User Guide

Version 5.x

Version 1.6

Dated 07/10/2019

Table of Contents

1. The Challenge

Globally, organisations are increasingly competing on the basis of service, and “the Customer Experience”, while simultaneously needing to continually reduce operational costs to remain commercially viable. In many organisations, people (staff) comprise the most-costly line-item.

Accordingly, Customer Service is increasingly being automated through technology with many companies having a goal of “zero-touch”; that is, where there is no human interaction with the Customer at all.

Organisations therefore attempt to provide customer-information through the following media, in the order below:

- Organisational websites
- FAQs (Frequently Asked Questions)
- IVR (Interactive Voice Response) telephone systems

This approach is failing across the board with many larger organisations reporting telephone queues at their Contact Centres of an hour or more after customers have failed to find the information they need on the organisational website or in the FAQs listed, which can run to literally hundreds of pages.

Many organisational websites contain a “search” function, but most of these search functions fail to provide the information required in any useable or readily-understandable format. Most will direct customers to a web-page, or series of pages, which are difficult for many customers to correctly interpret.

2.The SimSage™ Approach

SimSage has taken a different approach to the majority of the market supplying AI bots and/or Search solutions; one that is proving highly-effective. SimSage provides an integrated AI bot and search solution that not only provides a verified answer to customer queries from a managed “knowledge base”, but also provides additional relevant information from chosen, specific information repositories.

On deployment in any organisational environment, SimSage is specifically “tuned” to that industry’s/organisation’s environment to ensure that query results and highly-relevant, and accuracy/relevancy thresholds can be set by the organisation.

Where information is not immediately available in the knowledge-base, queries are directed to a subject-matter expert (this may be a Contact Centre operator, or another chosen representative), who can engage in a “chat” with the customer, and/or update the knowledge-base in real-time. This means rapid expansion of organisational knowledge-bases, ensuring consistency, relevancy and timeliness of information provided to customers, and reducing risk from providing inaccurate information. It also means less information carried in the heads of employees, and faster on-boarding of new staff.

Additionally:

Search

- The natural language processing approach taken within SimSage ™ is proprietary, and the core engine’s ability to “understand” the context of search queries is extremely effective.

AI “bot”

- SimSage ™ has developed a highly-effective AI bot that allows questions to be posed in “natural language” (ie. Understanding that different people ask the same question in different ways. (Eg: “What is the population of Cornwall?” vs. “How many people live in Cornwall?”).
- The AI bot is not just a hard-coded Q&A programme as many other bots are that claim to use advanced AI technologies. SimSage ™ processes queries against a proprietary “SimSage Mind™”, taking a different approach to mainstream technologies, breaking down sentences into their component linguistic parts and comparing them with information within the SimSage Mind™, and delivering a very high-success rate in terms of responses to queries posed.

Integrated Search and AI bot

- SimSage ™ has integrated the AI bot and Search functions, such that very specific answers can be provided by the bot to queries posed, and supported by “other relevant information” that is exposed by SimSage ™ Search.

SimSage can be deployed on a Public Cloud, a Private Cloud, or on servers racked inside an organisation’s firewall. It can operate with multiple knowledge-bases (useful when organisations have multiple services or they are based in multiple geographical locations with different operating parameters).

3. About this User Guide

It is intended that this guide will help System Administrators, Information Owners, and Web Designers understand the key components that make up SimSage, and in doing so be armed with the information required to install the solution within your desired location.

The first section of the User Guide is intended to be informative and help you gain an understanding of:-

The components of the knowledge base
How functions of the roles and the relationship they have with regards knowledgebase(s)

The second section is intended to help step you through the following initial set-up tasks:-

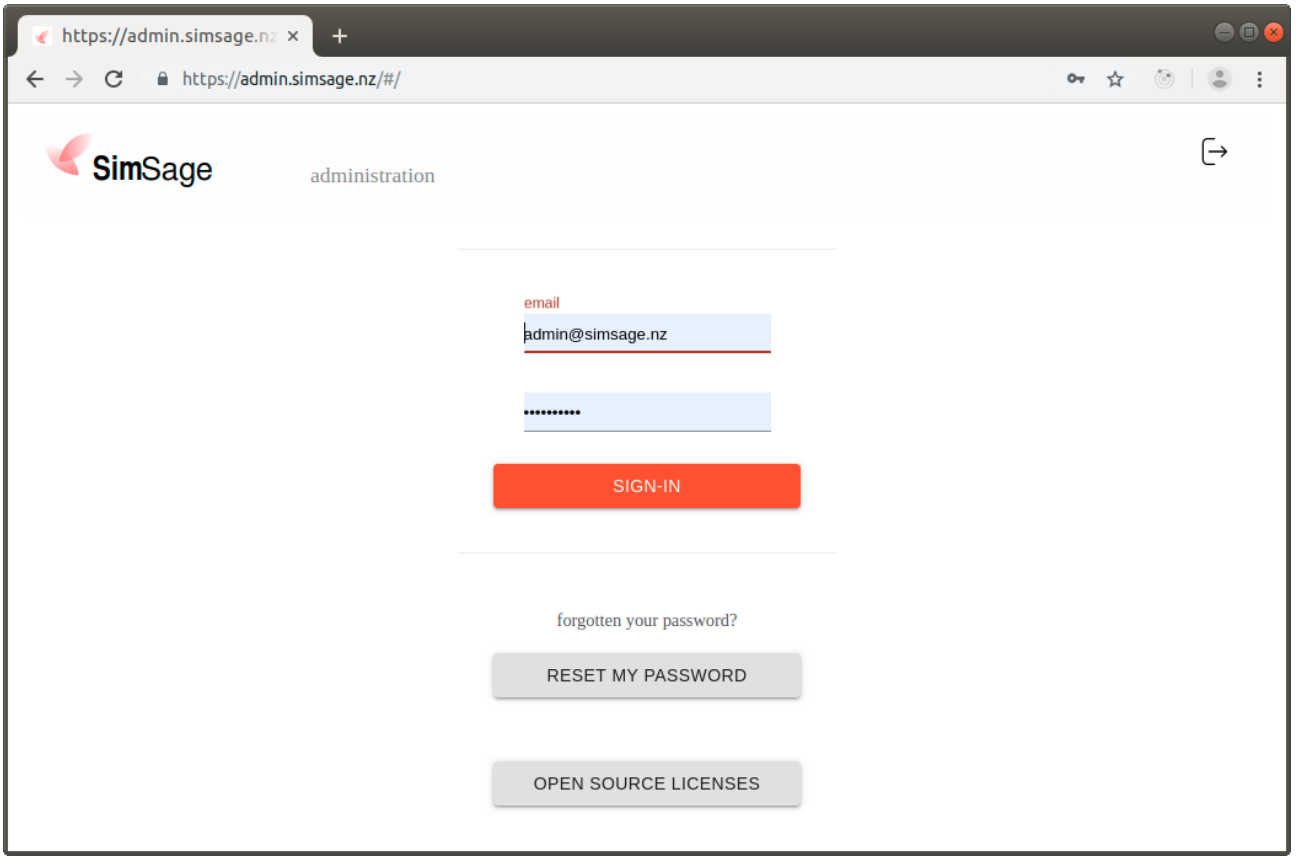
Establish a knowledgebase
Establish the users and roles required to administer and manage your knowledgebase(s)
Understand how to test that your knowledgebase is working as expected.

The third section will then guide you through the process of setting up a bot interface for the SimSage platform. This includes:-

Downloading the bot-interface and hosting it on a local web-server.
Downloading the search-interface and host it on a local web-server.
Querying the established knowledgebase(s) using these interfaces.

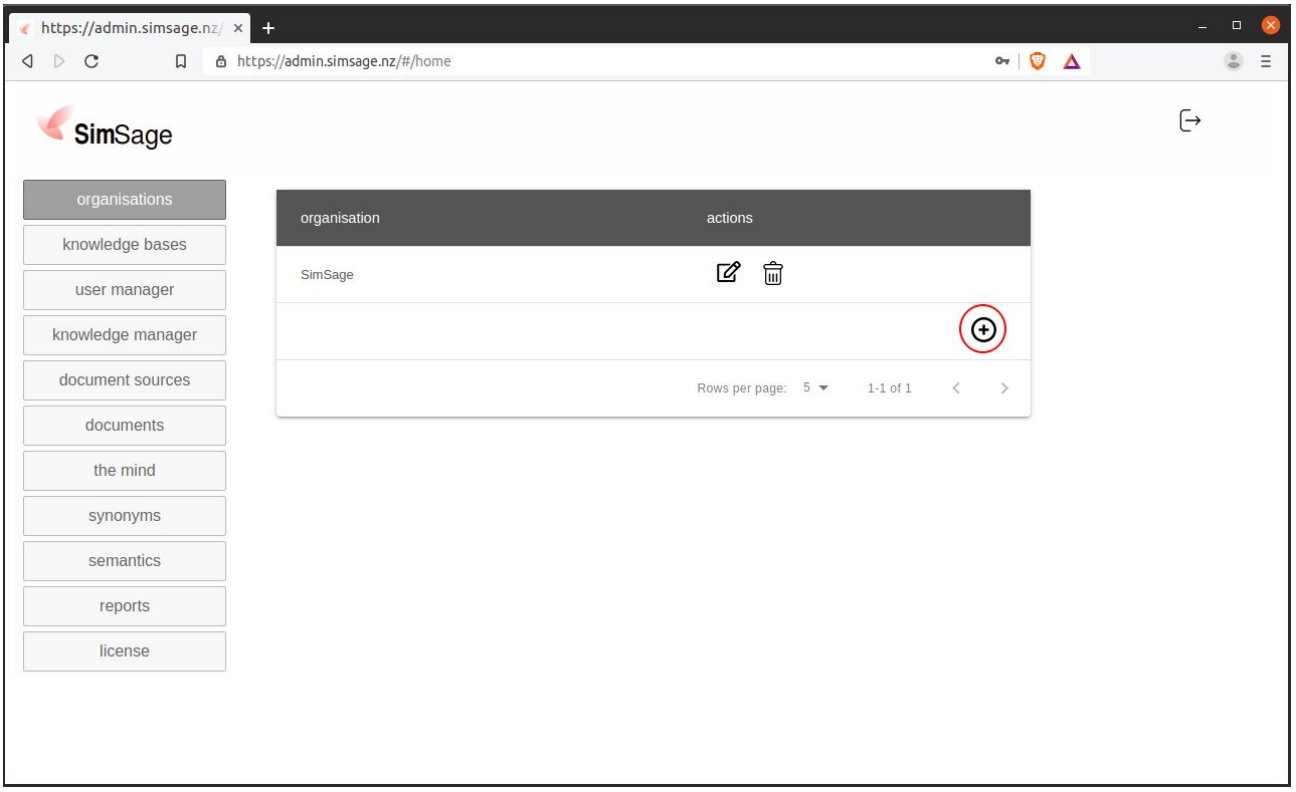
4. The knowledge base

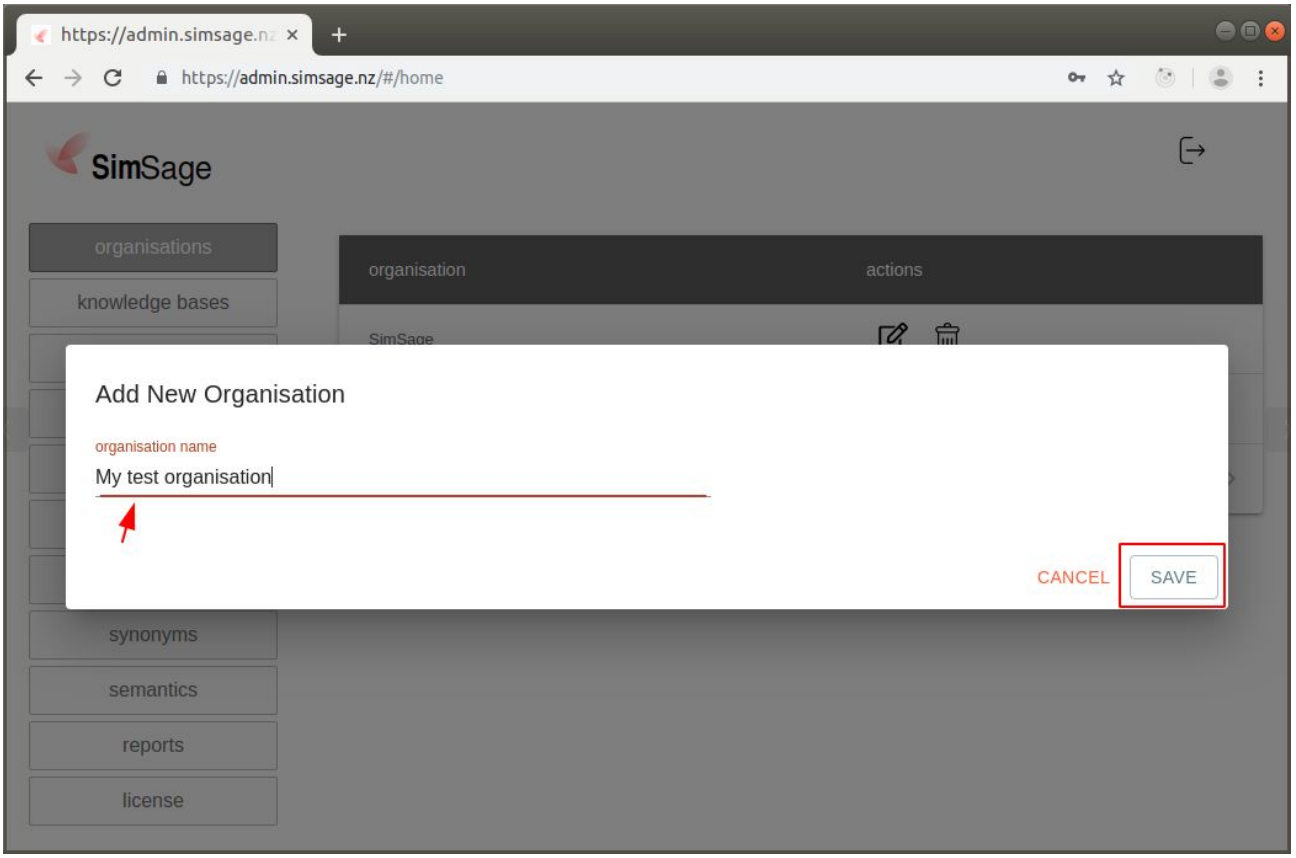
Start by signing into SimSage. You will have been given a URL and credentials for your instance that takes you directly to a sign-in page like the one shown below.



