



ENTERPRISE KNOWLEDGE

Global Knowledge & Information Management Services

Five Steps to Implement Search with a Knowledge Graph

James Midkiff

Knowledge Graphs and Search are commonly linked together to support search use cases such as:

- returning contextual relationships with search results;
- displaying relevant topics in a knowledge panel; or
- powering an expert finder.

These advanced use cases enable an organization to provide more domain context and organizational information to users, reducing user time spent searching and improving a user's ability to discover new content through recommendations. The five steps that EK recommends to implement search with a knowledge graph are as follows.

1. Analyze the Search Content
2. Develop an Ontology for the Knowledge Graph
3. Design the User Search Experience
4. Ingest the Data
5. Implement and Iterate

Depending on your workflow, these steps may not occur in a waterfall order, so keep in mind that, for example, step 3 could be started while step 2 is still in progress. Also, these steps are analogous to the steps necessary to implement a semantic architecture.



Step 1: Define the Use Case

The first step to a successful knowledge graph search implementation is to analyze the information available for users. If you are just starting a search effort, start small and analyze a handful of data sources that contain key information that end-users always need. This step often involves interviews with business and technical data source owners as well as **users** to answer the following questions. At the end of this step, you will have a collection of information about data source content with answers to what, where, and how the information can be leveraged.

What information is available?

We want to identify each type of information available from a data source. If we are analyzing a content management system, it may contain deliverables and reports. However, do not stop there. Continue asking questions to dive deeper into *what* is available.

- What **metadata fields** exist on a report?
- Can we **segment the deliverables** at all? i.e. Can we retrieve or link to the pages separately?
- What **users** worked on this document?



As you dive deep into the content, you will surface key pieces of information that can be put together to solve user needs.

Where is the information and how do we get it?

These two questions inform the development process later on and ensure that information is actually available for use. We find it key to meet with data source technical owners as they will be

able to figure out where information lives within a system, how it is generated, and, most importantly, how we can extract the information for use in the knowledge graph. It is best to start this conversation early as often there may be security concerns or development steps that need to be taken in order to build out an integration point.

How is the information related to other information?

Once you know what information is available, facilitate a conversation with the business owners to determine where the information originates and how the information relates to other data sources. With this question, we are hoping to surface concepts like

- **content lifecycle** processes that could be tracked to add more context to search;
- opportunities to **combine information** from multiple data sources together; or
- **new data sources** that we should analyze in the future.



Knowledge graphs are great at representing and querying interconnected data as well as providing means to infer additional relationships. We want to take advantage of this feature as much as possible since it helps drive the search user interface design (that we will talk about later).

By collecting the answers to these questions, you are making it easier to take the next steps in implementing search. If step one is still unclear, think of it like **designing a content type** and consider that our main goal is to create custom search results that utilizes all information at your organization. Understanding not only where information is but where it comes from and how it will change over time is crucial to the next step of modeling the information.

Step Two: Develop an Ontology for the Knowledge Graph

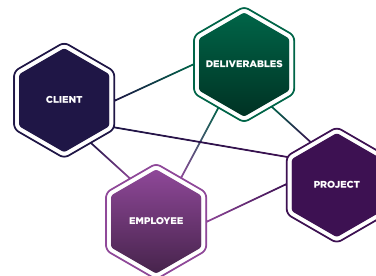
At the end of the first step, we have a large amount of data describing the information contained in all of our data sources and how they relate to each other. The next step is to figure out how we can leverage the information to answer user questions and build a model to support them.

This model, academically referred to as an **ontology**, is the data model of the knowledge graph that we will be piecing together in step four. Define the User Questions

We strongly believe that the best way to ensure any solution's success is to gather requirements from the users. EK usually runs a **search workshop** to facilitate a session with end-users and business stakeholders to elicit feature requirements and determine what information users find helpful. In step one, you collected a lot of data describing the types of information available. Use this data to ask pointed questions, gauging user interest in the data you uncovered. Work with the group to determine how they would like to see information displayed and what questions they would ask of the data. This is an opportunity for users and stakeholders to think outside the box and come up with their ideal solution, no matter how out of scope it may seem at the time. Every idea may be used later while iterating on the solution or to influence the creation of similar features.

