User Manual for iFixit Knowledge Graph Application

# Introduction

The application built is designed to create a knowledge graph and ontology around iFixit instruction manuals. It allows users to browse procedures, tools, and items while enforcing certain relationships and querying capabilities. This manual provides an overview of the ontology, key queries, and instructions on how to interact with the knowledge graph.

# Overview of the Schema and Ontology Rules

The main concepts of the ontology are:  
 - Procedure: Represents a set of steps for fixing an item.  
 - Item: Any physical object described in the procedure.  
 - Part: A component that makes up an item.  
 - Tool: Tools used within a procedure's steps.  
 - Step: Individual actions or tasks within a procedure.  
   
 Relationships include:  
 - Part-of: Specifies if one item is a part of another.  
 - Uses Tool: Links a tool to the steps where it is used.  
 - Sub-procedure: A procedure can be part of another procedure for the same or related items.

# Example Queries

Example SPARQL Queries:  
  
 1. Find all procedures with more than 6 steps:  
 SELECT ?procedure WHERE { ?procedure :hasStep ?step . FILTER (COUNT(?step) > 6) } GROUP BY ?procedure  
  
 2. Find all items with more than 10 procedures:  
 SELECT ?item WHERE { ?item :hasProcedure ?procedure . FILTER (COUNT(?procedure) > 10) } GROUP BY ?item  
  
 3. Identify tools not mentioned in procedure steps:  
 SELECT ?tool WHERE { ?procedure :hasTool ?tool . FILTER NOT EXISTS { ?step :usesTool ?tool } }  
  
 4. Find hazardous steps:  
 SELECT ?step WHERE { ?step :description ?desc . FILTER (CONTAINS(?desc, "careful") || CONTAINS(?desc, "dangerous")) }

# Instructions for Adding, Updating, and Removing Data

To add data:  
 Use RDFLib's 'add' function to insert a triple into the graph.  
 Example: g.add((procedure\_uri, URIRef("http://ontology/usesTool"), tool\_uri))  
  
 To update data:  
 Remove the old triple and insert a new one.  
 Example: g.remove((procedure\_uri, URIRef("http://ontology/usesTool"), old\_tool\_uri))  
  
 To remove data:  
 Simply use the 'remove' function.  
 Example: g.remove((procedure\_uri, URIRef("http://ontology/usesTool"), tool\_uri))

# Running the Application

Command-line:  
 To run the application via command line, execute the following:  
 python load\_data.py  
 python query\_ontology.py  
  
 Flask Application:  
 To start the Flask application, run:  
 flask run  
 Then access the application on localhost:5000 in a web browser.