Once signed-in we arrive at the organisation's tab. Create a new organisation by clicking the

⊕ plus icon.

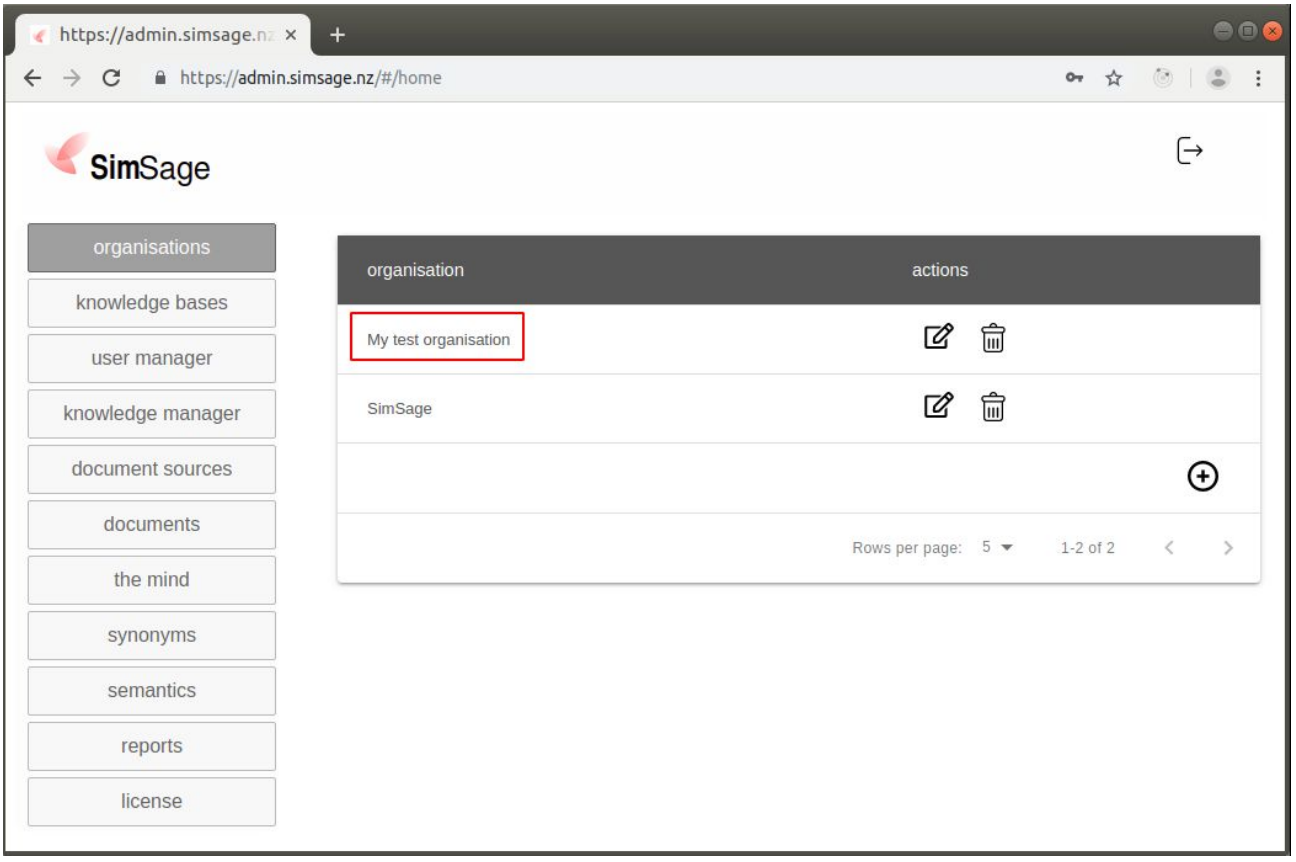




We name our new organisation. The organisation name can be anything you want it to be; we have named the one above “My test organisation”. We suggest you create a unique organisation for each and every customer. An organisation can have as many different

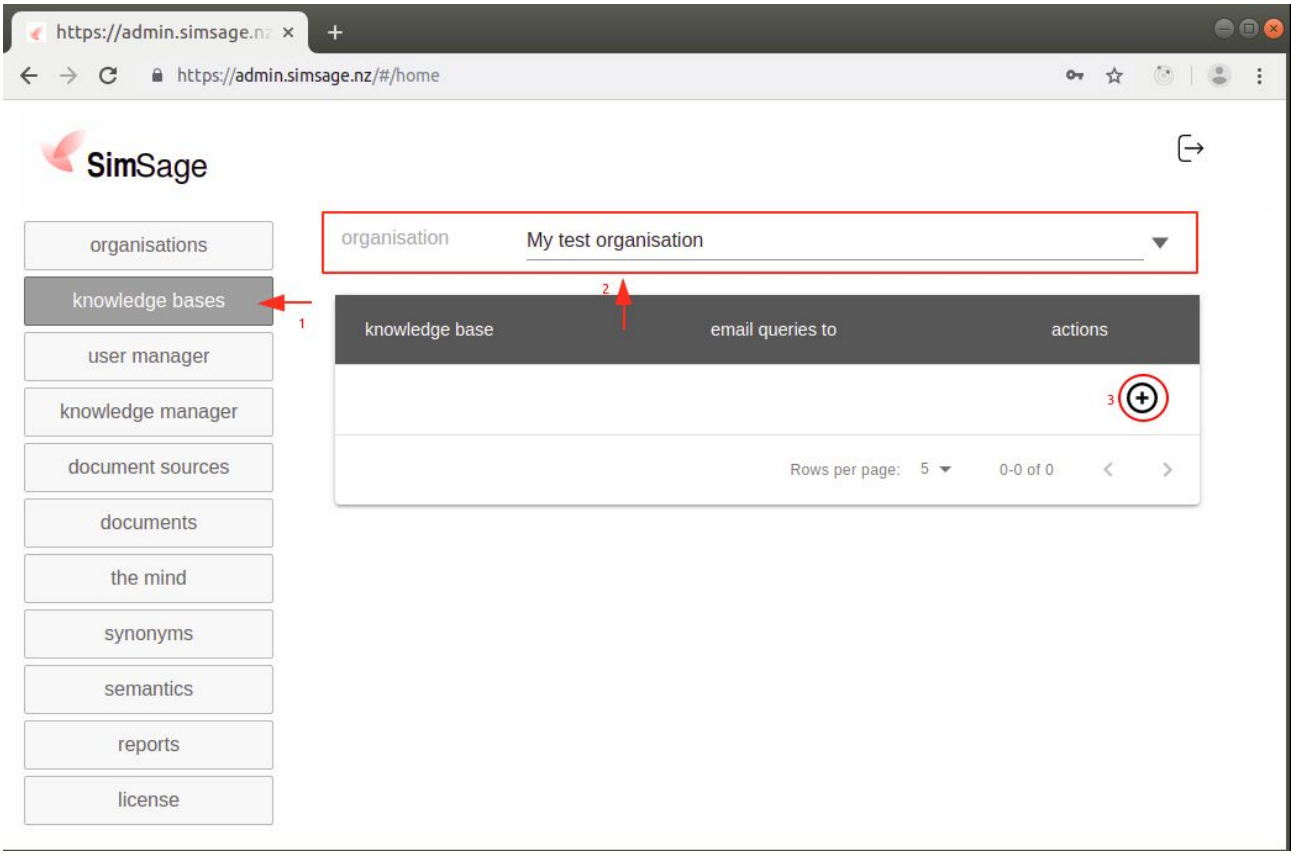
knowledge bases as you/they want. For example, a Council may have separate knowledge bases for: Dog Licensing, Parking, Parks & Recreation, etc.

Click the SAVE Save button. This will close the dialog box and show you the newly created organisation (all sorted alphabetically by name).

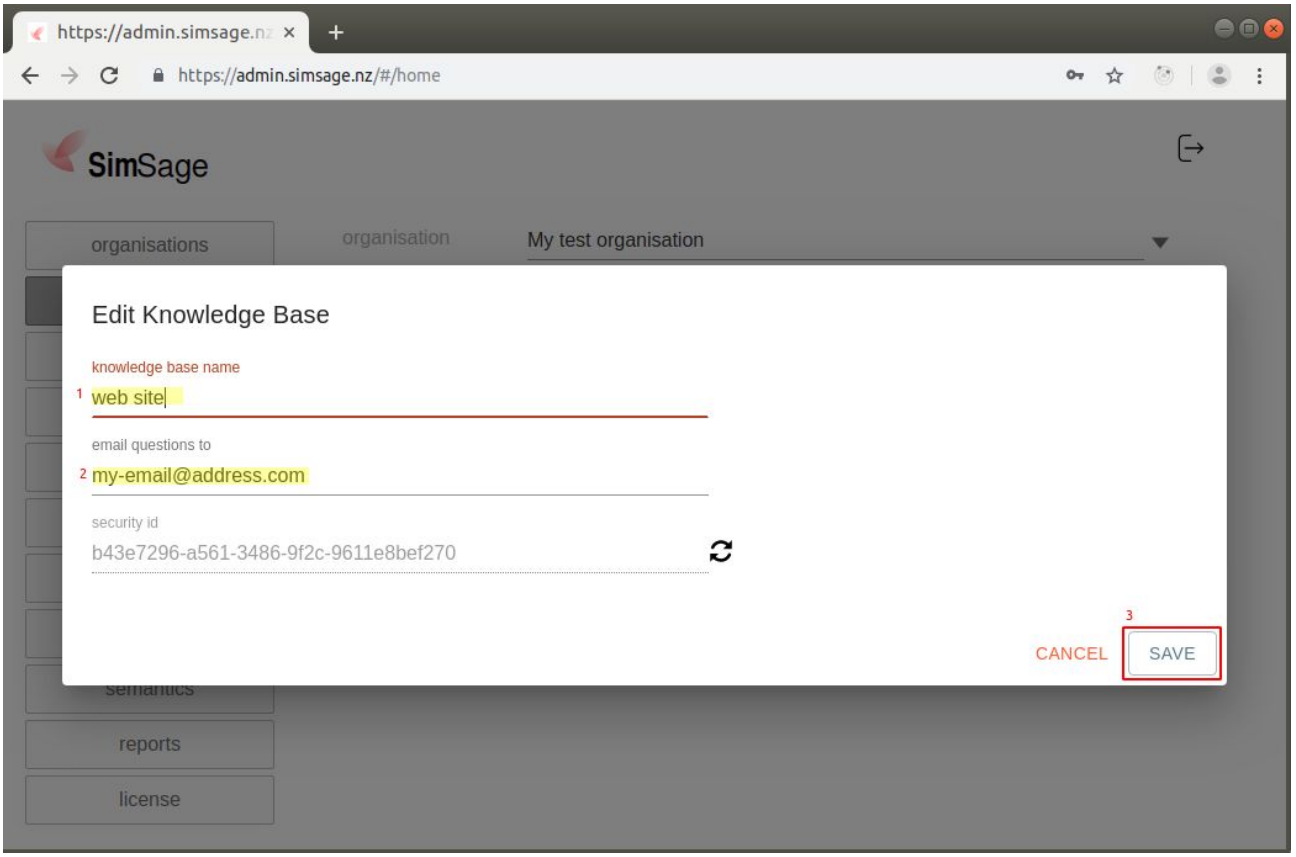


We could in principle use any existing organisation to add a new knowledge base to. However, we have opted to create a new organisation first.

Click on the knowledge bases knowledge bases button and select our newly created organisation from the drop-down box at the top of the screen.



Click its ⊕ plus button to add a new knowledge base.