Determine the Classes, Attributes, and Relationships

Almost all information can be represented using classes, their attributes, and the relationships between them. Once you know the questions that users want to ask and the requirements for the solution, you can begin to break down the data from step one into classes. For this process, you can follow the following questions.

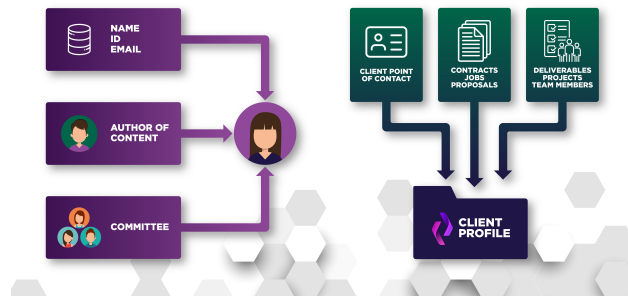


- What **types of information** does search need to display?
(i.e. employees, deliverables)
- For each type, what **properties** are necessary to display the information in an intuitive way for users?
(i.e. do end-users need to see the employee's email?)
- For each type, what **relationships** exist to other types of information?
(i.e. are employees related to deliverables at all?)

A majority of these questions will leverage the data collected in step one, but the data is now tuned to match the needs of the users and stakeholders. Use [ontology design best practices](#) to validate the reusability and scalability of the data model. The selected classes (types), properties (attributes), and relationships form the initial ontology.

Map the Data Sources to the Ontology

It is critical to keep a mapping of the data source information to the ontology so that you can maintain and upgrade the ontology in future iterations. Keep track of where each type of information originates, how attributes are calculated, and what steps are taken to extrapolate relationships within the information. While developing the mapping, pull a sample set of information from the data sources and mock up some data. Use this mocked data to validate the data types that should be used for each attribute with a technical member of the team. This ensures that the mapping has realistic inputs and outputs that can be leveraged when creating the data pipelines in step four.



Use the knowledge you already have to create complete views of your organization's information, including people and clients.

Step Three: Design the User Search Experience

In steps one and two, we put our full attention on the data sources, interpreting the available information into a data model that will enable us to populate a knowledge graph. In this step, we want to shift our focus to the end-users and make sure we build a search solution that will solve user needs through an intuitive interface, leveraging the full capabilities of a knowledge graph.

Define the Search User Stories

Work with the application stakeholders and users to define user stories that will help guide the user interface design. Here's a blog we have written about [three key benefits of user stories](#).



Perspective

Define the search and interface requirements (not features!) from the view of a user. What does a user need from the search solution?



Purpose

Determine *why* requirements are needed and the benefits they bring. This enables the team to brainstorm and build the best feature to meet the requirements.



Priority

Work with users and stakeholders to order the requirements. A prioritized backlog of requirements ensures the team delivers high interest items first.

When defining the user stories, keep an eye out for use cases that could be solved through an [action-oriented search result](#). We want to note what data points are important to users so that we can best leverage them in the design process to enable users to take immediate action.

Design using Search Best Practices

Start simple and include your basic search features, the search bar, results, and facets. These basic features ensure that anyone, regardless of their background, can find and discover information within search. Facilitate design workshop sessions with users and stakeholders to design search results for types of information and include [search best practices](#).



Use a consistent view when displaying the same content on multiple pages.

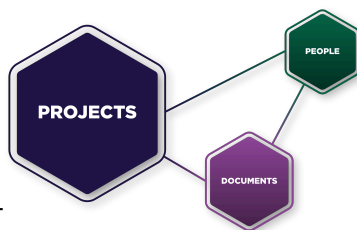
Determine which attributes and relationships in the data need to be highlighted in search results versus those that should be only displayed in spots requiring an additional click, like an accordion dropdown or an entirely new page. When designing the interface, standardize how users will interact with the interface and different content types. The consistent interactions build trust with users and ensures that interaction with search is intuitive.

Innovate with Knowledge Graph Search Features

Up until now, step three has been all about designing the search solution using search design best practices. Now that we have that baseline, we want to include knowledge graph specific use cases like the below.

