API walkthrough

Table of Contents

Walkthrough: Using APIs to create a search experience and send events

Abstract

Overview of the requests to send the Search and Recommendation API and the Events API to create a search experience with Sitecore Search.

After an administrator configures attributes, sources, and search features , you can integrate with Sitecore Search to create search experiences for your visitors.

To do this through an API, make calls to the Search and Recommendation API to request content and use the response to create the experience you want. At the same time, we recommend that you also send all applicable event data to Search through the Events API.

IMPORTANT

This walkthrough contains the API requests needed to create a specific search experience. While your implementation might require other features, use this walkthrough as a guideline on how to build a request on top of previous requests.

To create different experiences, study the Search and Recommendation API to find the object you need and then use that object to create a request based on the examples in this walkthrough. Similarly, study the objects in the Events API to send all applicable events to Search.

This walkthrough is based on a scenario where a visitor to your site lands on a search results page and sees facets. Then, the visitor enters a keyword in the search bar and sees suggestions in a preview search widget. When the visitor clicks a suggestion, they see the results page with new results, and they select a facet value to filter the results. They then sort the results in descending order of content titles and click a content item.

This walkthrough describes how to form API requests to:

- Create a search results page with facets
- Get recommendations to show in a preview search
- Create a search results page based on a keyword
- Create a search results page filtered by a facet value
- Create a search results page with sorting applied
- Redirect the visitor to the content page
- Send events to Search

NOTE

With the exception of the API request to show suggestions in a preview search widget, you see that each of the following requests builds on the previous request.

Create a search results page with facets

When a visitor first lands on your search page, you want to show them search results and facets.

To create a search results page with facets:

- add the context, content, and facet objects, and pass the ID for the search results page as the value of `rfk_id`.

Here's a sample CURL request to get data to show a search results page with facets:

```
curl --location 'https://discover.sitecorecloud.io/discover/v2/159871551' \
--header 'Authorization: 01-22977f73-b96bm327df2ffc7ip16d4c030a201240ced2487n' \
--data '{
    "context": {
        "user": {
            "UUID": "159828561-m7-17-44-1p-c0ya4nb69t8o0iho2eia-1667926756187"
        },
        "locale": {
            "country": "us",
            "language": "en"
        }
    },
    "widget": {
        "items": [
            {
                "entity": "content",
                "rfk_id": "rfkid_7",
                "search": {
                    "content": {},
                    "facet": {
                        "all": true,
                        "max": 100
                    }
                }
            }
        ]
    }
}'
```

Here's a sample response:

```
{
    "widgets": [
        {
            "rfk_id": "rfkid_7",
            "type": "content_grid",
            "entity": "content",
            "facet": [
                {
                    "name": "type",
                    "label": "Type",
                    "value": [
                        {
                            "id": "facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoidHlwZSIsInZhbHVlI
                            "text": "News",
                            "count": 688
```

```
                    },
                    {
                        "id": "facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoidHlwZSIsInZhbHVlI
                        "text": "Customers",
                        "count": 576
                    },
                    ...
                ]
            },
            {
                "name": "review_rating",
                "label": "Rating",
                "value": [
                    {
                        "id": "facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoidHlwZSIsInZhbHVlI
                        "text": "5",
                        "count": 310
                    },
                    {
                        "id": "facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoidHlwZSIsInZhbHVlI
                        "text": "4",
                        "count": 780
                    },
                    ...
                ]
            }
        ],
        "content": [
            {
                "description": null,
                "id": "https___www_sitecore_com_company_newsroom_news_2023_03_siteco
                "review_rating": "5",
                "image_url": "https://wwwsitecorecom.azureedge.net/-/media/sitecores
                "source_id": "730959",
                "title": "Sitecore named a Leader for DXP",
                "keywords": ["DXP","gartner"],
                "type": "News",
                "url": "https://www.sitecore.com/company/newsroom/news/2023/03/sitec
            },
            {
                "description": " active and real-time content integration framework
                "review_rating": "4",
                "image_url": "https://wwwsitecorecom.azureedge.net/-/media/sitecores
                "source_id": "730959",
                "title": "Connector & App Marketplace",
                "keywords": ["connector","content hub", "data synchronization"],
                "type": "Others",
                "url": "https://www.sitecore.com/products/marketplace"
            }
            ...
        ],
        "total_item": 3443,
        "limit": 10,
        "offset": 1
    }
```

```
    ],
    "dt": 45,
    "ts": 1680573661787
}
```