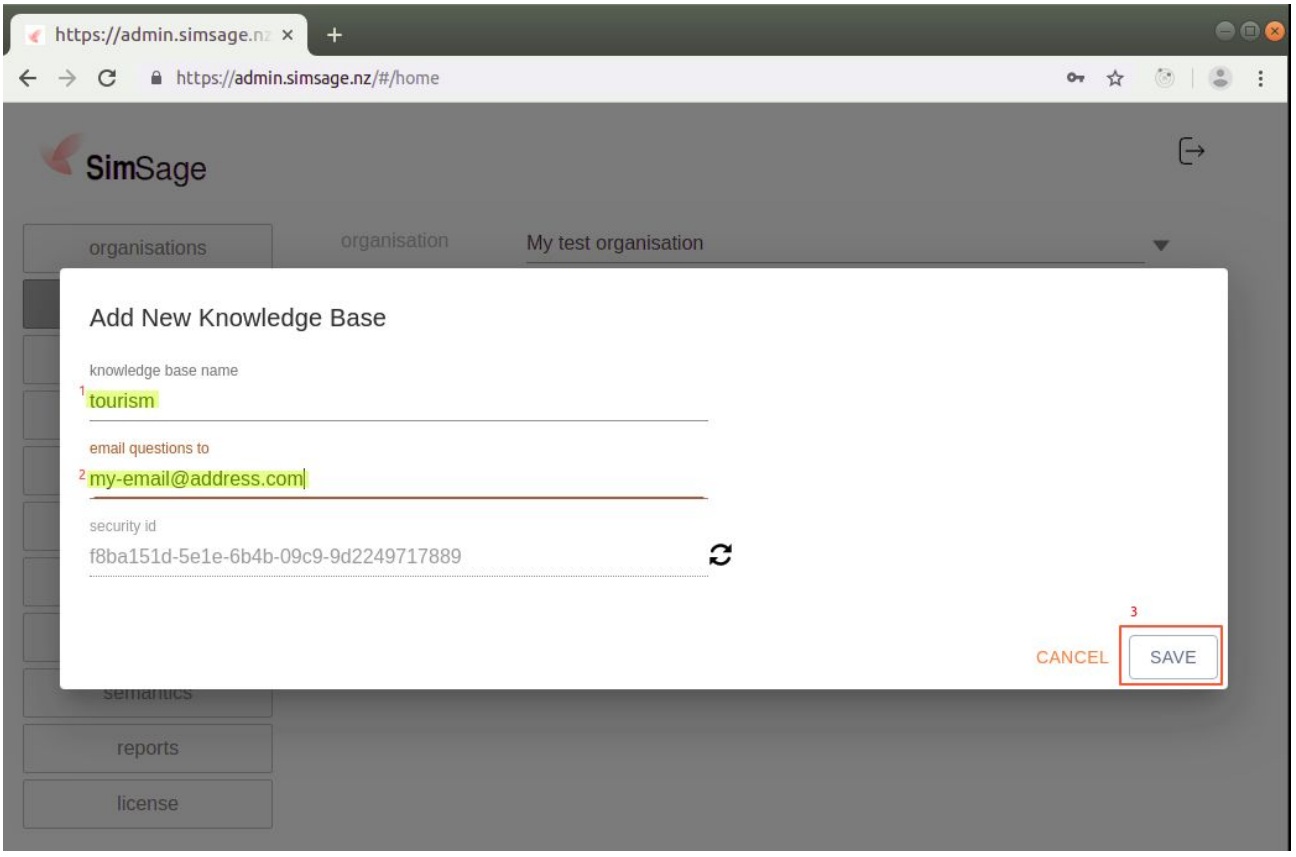


The *knowledge base name* should be descriptive if possible (e.g. web-site, or tourism). We have named our knowledge base “web site” in this example. The *email questions to* field takes a single email address. This email address will be mailed questions that fall through the gaps as a last resort where there is no available answer in the knowledge base and no “operator” available.

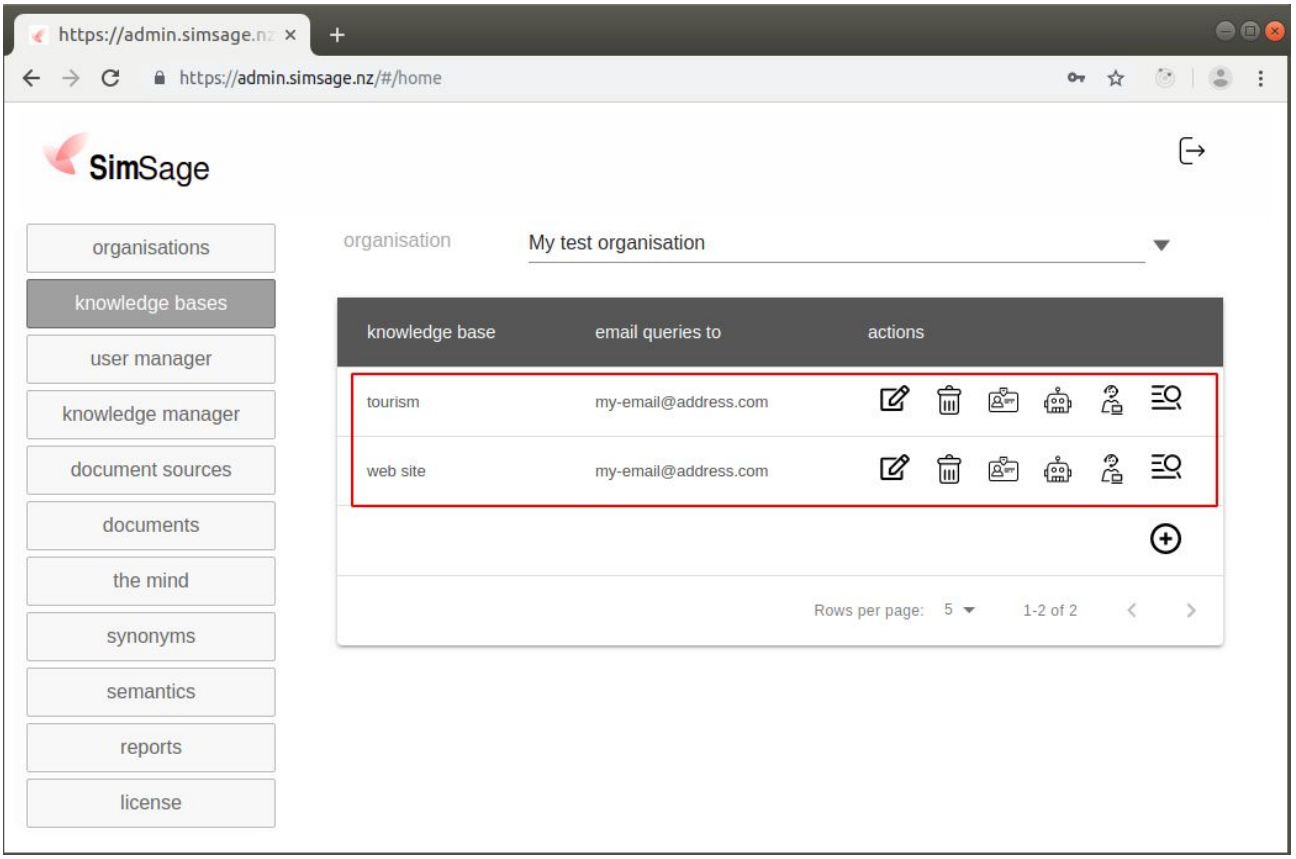
Click the SAVE save button to close this dialog box.

SimSage’s bots will first look in their knowledge bases for an answer. If no answer can be found, SimSage will then perform a semantic search for any information in file-shares or websites. If that too fails, SimSage will ask the user for their email address. SimSage will then redirect any queries the user might have to the email address specified here.

We will add one more knowledge base called “tourism”.



Click the SAVE save button to close this dialog box.
Our knowledge-bases for “My test organisation” now look something like this.



We now have several options to proceed.

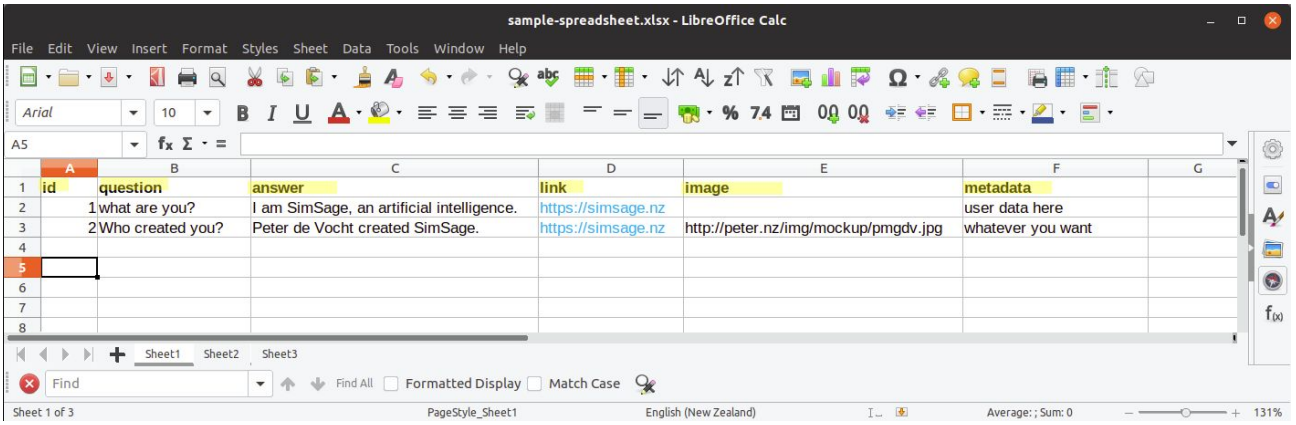
We can upload a spreadsheet to each knowledge base to teach SimSage through knowledge manager
We can manually teach a few items in each knowledge base through the mind
We can use the operator interface to teach SimSage (not covered in this manual)
We can setup semantic search for each knowledge base

3.1. Uploading a spreadsheet

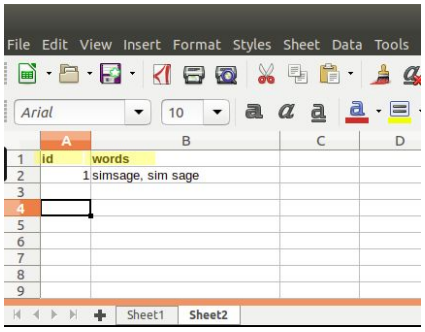
Spreadsheets must be populated in a precise manner or they will not work properly. They have an exact header which must be copied. They have a fixed number of columns which must not be changed.

You can download a sample spreadsheet from our website.
https://simsage.nz/assets/sample-spreadsheet.xlsx

This spreadsheet has two sheets. The name of the sheet is not important. The first sheet is a set of Questions and Answers shown below.

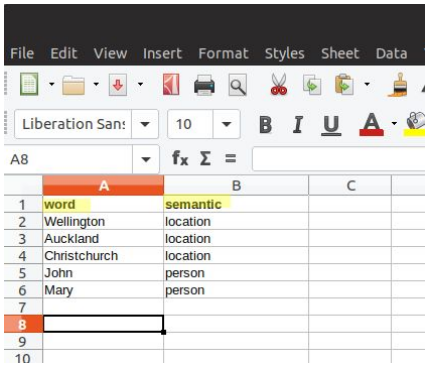


The second sheet shown below contains synonyms.



All these items have unique ids. The ids are text based and don't have to be numbers as shown. The id are the primary keys for this data, however you will update items if you change an item in the spreadsheets but keep the id the same.

The spreadsheet can include a third sheet where you define semantics. Semantics don't have ids but are defined by the word. A semantic is a linguistic generalisation relationship. Semantics are used to define concepts. For instance, "Wellington is a location" is a semantic. Defining Wellington as a location. Semantics are arbitrary. You can define your own semantics as you like. Location, person, vehicle, product, whatever your application needs.



We thus define the following spreadsheet columns.

Spreadsheet format for Question and Answers

Column name	Description
id	A unique id for a question / answer pair.
question	An expression that defines one or more related questions that relate to an answer.
answer	The answer to the question. A literal piece of text that is used to respond with when the question matches.
link	An optional link location referencing something external related to the question / answer pair. Links are optional and can be left blank.
metadata	Optional metadata returned for your convenience. The column must exist; however, its contents is optional.