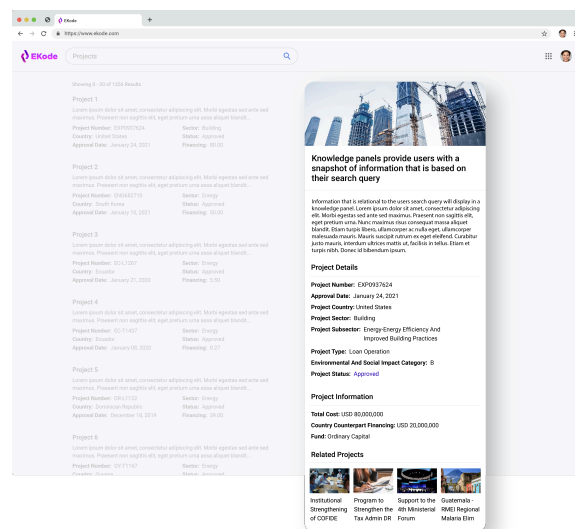
Identify the Search Subject

Use named-entity recognition (NER) or a knowledge graph entity lookup to identify *what* a user is looking for and present the user with all relevant compiled information about that entity. For example, imagine the search information includes **people, documents, and projects**. If a user searches for the id of a project, design a project page that the user is redirected to that includes all of the project metadata, links to the documents associated with the project, and all team members that worked on the project. Creating these encyclopedic-like pages for an organization's content can greatly improve the user's ability to find the information they are looking for.



Extend the Search Results

Along the same line as the above, surface additional information, **properties and relationships**, about the search query and search results from the knowledge graph. If a specific term or entity is recognized in the search query, use that to populate a knowledge panel on the right hand side with all relevant information about that term or entity. A knowledge panel provides users with a snapshot of information based on their search query. When displaying the knowledge panel and search results, pull the most up-to-date contextual information about a search result from the knowledge graph. For example, contextual information could include project statuses, most recent documents, or most similar content within the knowledge graph by metadata.



A knowledge panel collects and highlights project details in one place for a user search.

Natural Language Search Across Data

One of the most powerful resources for a knowledge graph search is natural language processing (NLP). NLP enables search to recognize entities in the graph as well as user intent. In one of our knowledge graph projects, EK **developed an NLP-based search** that recognized what entities a user was asking for and used that context to automatically collect and prioritize big data in a tabular format for analysts to review. This gave business analysts quick access to the data insights they needed from multiple large datasets. The ability to recognize the *intent* behind a user's query enables the search interface to adapt and provide specialized answers to the most important questions.

Step Four: Ingest the Data

Steps one, two, and three focus on analyzing, prepping, and designing the knowledge graph search solution. Now that we have our initial plan, we can pull the data together through extract, transform, and load (ETL) pipelines and populate our knowledge graph.

Index the Data Source Information

Using the data collected in step one, build out the integrations with each of the required data sources. When possible, use application programming interfaces (APIs) or other feeds to extract content from the source systems. If this is not possible, database connections or temporary data exports may be required in order to proof out the integrations. Next, determine how content will be indexed from the sources by answering the following questions.

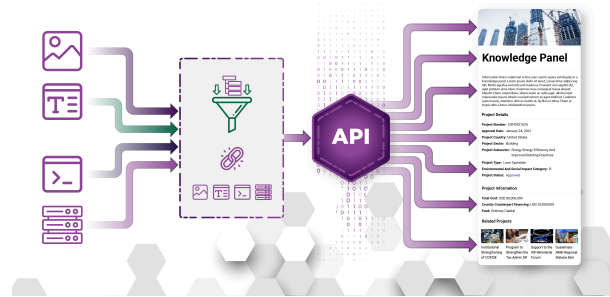
- What **amount** of content should be extracted each time the pipeline runs?
- How **often** should the content be indexed?
- Does the content from this data source need to be **combined** with any other data source?



There are numerous types of indexing techniques in order to ensure that the knowledge graph and search data are kept accurate and up-to-date without overloading the search indexing or data pipelines.

Transform Information using the Ontology Mapping

When designing the ETL pipelines, reference the ontology and ontology data source mapping to ensure that all information is transformed into the expected format. In most cases, this involves using transformation techniques like object mapping (i.e. Entity Framework) or XSLT to transform information from the source format into a graph data format (i.e. RDF) or into a document format for search (i.e. JSON). This is the first time that all information from a data source is being transformed so expect some data quality issues. Required fields may not always be present in the data, the values may not match the expected type, and data standardization issues may need to be addressed. Work with your stakeholders and data source owners to determine where and how issues should be addressed.