To use the response to create the search experience:

1. To show a list of content items on the page, see the `widgets.content` object in the response. For each content item, get the title, description, image, and URL from the *title*, *description*, *image_url*, and *url* fields.
2. To show facets, see the `widgets.facet` object in the response and then:
   - Get the name of the facet from the `facet.label` field. Here, you have two facets, *Type* and *Rating*.
   - Get the facet values for each facet from the `facet.value` object. Use the value of the `text` key as the name of the facet value to show on your page. Use the value of the `count` as the number of content items for that facet value.
     For example, for the *type* facet, you can use a facet value called *News* with *688* content items. Similarly, for the `review_rating` facet, you can use a facet value called *5* with *310* content items.

Get recommendations to show in a preview search

When the visitor enters a keyword in the search bar, you want to show suggestions.

To get recommendations to show in a preview search widget:

- add a query object, a suggestions object, and pass the ID of the preview search widget as the `rfk_id`.

In this example, the visitor enters the keyword *clou* and you want to show suggestions based on the *Keyword* attribute.

Here's a sample CURL request to get data to show recommendations based on the *keyword* attribute in a preview search widget:

```
curl --location 'https://discover.sitecorecloud.io/discover/v2/159871551' \
--header 'Authorization: 01-21667f73-b96bn327df2ffc7ab16d4c030a201240ced2490a' \
--data '{
    "context": {
        "user": {
            "UUID": "159828561-m7-17-44-1p-c0ya4nb69t8o0iho2eia-1667926756187"
        },
        "locale": {
            "country": "us",
            "language": "en"
        }
    },
    "widget": {
        "items": [
            {
                "entity": "content",
                "rfk_id": "rfkid_6",
                "search": {
                    "content": {},
                    "query": {
                        "keyphrase": "clou"
                    },
```

```
                    "suggestion": [
                        {
                            "name": "keyword_suggester"
                        }
                    ]
                }
            }
        ]
    }
}'
```

You see that:

- The value you pass for the `suggestion` object is the API name of the suggestion block and administrator configured in the Customer Engagement Console (CEC).
- The value you pass for `rfk_id` is the ID of the default preview search widget in the CEC.

Here's a sample response to this request:

```
{
    "widgets": [
        {
            "rfk_id": "rfkid_7",
            "type": "content_grid",
            "entity": "content",
            "suggestion": {
                "keyword_suggester": [
                    {
                        "text": "cloud faq",
                        "freq": 14
                    },
                    {
                        "text": "cloud addendum",
                        "freq": 3
                    },
                    {
                        "text": "cloud portal",
                        "freq": 3
                    }
                    ...
                ]
            },
            "total_item": 0,
            "limit": 10,
            "offset": 0
        }
    ],
    "dt": 13,
    "ts": 1680632333374
}
```

To use the response to create this search experience:

1. To show suggestions, see the `widgets.suggestion.name_suggestor` array in the response. This array has a list of suggestions, each with a `text` and a `freq`.
2. To display one suggestion in the preview search widget, the value of the `text`.

The value of the `freq` key tells you how often a phrase appears in that part of the content collection that is relevant to this request. For example, *cloud faq* appears 14 times in the pool of search results that are relevant to this query.

Create a search results page based on a keyword

When the visitor clicks a suggestion, they see a search result page that's modified base on the keyword.

To create a search results page based on a keyword:

- modify the previous request you sent when you created a search results page with facets [4] to add a query object.

In this example, the visitor clicks the suggestion *cloud portal*:

Here's a sample request to get data to show a search results page based on a keyword:

```
{
    "context": {
        "locale": {
            "country": "us",
            "language": "en"
        },
        "user": {
            "uuid": "43533744-1y-6i-4j-1p-nn0if7kfzq8kw3am4i17-1664982959269"
        }
    },
    "widget": {
        "items": [
            {
                "entity": "content",
                "rfk_id": "rfkid_7",
                "search": {
                    "content": {},
                    "facet": {
                        "all": true,
                        "max": 100
                    },
                    "query": {
                        "keyphrase": "cloud portal"
                    }
                }
            }
        ]
    }
}
```

Create a search results page filtered by a facet value

When the visitor selects a facet value to filter results, they see results filtered according to that facet value.