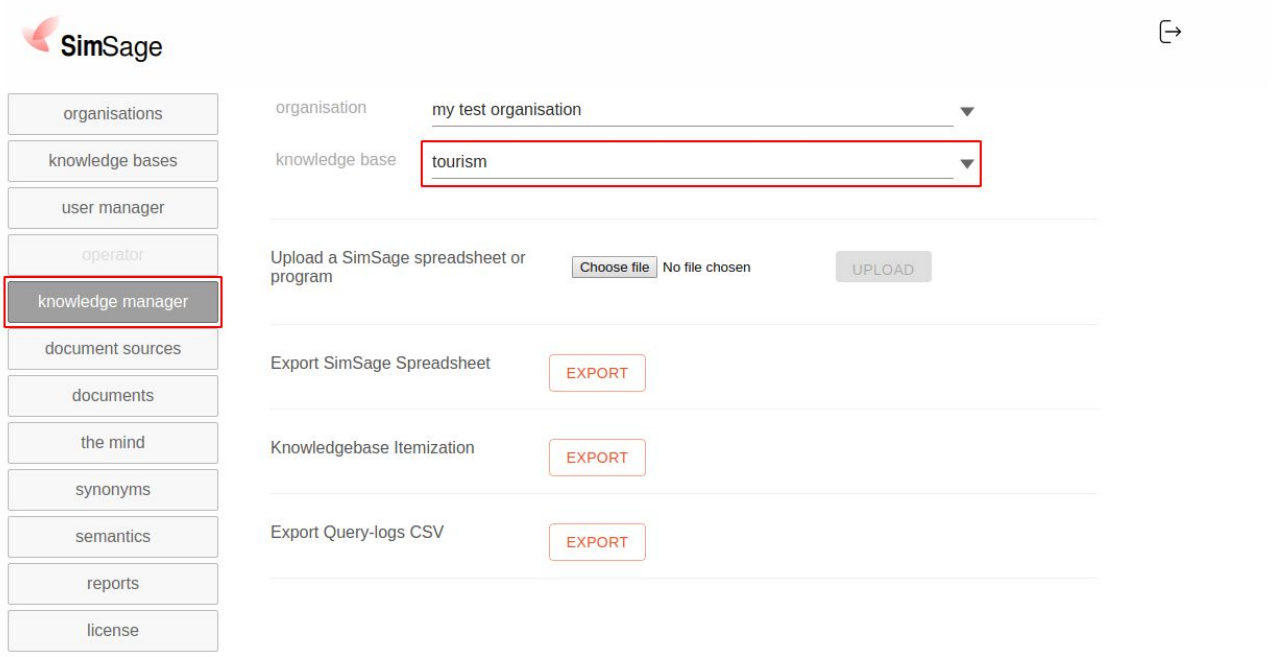
Spreadsheet format for Synonyms

Column name	Description
id	A unique id for this synonym set.
words	A comma separated list of words (must have at least two words) that form the synonyms.

Spreadsheet format for Semantics

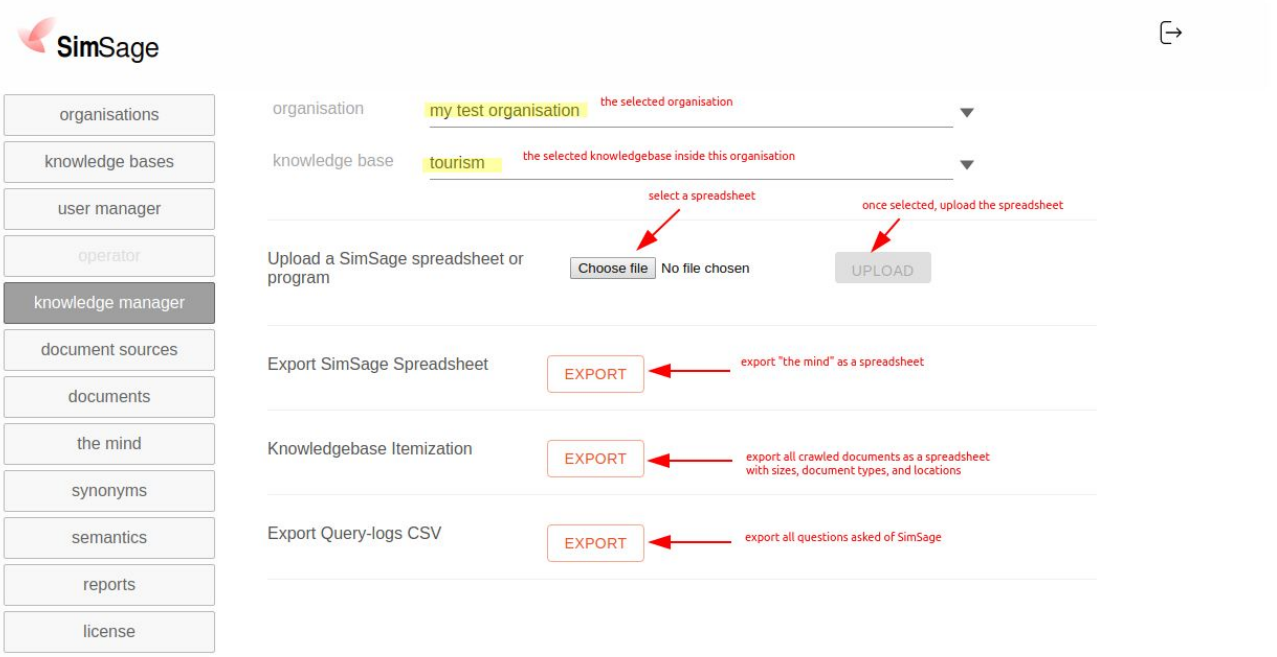
Column name	Description
word	The word (and primary key) a semantic is defined for (the less general of the two, e.g. Wellington)
semantic	The semantic, the more or less general of the two defining the semantic itself (e.g. location)

Prepare a spreadsheet like the one shown and proceed to upload it. Select the knowledge manager button and select the tourism knowledge base from the drop-down box. You should see the following screen.

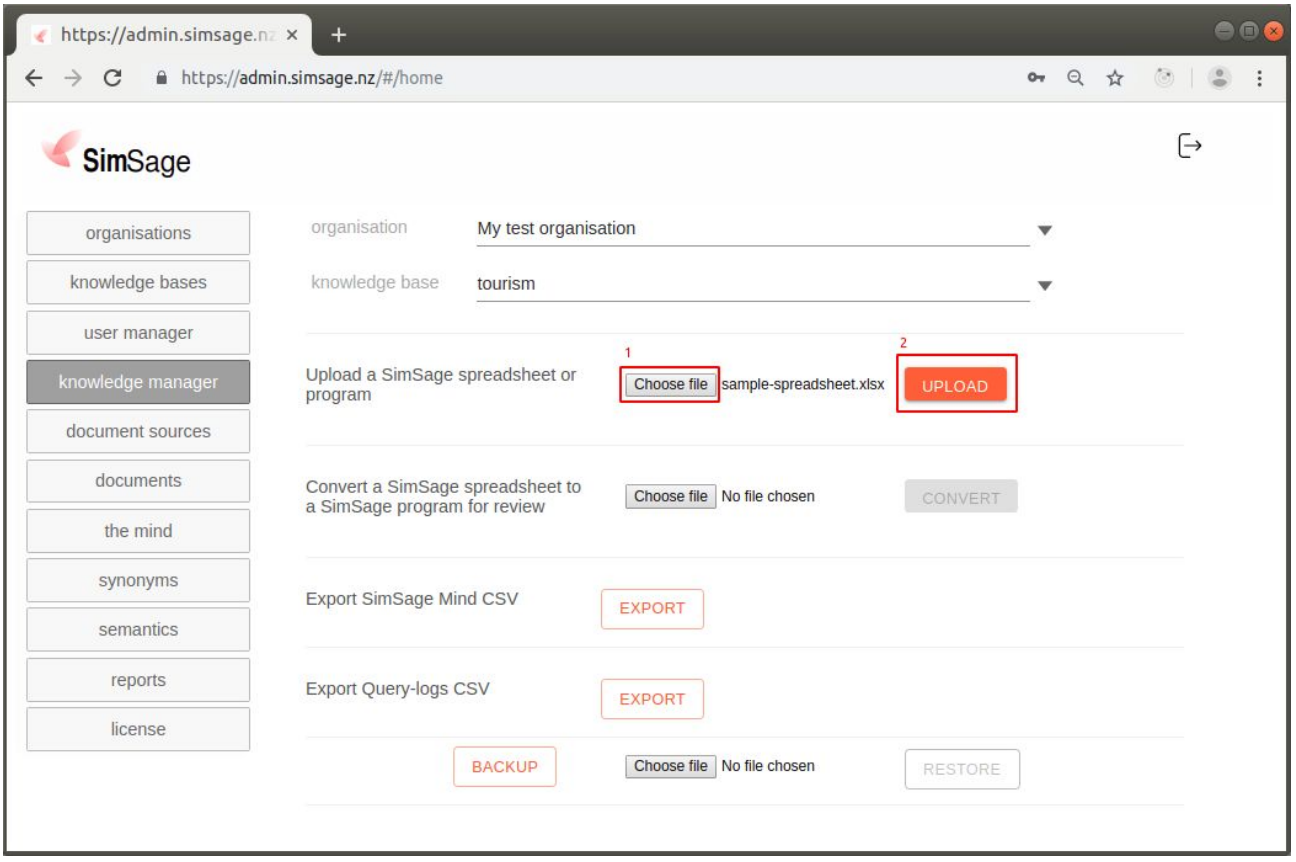


There are several actions you can perform on this screen pertaining to the selected organisation and knowledge base.

Upload a spreadsheet to teach SimSage
Upload a SimSage text-based program (see Chapter 9, Programming SimSage)
Export SimSage's taught items as an Excel Spreadsheet
Export a Spreadsheet exploring what documents SimSage has seen (Itemization)
Export SimSage's query logs (what they users have been asking it, when and what)

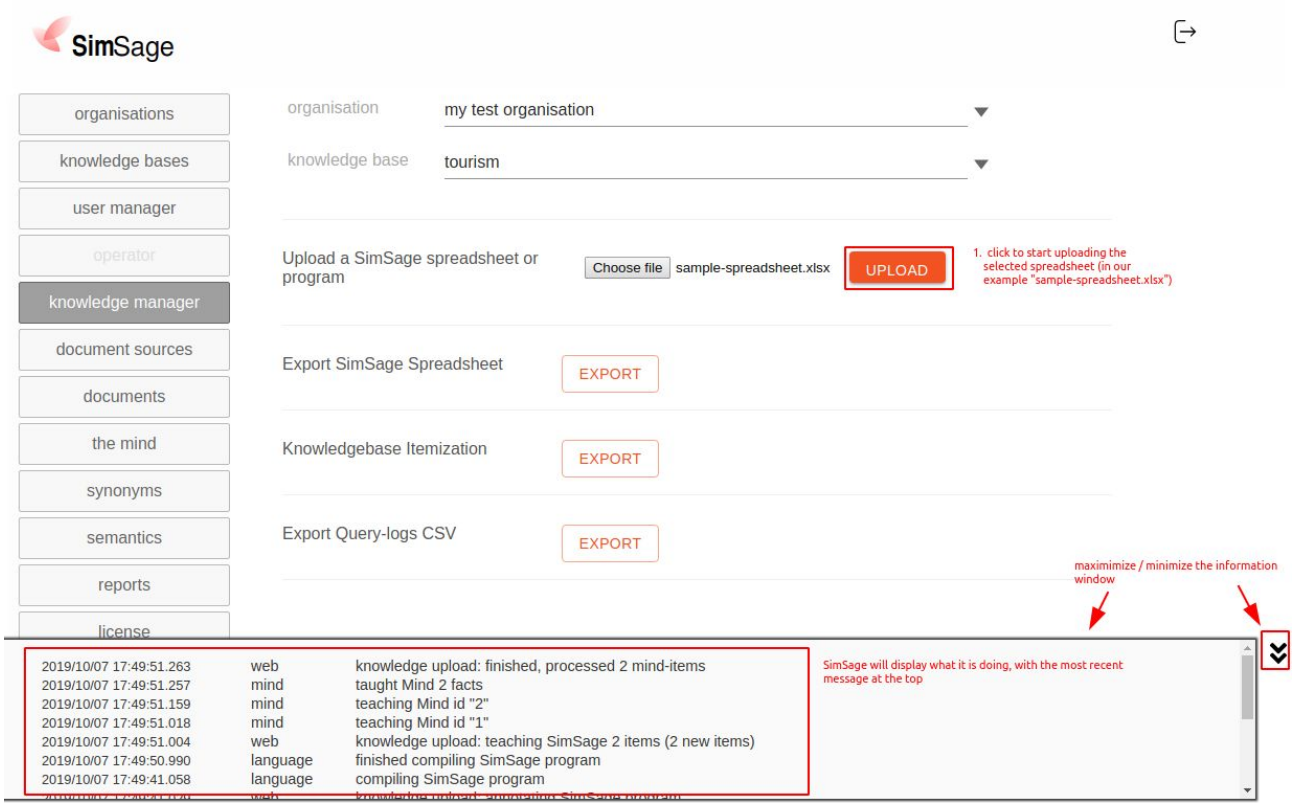


Click “upload file” of the “Upload SimSage spreadsheet or program”. A file dialog will pop-up. Select the spreadsheet you have created.



