

# KGSnap!: query Knowledge Graphs by Snap!

Vincenzo Offertucci<sup>1</sup>, Maria Angela Pellegrino<sup>1</sup>[0000–0001–8927–5833], and  
Vittorio Scarano<sup>1</sup>[0000–0001–8437–5253]

Dipartimento di Informatica, Università degli Studi di Salerno, Italy  
v.offertucci@studenti.unisa.it, {mapellegrino,vitsca}@unisa.it

**Abstract.** As the block programming paradigm has been successfully used to teach programming skills, this demo proposes **KGSnap!**, an extension of the block-based programming environment **Snap!**, which allows lay users to build and run queries on a SPARQL endpoint. The proposed approach has the potential to enable lay users to access knowledge graphs without requiring technical skills in query languages.

**Keywords:** SPARQL · Block-based programming · Snap!

## 1 Introduction & Related Work

The Semantic Web technologies are increasingly used to model any field of interest, increasing the quantity and diversity of available data modeled by Knowledge Graphs (KGs). However, its potential risks to be left untapped due to the SPARQL complexity [6]. Lay users interested in consuming KGs require tools to mitigate the technical challenges SPARQL poses.

Block programming languages that guide users in dragging and connecting fragments shaped like jigsaw puzzle pieces have successfully introduced programming to non-experts [3]. Just think the vast exploitation of Blockly<sup>1</sup> as part of code.org’s Hour of Code, Scratch [4] to create animations and games and MIT App Inventor [8] to build Android Apps. The proposal of letting lay users query data by block-based programming is not new. **SQheLper** [2], **DBLearn** [7], and **DBSnap++** [5] are block-based programming interfaces to query databases. In the Semantic Web community, Bottoni and Ceriani [1] proposed **SPARQL Playground**, introducing KGs querying in Blockly. This demo proposes **KGSnap!**, an extension of **Snap!** to query KGs that expose SPARQL endpoints.

## 2 KGSnap!: query Knowledge Graphs by Snap!

**Snap!**<sup>2</sup> (formerly Build Your Own Blocks) is a free, open-source, block-based educational graphical programming language and online community allowing learners to explore, create, and remix interactive animations, games, and stories while learning about mathematical and computational ideas.

---

<sup>1</sup>Blockly, <https://developers.google.com/blockly>

<sup>2</sup>Snap! <https://snap.berkeley.edu>

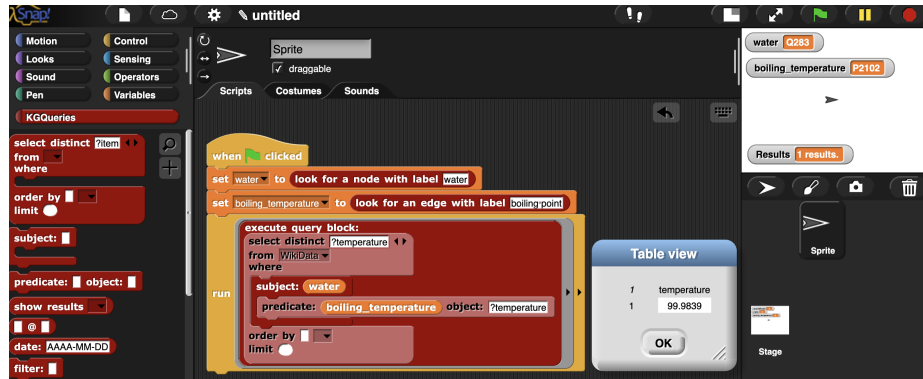


Fig. 1. KGSnap! interface.

KGSnap! (visible in Fig. 1, freely accessible online<sup>3</sup> and available on GitHub with an Open Source license<sup>4</sup>) extends the Snap! architecture by introducing the possibility to perform SELECT queries on KGs provided with a working SPARQL endpoint. By default, KGSnap! is configured to query Wikidata. However, users can easily introduce any SPARQL endpoint of interest by a dedicated block, `define a new endpoint`, shown in Fig. 2 among others. KGSnap! mimics the structure of SPARQL queries to follow the philosophy of block programming to guide learners to experiment gradually with the underlying language and, in the end, to be able to switch to programming in that language. Hence, SPARQL



Fig. 2. Blocks implemented in KGSnap!.

<sup>3</sup>KGSnap!: <https://isislab-unisa.github.io/Snap/snap.html>

<sup>4</sup>KGSnap! <https://github.com/isislab-unisa/KnowledgeGraphsAndSnap>

queries are formulated by specifying triples (subject, predicate, object). The complete set of supported features is visible in Fig. 2 and concerns entity resolution from user-defined labels, components to assemble SELECT queries cov-

The screenshot shows a Snap! script with the following blocks:

- when clicked**
- set author to look for an edge with label writer**
- set Rowling to look for a node with label J.K.Rowling**
- set published to look for an edge with label published**
- set character to look for an edge with label characters**
- set Harry Potter to look for a node with label HarryJamesPotter**
- run** block containing:
  - select distinct** ?book ?label
  - from** WikiData
  - where**
    - subject:** ?book
    - predicate:** author **object:** Rowling
    - predicate:** published **object:** ?date
    - predicate:** character **object:** Harry Potter
    - predicate:** rdfs:label **object:** ?label
  - filter:** ?date > date: 2000-01-01
  - filter:** language of: ?label is en
  - order by** ?date ASC
  - limit** 5

Below the script is a **Table view** showing the results:

	book	label
5		
1	http://www.wikidata.org/entity/Q46751	Harry Potter and the Goblet of Fire
2	http://www.wikidata.org/entity/Q80817	Harry Potter and the Order of the Phoenix
3	http://www.wikidata.org/entity/Q46887	Harry Potter and the Half-Blood Prince
4	http://www.wikidata.org/entity/Q46758	Harry Potter and the Deathly Hallows
5	http://www.wikidata.org/entity/Q20711488	Harry Potter and the Cursed Child

An **OK** button is visible at the bottom of the table view.

**Fig. 3.** Use case of KGSnap!. We query Wikidata to retrieve books authored by J.K.Rowling with Harry Potter as character.