To create a search results page filtered by a facet value"

- modify the previous request you sent when you created a search results page based on a keyword [8] to add a filter object.

In this example, the visitor selects *Blogs* in the *Type* facet.

Here's a sample request to get data to show a search results page that is filtered by facet value *facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoicHJvZHVjdCIsInZhbHVlIjoiRGlzY292ZXIifQ*:

NOTE

facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoicHJvZHVjdCIsInZhbHVlIjoiRGlzY292ZXIifQ is the ID of the *Blogs* facet value. You can this value from the response to the previous request, *Create a search results page based on a keyword*, or from any request that asks for facets.


```
{
    "context": {
        "locale": {
            "country": "us",
            "language": "en"
        },
        "user": {
            "uuid": "43533744-1y-6i-4j-1p-nn0if7kfzq8kw3am4i17-1664982959269"
        }
    },
    "widget": {
        "items": [
            {
                "entity": "content",
                "rfk_id": "rfkid_7",
                "search": {
                    "content": {},
                    "suggestion": [
                        {
                            "name": "title_context_aware"
                        }
                    ],
                    "facet": {
                        "all": true,
                        "max": 100,
                        "types": [
                            {
                                "name": "type",
                                "filter": {
                                    "values": [
                                        "facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoicHJvZHVjdCI
                                    ],
                                    "type": "or"
                                }
                            }
                        ]
                    },
                    "query": {
                        "keyphrase": "cloud portal"
                    }
                }
            }
```

```
        ]
    }
}
```

Create a search results page with sorting applied

When the visitor sorts results by descending order of title, they see the search results in the sorted order.

To create a search results page with sorting applied:

- modify the previous request you sent to *Create a search results page filtered by a facet value* to add a sort object.

In this example, the visitor sorts by descending order of title.

Here's a sample request to get data to show a search results page that is sorted in descending order of content titles:

```
{
    "context": {
        "locale": {
            "country": "us",
            "language": "en"
        },
        "user": {
            "uuid": "43533744-1y-6i-4j-1p-nn0if7kfzq8kw3am4i17-1664982959269"
        }
    },
    "widget": {
        "items": [
            {
                "entity": "content",
                "rfk_id": "rfkid_7",
                "search": {
                    "content": {},
                    "facet": {
                        "all": true,
                        "max": 100,
                        "types": [
                            {
                                "name": "type",
                                "filter": {
                                    "values": [
                                        "facetid_eyJ0eXBlIjoiZXEiLCJuYW1lIjoicHJvZHVjdCI
                                    ],
                                    "type": "or"
                                }
                            }
                        ]
                    },
                    "query": {
                        "keyphrase": "cloud portal"
                    },
                    "sort": {
                        "choices": true,
```

```
                    "value": [
                        {
                            "order": "asc",
                            "name": "desc_name"
                        }
                    ]
                }
            }
        }
    ]
  }
}
```

In the `sort` object, pass the API name of the sorting option an administrator created when they configured sorting. Here, you use the sorting option called *desc_name*.

Redirect the visitor to the content page

After the visitor sorts content, they click a content item. To handle this, create a redirect to the content item's *URL* when the visitor clicks anywhere within the content item.

Send events to Search

As visitors interact with your page by clicking on widgets, logging in, upvoting articles, and so on, you must track those interactions and send them to Sitecore Search using the Events API.

Sitecore Search uses events to track visitor data across your site. This data is used for search ranking, generating analytics, and personalizing search results and recommendations. You can send events using the Events REST API.

In this walkthrough, send the following events to for each action the visitor takes:

Visitor action	Events to send
Land on a search results page and see content items and facets	`view widget`
	`view entity_page`
Enter a keyword in the search bar and see suggestions in a preview search widget	`view widget`
Click a suggestion and see a search results page with new results	`click widget` In the request, pass `suggestion` as the value of `action_cause`.
	`view widget`
Select a facet value to filter results	`view widget`
	`click widget` In the request, pass `filter` as the value of `action_cause`.
Sort results by descending order of content titles	`view widget`
Click a content item and go to the page for that content	`click widget`. In the request, pass `entity` as the value of `action_cause`.
	`view entity_page`