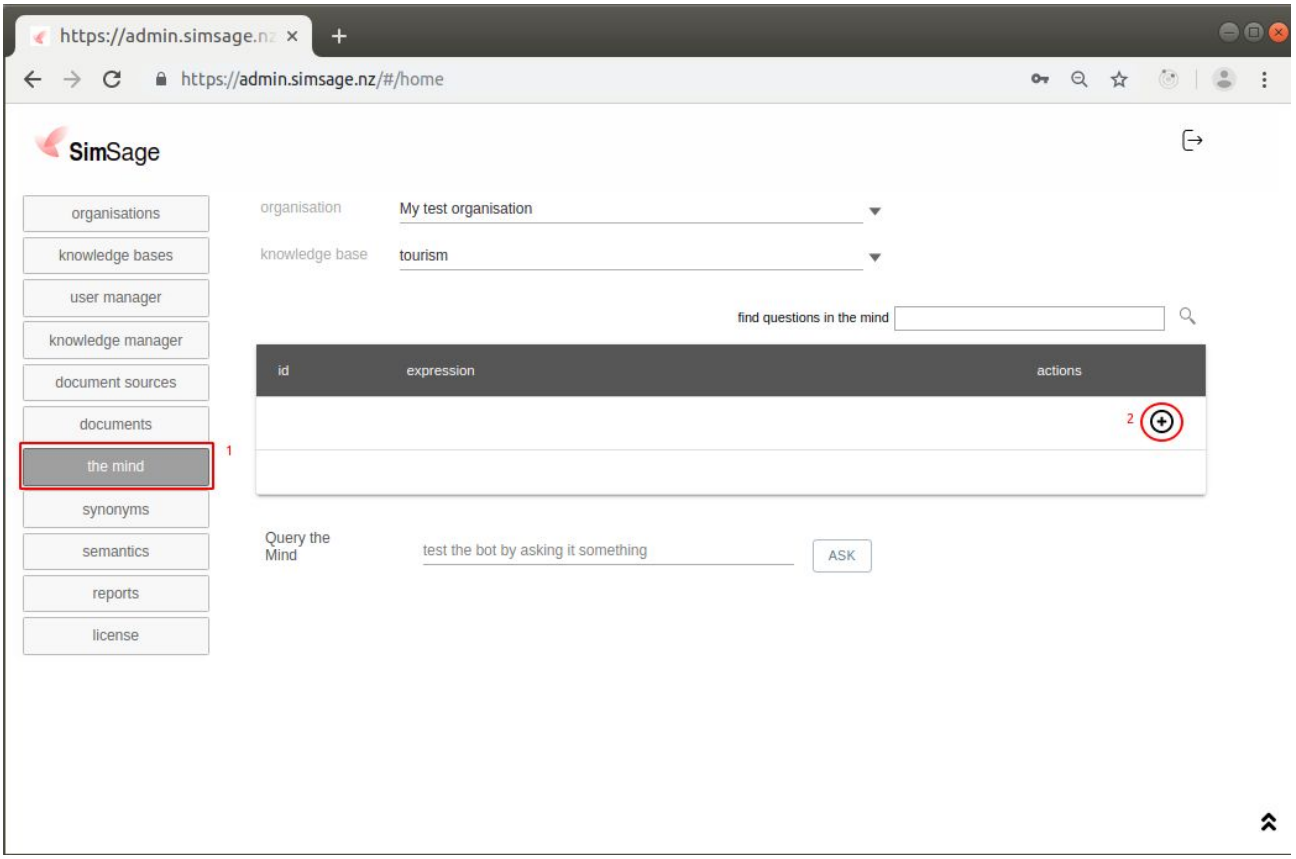
Once you have selected the spreadsheet, the UPLOAD upload button will turn a reddish orange. Click this button next.

This starts an asynchronous teaching process on the server. A yellow band at the bottom of the screen will keep you notified of its progress.

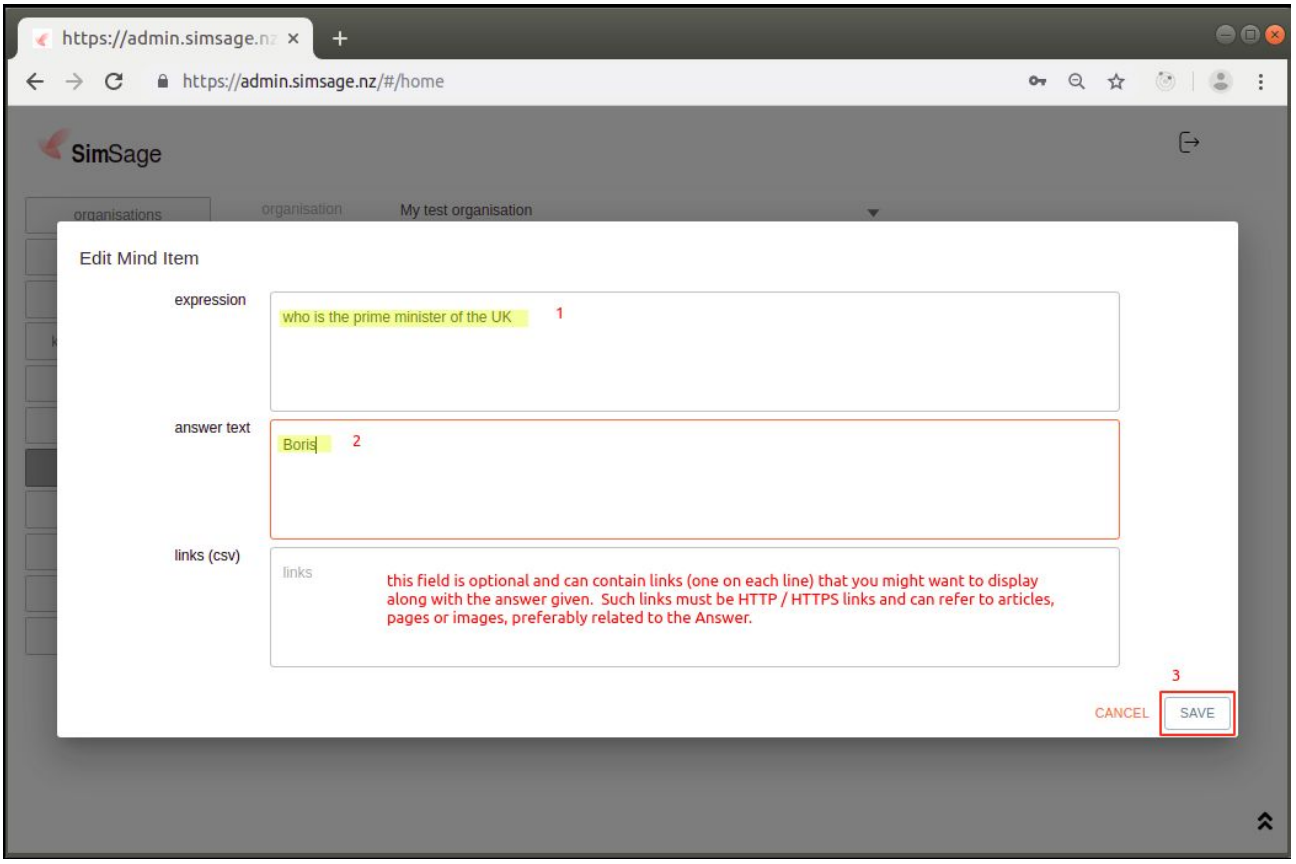


3. Manually teaching SimSage

Click the mind the mind button. This takes us to a section of the administrative interface where we can query the mind and teach new mind items as well as test queries.



Let's teach SimSage something new. Click the ⊕ plus icon to add a new "mind item".



The expression has a complex format, same as the spreadsheet. It can be as simple as a single question (we are teaching it “who is the prime minister of the UK”).

The expression can consist of many questions (assuming they represent the same answer). We too allow for variations in questions. Many questions can be separated by double bar symbols ||, e.g.

what are you || what is SimSage

However, you could in principle cut those two questions down into one by combining the words of the two questions. SimSage *understands* (processes) English. SimSage will conjugate verbs and understands that “are” and “is” are two forms of the “to be” verb.

what (are | is) (you | SimSage)

This expression expands to found possible alternatives

what are you
what are SimSage
what is you
what is SimSage

We do not need the expansion (are | is) as they are the same verb. It might read a little funny and might take some getting used to be either variation of

what are (you | SimSage)

or

what is (you | SimSage)

would be a valid substitution for your questions above.

These bracketed items can be combined with many more questions using the double bar symbols, e.g.

what is (you | SimSage) || who are you

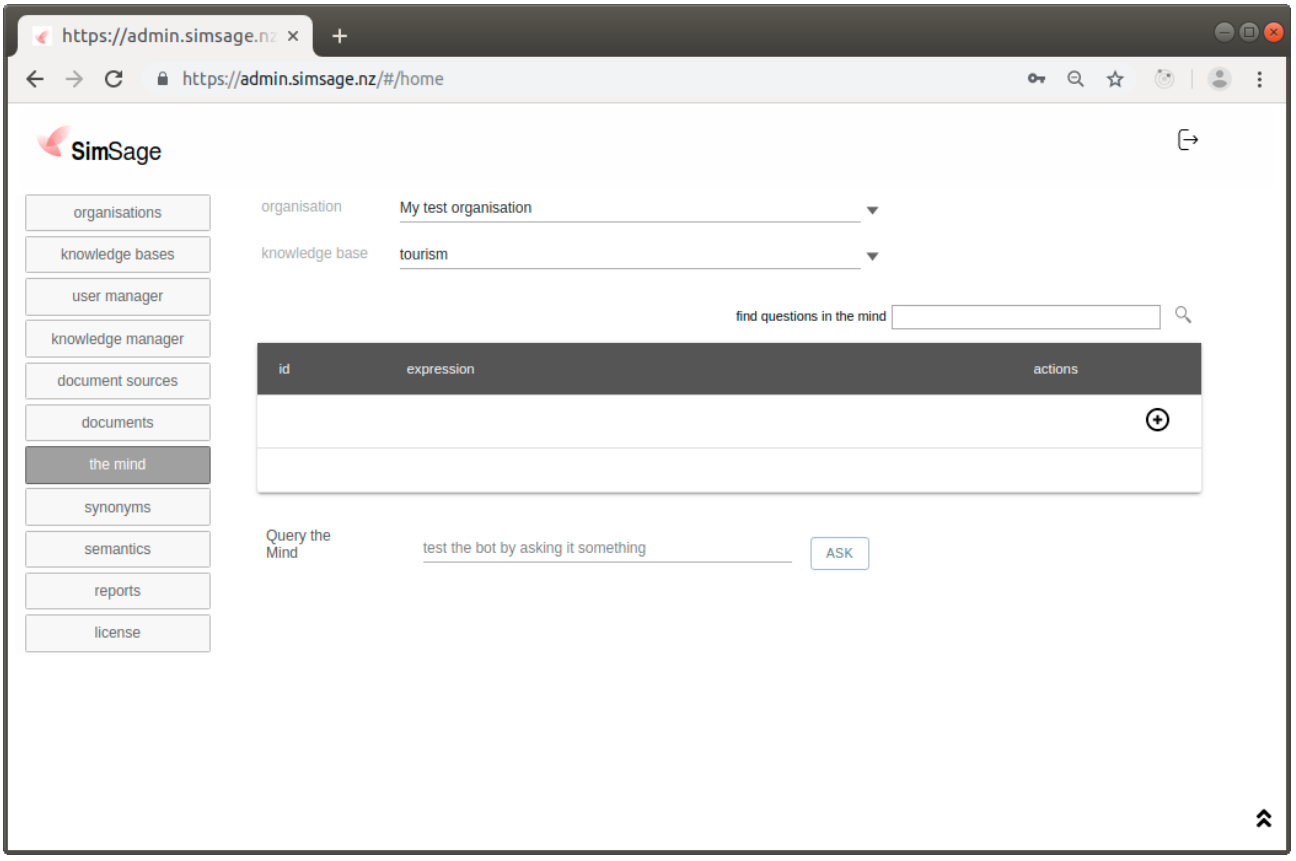
which we could simplify again by combining (what | who), etc. These same expressions can be used in your spreadsheets.

The *answer text* of the dialog box is what will be shown as the answer if this question is asked. Our simple answer is “Boris” (correct at the time of writing this document). The answer can contain multi-line answers and carriage return characters. However, keep in mind that most “bot interfaces” don’t have a whole lot of room to display a lot of text.

You can setup http / https links to external pages or images in the links *(csv)* field. Separate each link with a comma. This field is entirely optional.

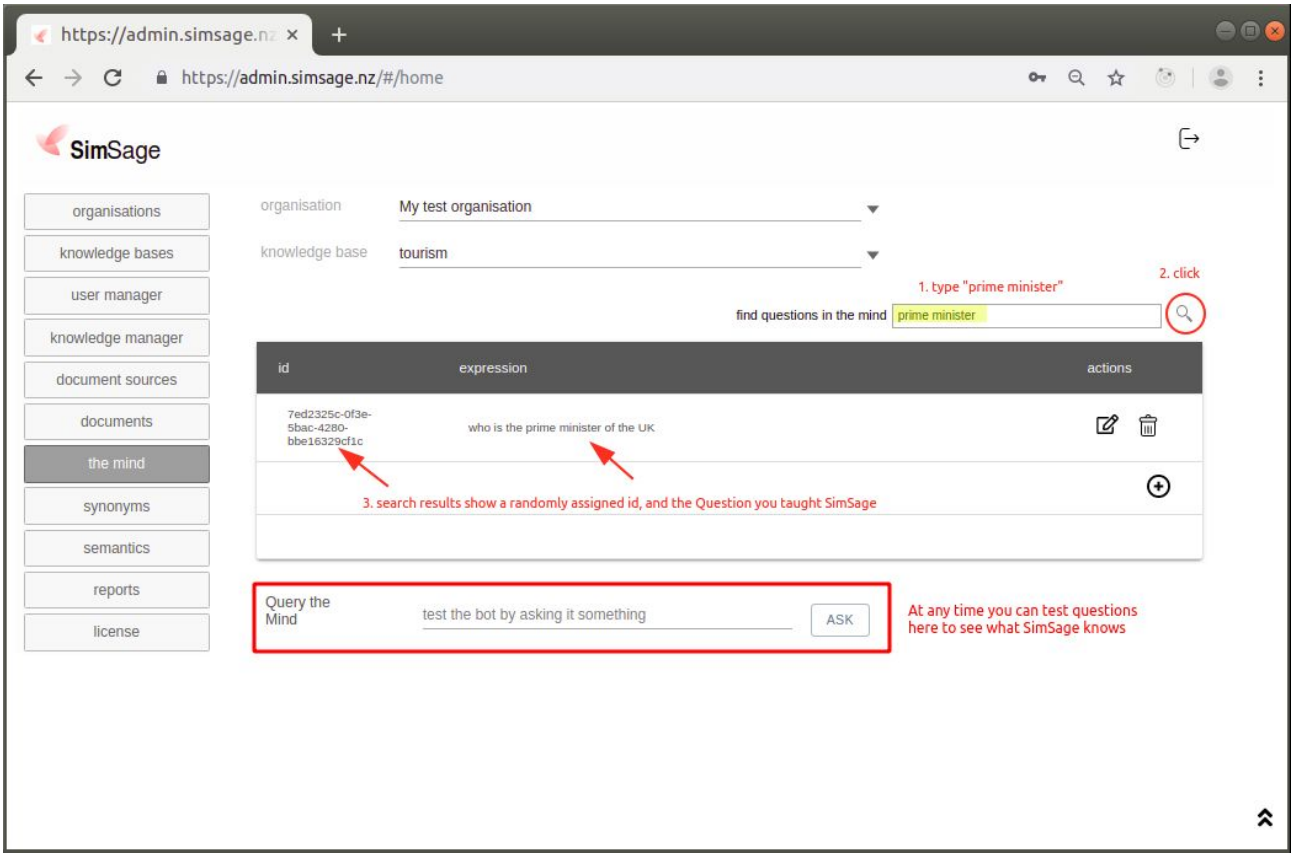
Click the

save button to close this dialog and teach SimSage this new fact. Once SimSage has finished processing this item the screen will return as it was previously.