Transform and enrich your knowledge with a consistent vocabulary to populate a knowledge graph and display that information to users.

Enrich the Information with Context

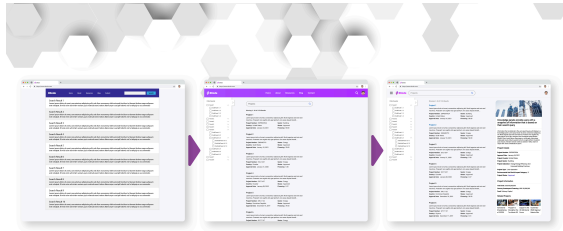
One key piece to developing relationships between information from various sources is to leverage NER or an existing **taxonomy**. As content is pulled into the knowledge graph and search, the metadata fields provided with each type of information may not be enough. Combining information together from multiple sources builds a better picture of each information type, but some of the best sources of similarity reasoning and clustering will come from associating content with entities through **auto-tagging a taxonomy** or NER and topic modeling of terms. When designing the ETL pipelines to bring in data, consider how content may be enriched by adding in auto-tagging and NER techniques to the pipelines.

Step Five: Implement and Iterate

At this point, the indexed information is within the knowledge graph and search platform. In this step, build out a prioritized feature set based on the search designs from step three. Depending on the selected development stack, it may be beneficial to build an API layer on top of the knowledge graph and search platform and leverage these APIs to pull data into the user interface. In order to get the interface in front of stakeholders quickly, you may need to leverage some of the sample data you created in step 2.

Developing a **user-centric product** requires feedback early and often. Validate the designs with both stakeholders and users through demos and user testing. Demos allow stakeholders to give instant feedback on the solution as soon as it is available. For user testing, provide users with tasks to perform and observe how users perform the task. It is important to note where users click, where their eyes are drawn to first, and how

design choices impact the flow of the website navigation.



Prioritize and iterate on the user interface based on user feedback and testing.

Make sure to continue to explore the unique features of knowledge graphs. Incorporating new relevant sources with relationships to existing data can help cover edge cases of your search queries that are not yet answered. Highlighting inferences made through traversing the knowledge graph at query time, can bring users previously undiscovered steps.

Always remember search is a journey.

Finally, iterate on the solution and respond to feedback. Steps one through five are meant to be repeated over and over as

- new data sources and information is considered for search;
- users ask different questions that requires updating the ontology;
- designs adapt to feedback and testing to provide a more intuitive user experience;

- pipelines are extended to extract more information from the data sources; and
- features and design changes are required to the end solution.

Additionally, don't forget to start small--EK recommends building out an end-to-end system, completing steps 1-5 in order for a subset of content and prioritized use cases. Use this first iteration to test capabilities and identify any integration risks or concerns.

Conclusion

These steps ensure that your organization builds the right search solution, creating a knowledge graph that answers your users' questions and surfaces the results in an intuitive interface. Building search on top of a knowledge graph enables your organization to provide tailored, advanced search features as well as create a foundation of organizational knowledge that can be leveraged for other use cases such as chatbots, recommendation engines, and data analysis.

Interested in expanding your organization's search to leverage the capabilities of a knowledge graph? [Contact us](#) and let's work together to build a search solution that fits your organization's needs.

Enterprise Knowledge (EK) is a services firm that integrates Knowledge Management, Information Management, Information Technology, and Agile Approaches to deliver comprehensive solutions. Our mission is to form true partnerships with our clients, listening and collaborating to create tailored, practical, and results-oriented solutions that enable them to thrive and adapt to changing needs.

Our core services include strategy, design, and development of Knowledge and Information Management systems, with proven approaches for Taxonomy Design, Project Strategy and Road Mapping, Brand and Content Strategy, Change Management and Communication, and Agile Transformation and Facilitation. At the heart of these services, we always focus on working alongside our clients to understand their needs, ensuring we can provide practical and achievable solutions on an iterative, ongoing basis.

info@enterprise-knowledge.com | 571-403-1109 | @EKConsulting