type": "SIGNATURE",
  "question": "Signature Question Title",
  "section_id": 3,
  "answer": {
    "signature": "base-64 encoded image",
    "name": "John Hancock"
  },
  "answer_id": 0,
  "shown": true
}

```

## Dropdown Menu List

The [Dropdown Menu List](#) is a series of Dropdown Menu questions where each row is a question. Each row has its own question ID and the response will be the reporting value of the option selected by the respondent.

json example:

```

"35": {
  "id": 35,
  "type": "MENU",
  "parent": 34,
  "question": "Row 1",
  "section_id": 3,
  "answer": "Answer Option 1",
  "shown": true
},
"36": {
  "id": 36,
  "type": "MENU",
  "parent": 34,
  "question": "Row 2",
  "section_id": 3,
  "answer": "Answer Option 2",
  "shown": true
}

```

## Textbox List

[Textbox List](#) questions will return an element per row with the question ID, the option SKU, and the response. The response will be the text that was entered by the respondent.

json example:

```
"37": {
  "id": 37,
  "type": "parent",
  "question": "Textbox List Question Title",
  "section_id": 3,
  "options": {
    "10068": {
      "id": 10068,
      "option": "Row 1",
      "answer": "text list answer"
    }
  },
  "shown": true
}
```

## Continuous Sum

The [Continuous Sum](#) will return an element per row with the question ID, the option SKU, and the response. The response will be the value that was entered by the respondent.

json example:

```
"38": {
  "id": 38,
  "type": "parent",
  "question": "Continuous Sum Question Title",
  "section_id": 3,
  "options": {
    "10070": {
      "id": 10070,
      "option": "Row 1",
      "answer": "6"
    },
    "10071": {
      "id": 10071,
      "option": "Row 2",
      "answer": "7"
    }
  },
  "shown": true
}
```

## Slider List

The [Slider List](#) is a series of Slider questions where each row is a question. The

return will include the question ID, the option SKU for the row, and the response, which is the value selected by the respondent.

json example:

```
"40": {
  "id": 40,
  "type": "parent",
  "question": "Slider List Question Title",
  "section_id": 4,
  "options": {
    "10074": {
      "id": 10074,
      "option": "Row 1",
      "answer": "30"
    },
    "10075": {
      "id": 10075,
      "option": "Row 2",
      "answer": "63"
    }
  },
  "shown": true
}
```

## Video / Audio Sentiment

The [Video Sentiment](#) question will return an element that displays the time-stamped feedback for the video content.

json example:

```

"42": {
  "id": 42,
  "type": "parent",
  "question": "Video Sentiment Question Title",
  "section_id": 4,
  "options": {
    "10078": {
      "id": 10078,
      "option": "Perception",
      "answer": "{\"duration\":365,\"data\":{\"0\":0,\"1\":0,\"2\":-6,\"3\":-2,\"4\":-2,\"149\":-1,\"194\":-1,\"195\":1,\"196\":-5,\"269\":10,\"309\":10,\"356\":10,\"357\":10,\"358\":10,\"359\":10,\"360\":10,\"361\":10,\"362\":10,\"363\":10,\"364\":10,\"365\":10,\"366\":10,\"367\":10}}}"
    }
  },
  "shown": true
}

```

## Cascading Dropdown Menu

The [Cascading Dropdown Menu](#) will return an element per menu with the question ID, the option SKU, and the response. The response will be the option that was selected by the respondent.

json example:

```
"45": {
  "id": 45,
  "type": "parent",
  "question": "Cascading Dropdown Question Title",
  "section_id": 4,
  "options": {
    "10086": {
      "id": 10086,
      "option": "Make",
      "answer": "Ford"
    },
    "10087": {
      "id": 10087,
      "option": "Model",
      "answer": "Focus"
    },
    "10088": {
      "id": 10088,
      "option": "Year",
      "answer": "2003"
    }
  },
  "shown": true
}
```

## Grouping (Open Card Sort)

The [Grouping \(Open Card Sort\)](#) returns an element that shows the card sort items (along with id's) and the buckets they were placed in (the answer field refers to the bucket label created by the respondent)

json example:

```
"46": {
  "id": 46,
  "type": "parent",
  "question": "Grouping (Open Card Sort) Question Title",
  "section_id": 4,
  "options": {
    "10089": {
      "id": 10089,
      "option": "Item 1",
      "answer": "Category 1"
    },
    "10090": {
      "id": 10090,
      "option": "Item 2",
      "answer": "Category 2"
    }
  },
  "shown": true
}
```

## Grouping (Closed Card Sort) / Quick Sort

The [Grouping \(Closed Card Sort\)](#) and [Quick Sort](#) questions return an element much like a radio button grid question. The answer field will refer to the bucket that the item was placed into.

json example:



```
"48": {
  "id": 48,
  "type": "RADIO",
  "parent": 47,
  "question": "Item 1",
  "section_id": 4,
  "answer": "Group 1",
  "shown": true
},
"49": {
  "id": 49,
  "type": "RADIO",
  "parent": 47,
  "question": "Item 2",
  "section_id": 4,
  "answer": "Group 2",
  "shown": true
}
```

## Dropdown Menu Grid

The [Dropdown Menu Grid](#) returns rows x columns elements that can be identified by the question ID (row) and the option SKU (column) combination. The response will be the option the respondent selected.

json example:

```

"53": {
  "id": 53,
  "type": "parent",
  "question": "Dropdown Menu Grid",
  "subquestions": {
    "54": {
      "10095": {
        "id": 10095,
        "type": "MULTI_TEXTBOX",
        "parent": 53,
        "question": "Row 1 : Column 1",
        "answer": "Answer Option 1",
        "shown": true
      },
      "10096": {
        "id": 10096,
        "type": "MULTI_TEXTBOX",
        "parent": 53,
        "question": "Row 1 : Column 2",
        "answer": "Answer Option 2",
        "shown": true
      }
    },
    "55": {
      "10095": {
        "id": 10095,
        "type": "MULTI_TEXTBOX",
        "parent": 53,
        "question": "Row 2 : Column 1",
        "answer": "Answer Option 3",
        "shown": true
      },
      "10096": {
        "id": 10096,
        "type": "MULTI_TEXTBOX",
        "parent": 53,
        "question": "Row 2 : Column 2",
        "answer": "Answer Option 1",
        "shown": true
      }
    }
  },
  "section_id": 4,
  "shown": true
}

```

## Textbox Grid

The **Textbox Grid** returns rows x columns elements that can be identified by the question ID (row) and the option SKU (column) combination. The response will be the text the respondent entered.

json example:

```
"56": {
  "id": 56,
  "type": "parent",
  "question": "Textbox Grid Question Title",
  "subquestions": {
    "57": {
      "10097": {
        "id": 10097,
        "type": "MULTI_TEXTBOX",
        "parent": 56,
        "question": "Row 1 : Column 1",
        "answer": "text grid row 1/column1",
        "shown": true
      },
      "10098": {
        "id": 10098,
        "type": "MULTI_TEXTBOX",
        "parent": 56,
        "question": "Row 1 : Column 2",
        "answer": "text grid row 2/column 2",
        "shown": true
      }
    },
    "58": {
      "10097": {
        "id": 10097,
        "type": "MULTI_TEXTBOX",
        "parent": 56,
        "question": "Row 2 : Column 1",
        "answer": "text grid row 2/column 1",
        "shown": true
      },
      "10098": {
        "id": 10098,
        "type": "MULTI_TEXTBOX",
        "parent": 56,
        "question": "Row 2 : Column 2",
        "answer": "text grid row 2/column 2",
        "shown": true
      }
    }
  },
  "section_id": 4,
  "shown": true
}
```

## Max Diff

In the SurveyResponse return for the [Max Diff](#) question type, the values assigned to each attribute based on how that individual respondent ranked the

values are exported. They range from 0 to 100. These values are *not* used in computing the aggregate bayesian average and rank order you see in the Summary Reports. Think of these as a snapshot of the rank order for that response alone. To learn more about this check out our [Max Diff Tutorial](#).

json example:

```
"59": {
  "id": 59,
  "type": "parent",
  "question": "Max Diff Question Title",
  "section_id": 6,
  "options": {
    "10099": {
      "id": 10099,
      "option": "Answer Option 1",
      "answer": 33.3333
    },
    "10100": {
      "id": 10100,
      "option": "Answer Option 2",
      "answer": 66.6667
    }
  },
  "shown": true
}
```

There is more detailed raw data that is reported in a `[[variable(ID)]]` field. This data is formatted as follows:

Shown Sku; Shown Sku; Shown Sku; Shown Sku: Best Selected Sku /Worst Selected Sku

Semantic Differential

The Semantic Differential element returns an array for each row with an indication of which option was selected.

json example:

```
"61": {
  "id": 61,
  "type": "RADIO",
  "parent": 60,
  "question": "Left|Right",
  "section_id": 4,
  "answer": "Column 1",
  "shown": true
},
"62": {
  "id": 62,
  "type": "RADIO",
  "parent": 60,
  "question": "Left|Right",
  "section_id": 4,
  "answer": "Column 2",
  "shown": true
}
```

## Contact Form and Custom Group

All [Contact Form](#) and [Custom Group](#) subquestions, the fields that actually collect data, have their own IDs and return as they would if they were not part of a grouping.

## Custom Table

[Custom Table](#) data is unique in many ways. First, unlike other grid questions, where the rows are the subquestions, in custom tables, the columns are the subquestions. Each column of a Custom Table has its own question ID. Custom Tables also make use of our [piping](#) functionality to create the fields in the cells of each row.

## Custom Table Radio Buttons and Dropdown Menus

Custom Table Radio Button and Dropdown Menus will return a field per row. Each field returned will include the subquestion ID (column), the row it was piped for, and the response. The response is the Radio Button or Dropdown Menu option selected.

json example:

```

"83": {
  "id": 83,
  "type": "parent",
  "question": "Custom Table Question Title",
  "subquestions": {
    "10001": {
      "id": 10001,
      "type": "RADIO",
      "question": "Radio Button Column",
      "section_id": 4,
      "answer": "Option 1",
      "answer_id": 10113,
      "shown": true
    },
    "10002": {
      "id": 10002,
      "type": "RADIO",
      "question": "Radio Button Column",
      "section_id": 4,
      "answer": "Option 2",
      "answer_id": 10114,
      "shown": true
    },
  },
  "section_id": 4,
  "shown": true
}

```

## Custom Table Checkboxes

Custom Table Checkbox questions return number of rows x number of option elements. Think of each answer option as a separate question where the answer can be selected or unselected. The return includes the question ID for the subquestion, the option SKU, the row header from the Custom Table, and the response. The response will be option selected.

json example:

```

"87": {
  "id": 83,
  "type": "parent",
  "question": "Custom Table Question Title",
  "subquestions": {
    "10005": {
      "id": 10005,
      "type": "parent",
      "question": "Checkboxes Column",
      "section_id": 4,
      "options": {
        "10119": {
          "id": 10119,
          "option": "Answer Option 1",
          "answer": "Answer Option 1"
        }
      },
      "shown": true
    },
    "10006": {
      "id": 10006,
      "type": "parent",
      "question": "Checkboxes Column",
      "section_id": 4,
      "options": {
        "10120": {
          "id": 10120,
          "option": "Answer Option 2",
          "answer": "Answer Option 2"
        }
      },
      "shown": true
    }
  },
  "section_id": 4,
  "shown": true
}

```

## Custom Table Essay and Textbox Fields

Custom Table Essay and Textbox fields will return a field per row +1. The first returned field will be empty. The second and later elements will include the question ID for the subquestion, the row header from the Custom Table, and the response. The response will be the text that was entered by the respondent.

json example:

```

"83": {
  "id": 83,
  "type": "parent",
  "question": "Custom Table Question Title",
  "subquestions": {
    "10003": {
      "id": 10003,
      "type": "ESSAY",
      "question": "Essay Column",
      "section_id": 4,
      "answer": "custom table essay answer",
      "shown": true
    },
    "10004": {
      "id": 10004,
      "type": "ESSAY",
      "question": "Essay Column",
      "section_id": 4,
      "answer": "custom table essay answer 2",
      "shown": true
    },
  },
  "section_id": 4,
  "shown": true
}

```

## Other Textboxes

**Other Textboxes** can be added to both Radio Button and Checkbox questions. The return will include two elements. The first element will be for the selected option. The return will include the question ID and the response, which is a reporting value. The second element will include the question ID, the option sku, and the response, which is the text the respondent entered in the text field.

json example:

```

"91": {
  "id": 91,
  "type": "RADIO",
  "question": "Radio Button Question With Other Textbox",
  "section_id": 1,
  "answer": "other text",
  "answer_id": "10123-other",
  "shown": true
}

```

## Other Row Header

**Other Row Headers** can be added to the following grid questions to collect



other-specify responses:

- Radio Button Grid
- Checkbox Grid
- Dropdown Menu Grid
- Textbox Grid
- Star Rating Grid

Other row headers will return the question ID, the text the survey respondent entered in the other row text field, and grid response.

json example:

```
"10001": {
  "id": 10001,
  "type": "RADIO",
  "parent": 92,
  "question": "Other Row Label",
  "section_id": 1,
  "answer": "Column 2",
  "shown": false
}
```

## Comments

[Comments](#) can be added to any question within the survey. They will appear at the bottom of the SurveyResponse return and will have an ID that corresponds to the question ID with which it is associated.

json example:

```
"5": {
  "id": 5,
  "type": "RADIO",
  "question": "Radio Button Question Title",
  "section_id": 1,
  "answer": "Answer Option 1",
  "answer_id": 10001,
  "shown": true,
  "comments": "Comment Input Text"
}
```

## Piped Question

When using [Question Piping/Repeating](#), a survey question is repeated for each answer selected in the source question.

Each repeated question will be identified by the source question answer option that it is based on. Using the below example, in the source question (Checkbox), "apple" has an ID of 10001. The repeated question (Radio Button) based on apple will be identified by "10001".

json example:

```
"2": {
  "id": 2,
  "type": "parent",
  "question": "Source Question Title",
  "section_id": 1,
  "options": {
    "10001": {
      "id": 10001,
      "option": "apple",
      "answer": "apple"
    },
    "10002": {
      "id": 10002,
      "option": "banana",
      "answer": "banana"
    },
    "10003": {
      "id": 10003,
      "option": "orange",
      "answer": "orange"
    }
  },
  "shown": true
},
"10001": {
  "id": 10001,
  "type": "RADIO",
  "question": "Repeated Question Title for apple",
  "section_id": 3,
  "answer": "1-3",
  "answer_id": 10004,
  "shown": true
},
"10002": {
  "id": 10002,
```

```
        "type": "RADIO",
        "question": "Repeated Question Title for banana",
        "section_id": 3,
        "answer": "4-6",
        "answer_id": 10005,
        "shown": true
    },
    "10003": {
        "id": 10003,
        "type": "RADIO",
        "question": "Repeated Question Title for orange",
        "section_id": 3,
        "answer": "6-10",
        "answer_id": 10006,
        "shown": true
    }
}
```