SimSage can be taught thousands of items, showing them, all makes little sense. You can however search for items taught by using the *find questions in the mind* filter box.

Type “prime minister” in this box and press enter (or click the magnification glass).



This shows the item we just taught SimSage. The id of this item is a randomly generated GUID. We can remove it by clicking the trash icon or edit and view / change it by clicking the pencil item.

3.2.3.1 Testing SimSage Question and Answer

Let's test SimSage. Type "who is the prime minister of the UK" in the "Query the Mind" text box and click the ASK ask button.

The answer shown is "Boris". The 1.00 is the confidence / probability SimSage has of this answer being correct. A value of 1.00 means 100%. A value of 0.91 means 91%.

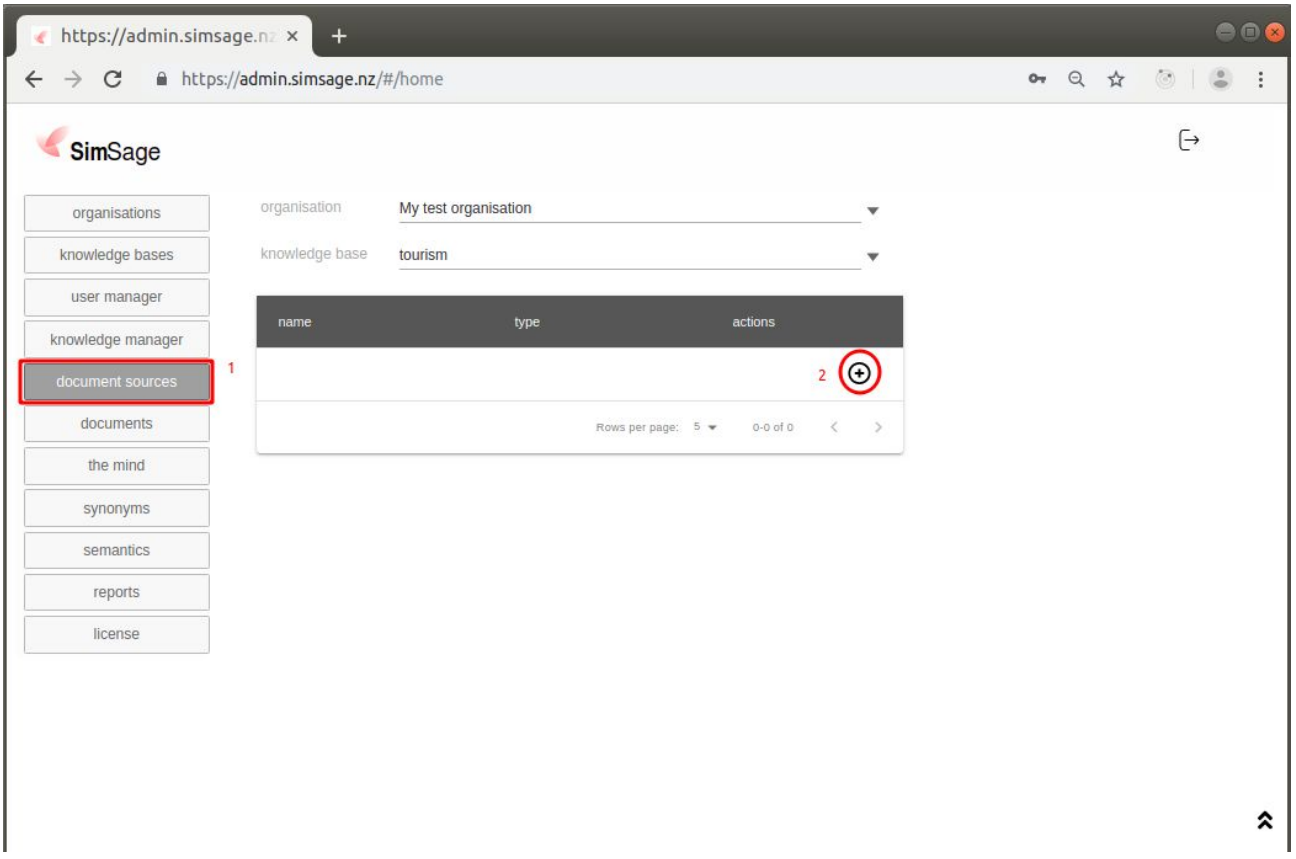
Play around and try a few variations. See what works and what doesn't. We know that the following combinations will work for the item taught above.

who is the pm of the UK
who is the pm of Great Britain
who is the pm of Britain
who is the pm of England
who is the prime minister of Great Britain
who is the prime minister of Britain
who is the prime minister of England

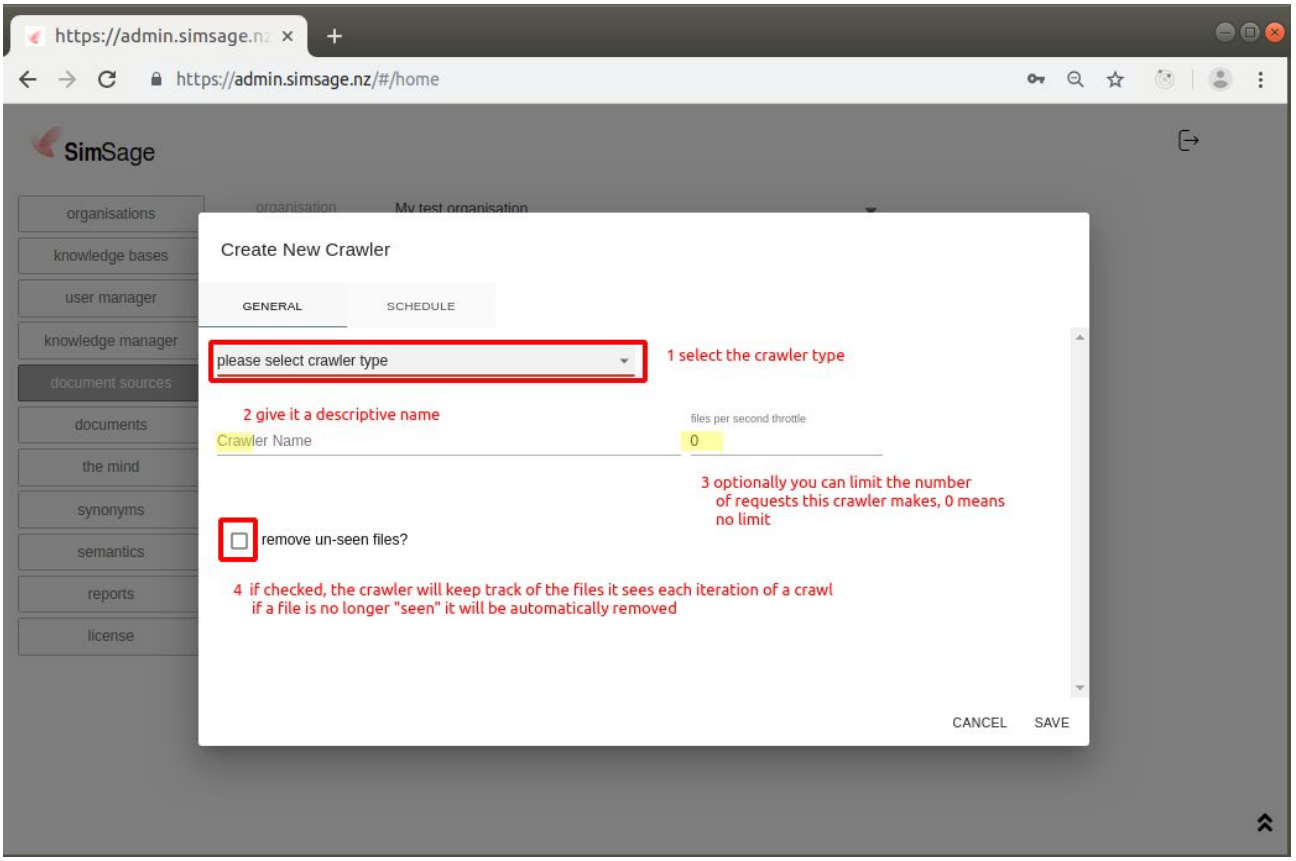
4. Setup Semantic Search

SimSage semantic search works across file-shares and websites. Click the document sources document sources button to navigate to the document sources section of SimSage. Here we can set up a crawler. A SimSage crawler is a little program part of the SimSage platform that can integrate with an external information silo, such as a Microsoft file-share or a website.

SimSage provides a powerful semantic-search engine as an additional feature for accessing information.



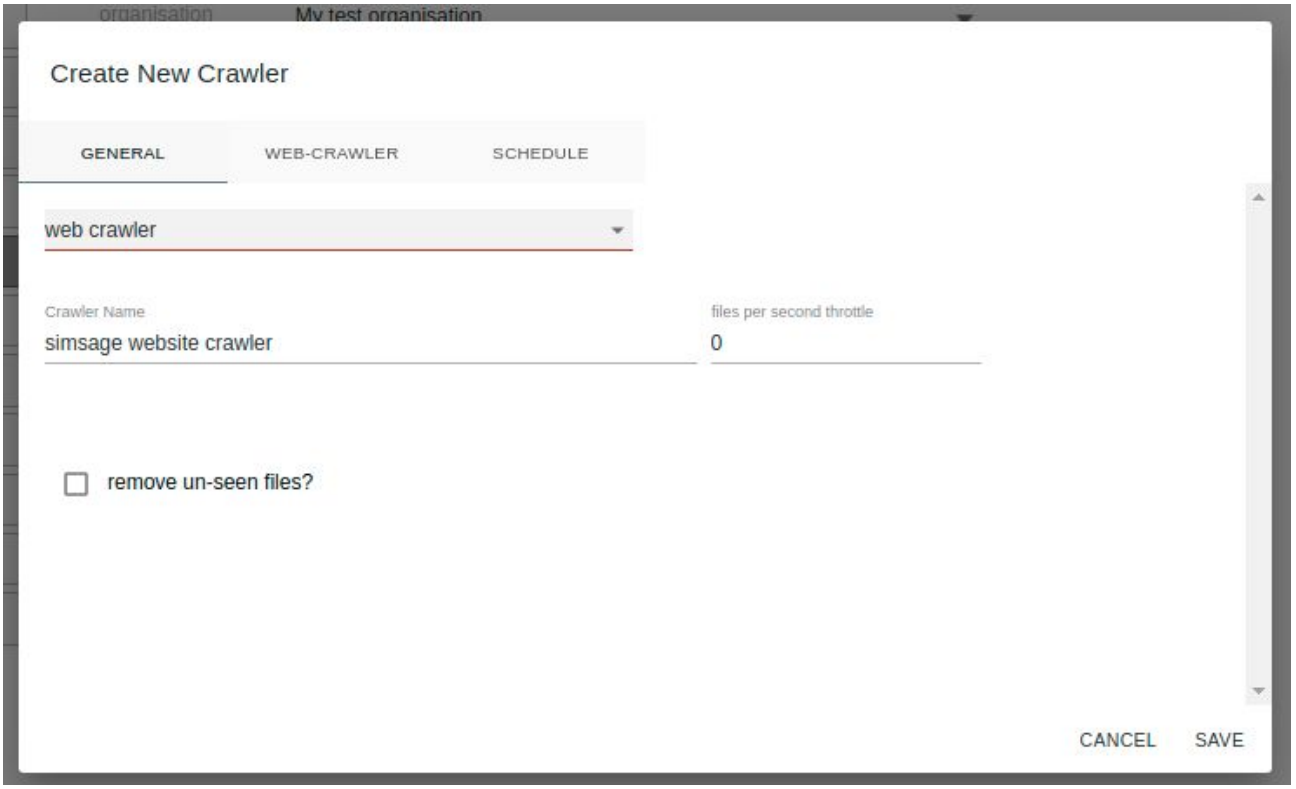
Click the ⊕ plus icon to add a new crawler for the *tourism* knowledge base of *My test organisation*.



The *Create new crawler* dialog takes you through creating a new crawler. Start by selecting the crawler type. Click the "please select crawler type" drop down box and select "web crawler". At present there are only two options in this drop-down box, "file crawler" or "web crawler". Other integration points can be added if required. Please contact SimSage if you are interested in other integration points for your customers. The moment you select either file- or web-crawler a third tab is added to this dialog for specific details regarding the selected crawler.

Crawler name is a descriptive name for this crawler, type "simsage website crawler". Files per second throttle is a speed limiter for the crawler. Leave it at 0. If we set this to for instance 10, then SimSage will throttle itself to 10 requests per second maximum.

The remove unseen files checkbox is for removing content that is no longer present. This facilitates a comparison of files previously seen by SimSage but no longer visible in subsequent crawls through data. Leave this box unchecked.

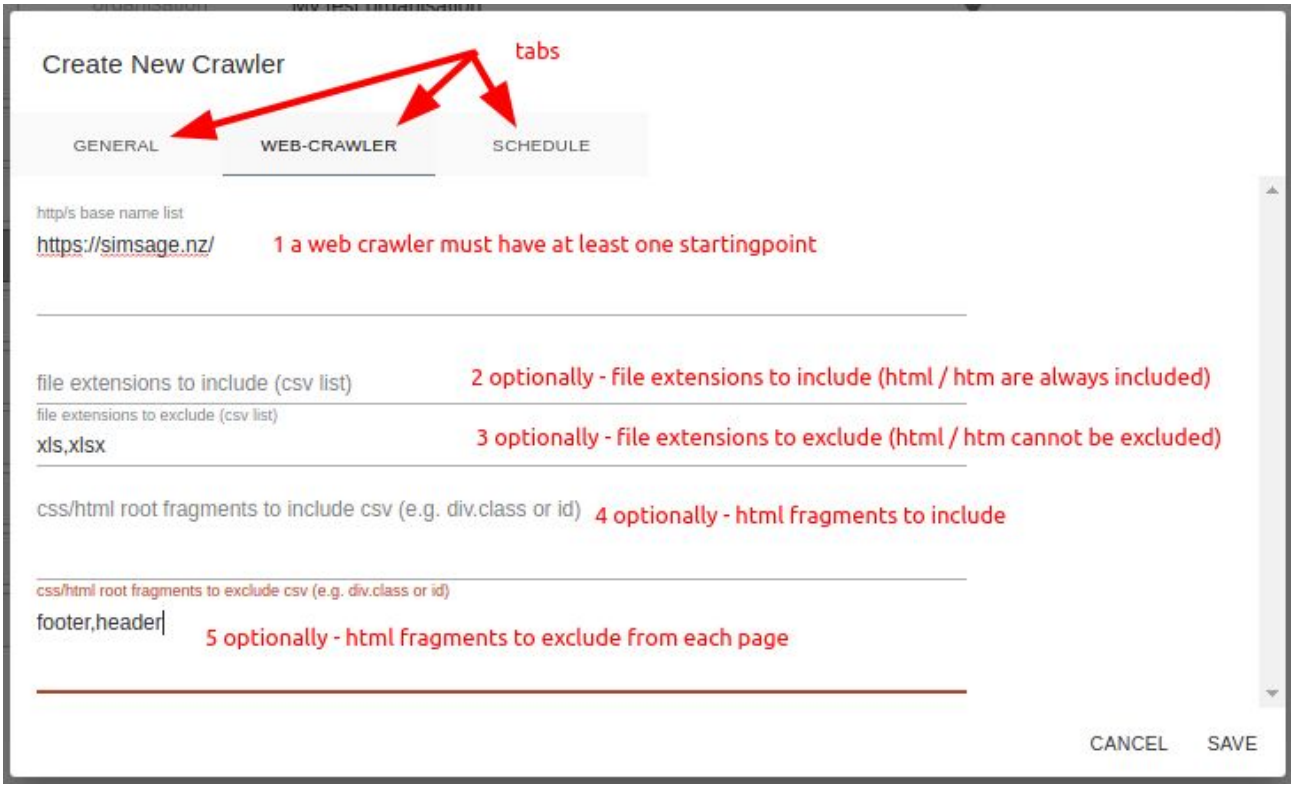


Click the web-crawler tab to proceed.

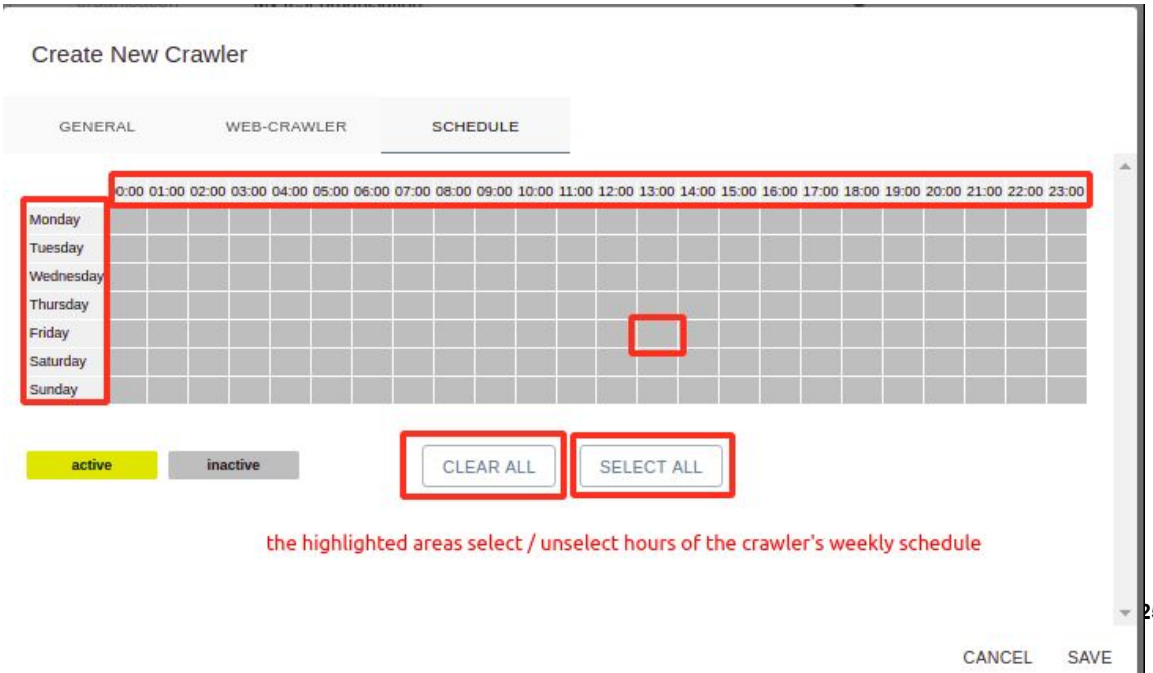
The only mandatory field here is the “http/s base name list”. We can supply a series of http / https addresses to crawl, each one on a new line.

File extensions to include / exclude are comma separated file extensions (minus the “.”) that are either to be included or excluded (do not supply values for both, pick one of the two). We opt to exclude Excel spreadsheets in our example.

CSS/html root fragments to include / exclude are for filtering parts of the web pages visited. Most web pages include menus and footers. We do not want to crawl this information as it muddles / muddies our search indexes. Investigate the websites you are crawling and see if you can figure out what elements you can do without or want to include only. Again, supply values for either include or exclude but not both.



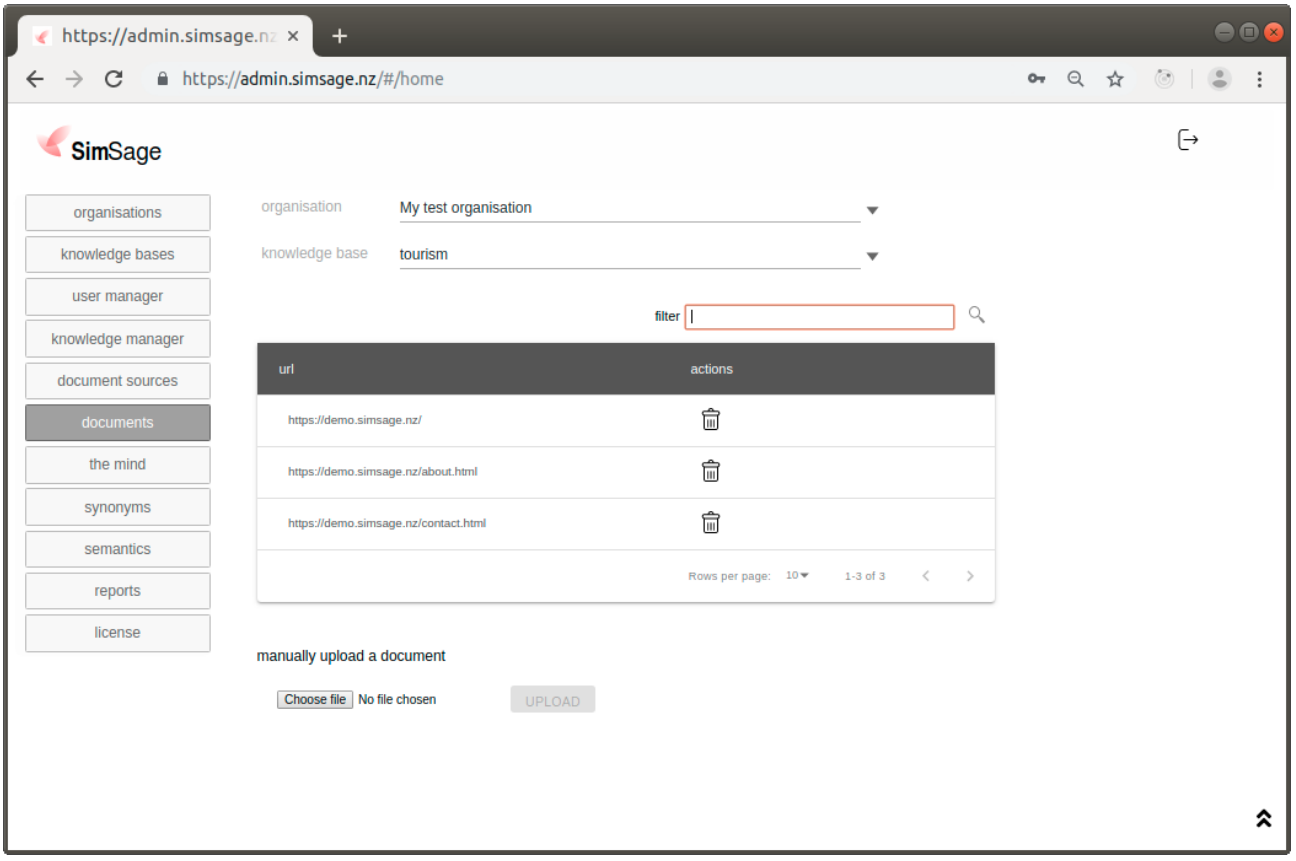
Click the schedule tab. The schedule tab pertains to the weekly crawling schedule. Here you can limit the times when the crawler looks at external sources. It could be that your customers have times when they don’t want you looking at their data (backup regimes etc.)



By default, all the times are off and the crawler will not do any work. Click the SELECT ALL select all button to keep the crawler active at all times.

Click the SAVE save button to start your crawler.

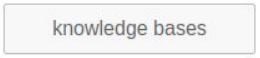
Wait a few minutes and click the documents documents button. You should start seeing the results of the crawler fetching information.



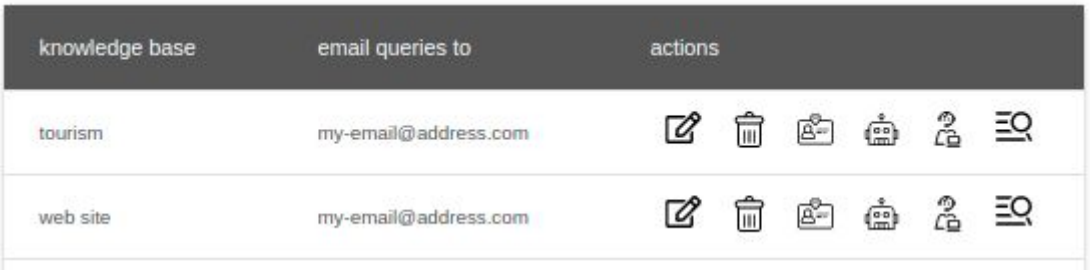
3.3.4.1 Downloading the bot interface

We now have a SimSage knowledge base loaded with a few questions and semantic search information. The admin interface is part of the SimSage platform. However, bot and search interfaces are for you to set up. We provide a download facility for skeleton implementations as examples. Please use these as you see fit.

You can also find the source code for these items on our GitHub page at https://github.com/simsage-nz

Click the 

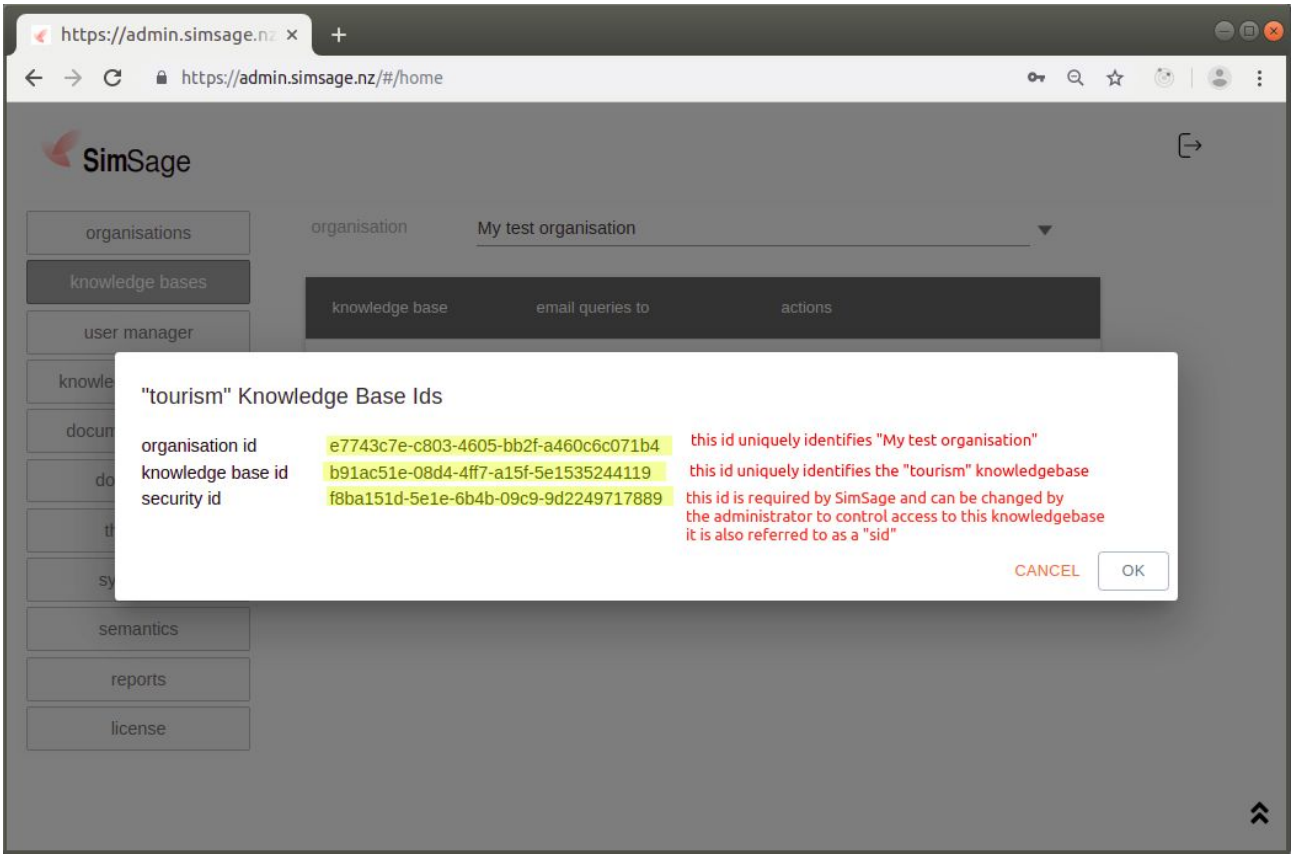
knowledge bases button. We now focus on the detailed icons provided with each knowledge base.



The icons are

Icon	Description
	Edit this existing knowledge base, change its name or security-id
	Remove this knowledge base and all its associated information.
	View the ids of this organisation, knowledge base and security-id
	Download the sample bot HTML for this organisation / knowledge base with the correct ids set
	Download the sample operator / helpdesk HTML for this organisation / knowledge base with the correct ids set
	Download the sample search interface HTML for this organisation / knowledge base with the correct ids set

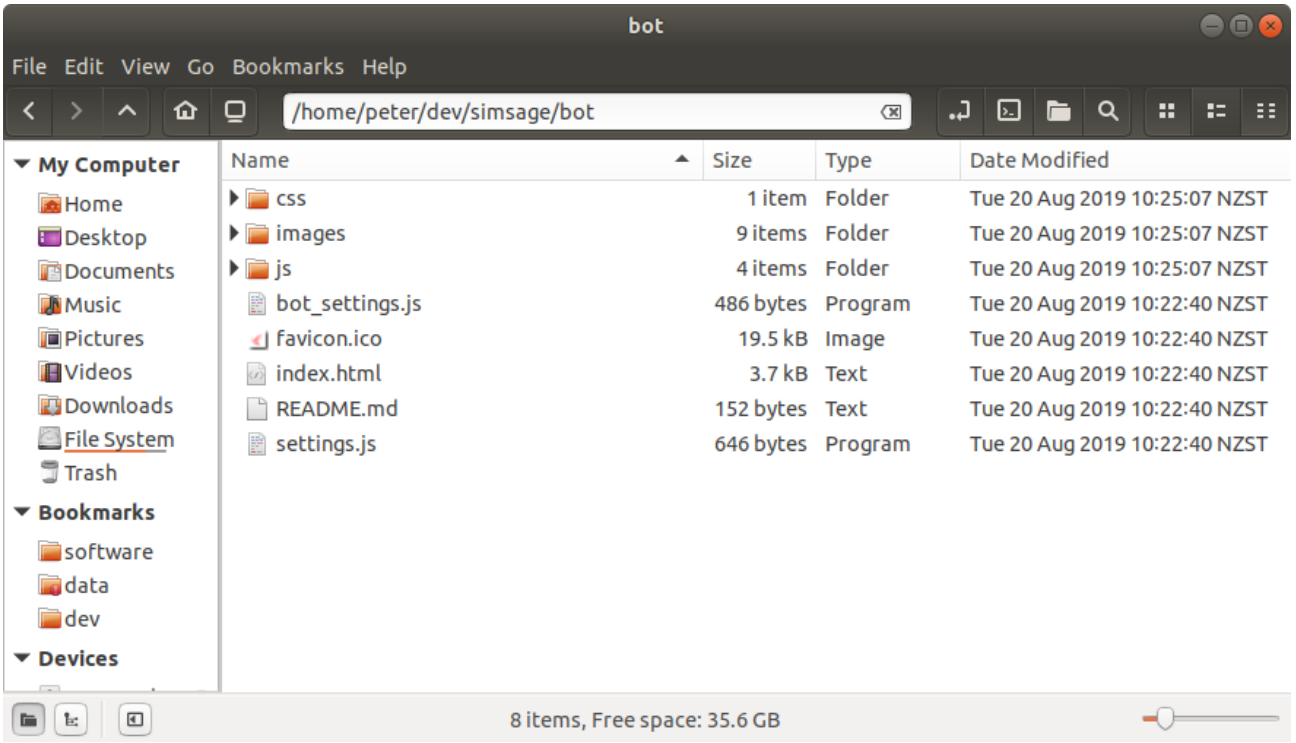
Our example has been working with the *tourism* knowledge base. View its ids by clicking its ID icon.



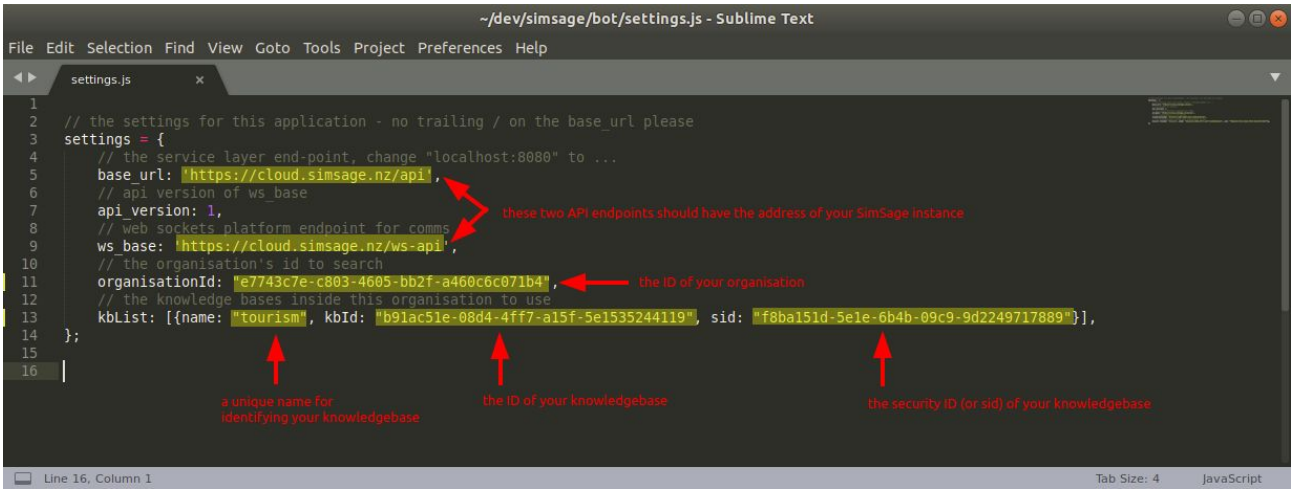
The id numbers shown in this document will be different from your own id numbers. These id numbers are randomly generated for each organisation, knowledge base. They are fixed for the rest of the lifespan of the organisation / knowledge base. Only the security-id can be changed.
These numbers are required for SimSage to determine what information sources you want your bot / search interface to query. You can copy them from this interface. Better yet, you can download our sample HTML with those id numbers set.

Click the download bot HTML icon for the *tourism* knowledge base. This will download the HTML to your browser. The experience of this download varies between browsers. Chrome shows the downloaded file at the bottom of the page like so.

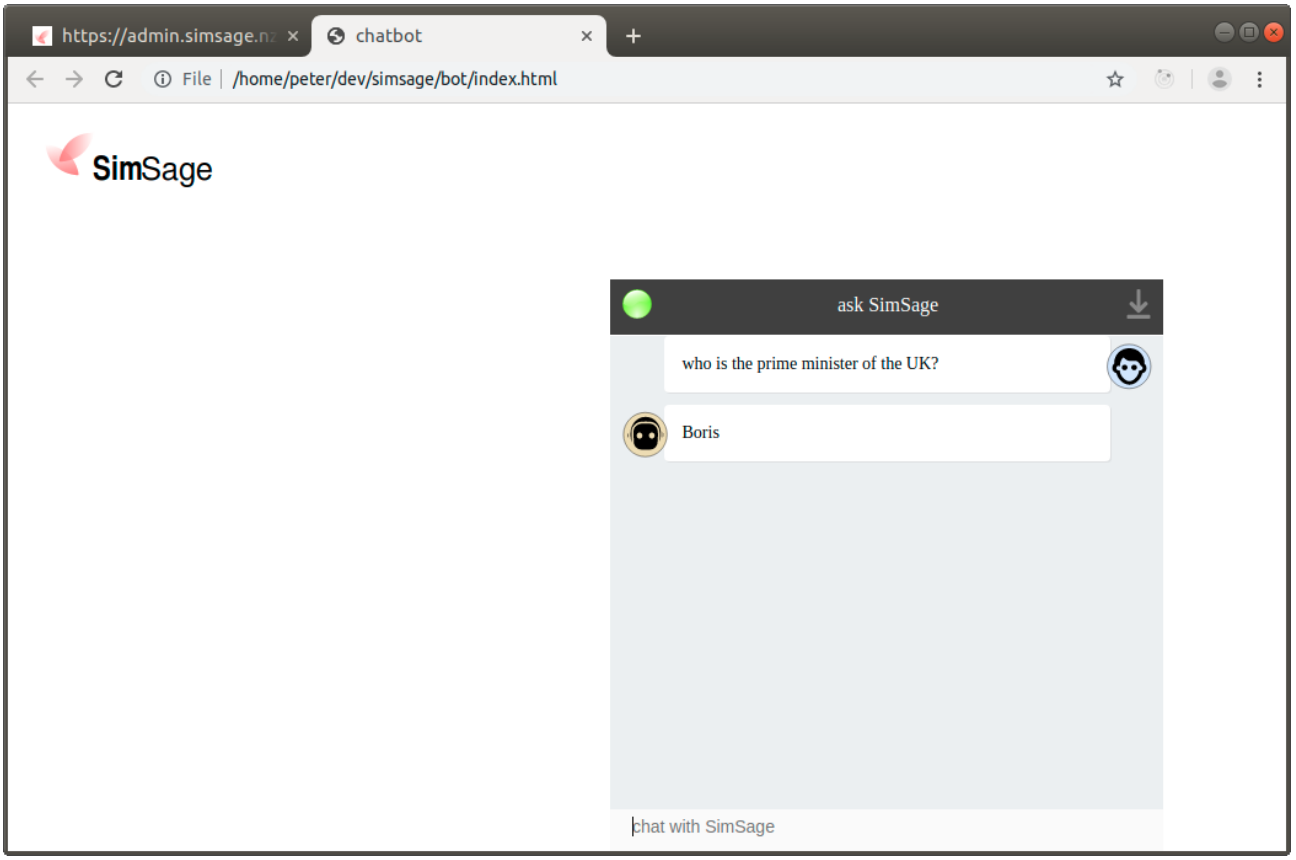
Unzip the file and inspect its contents.



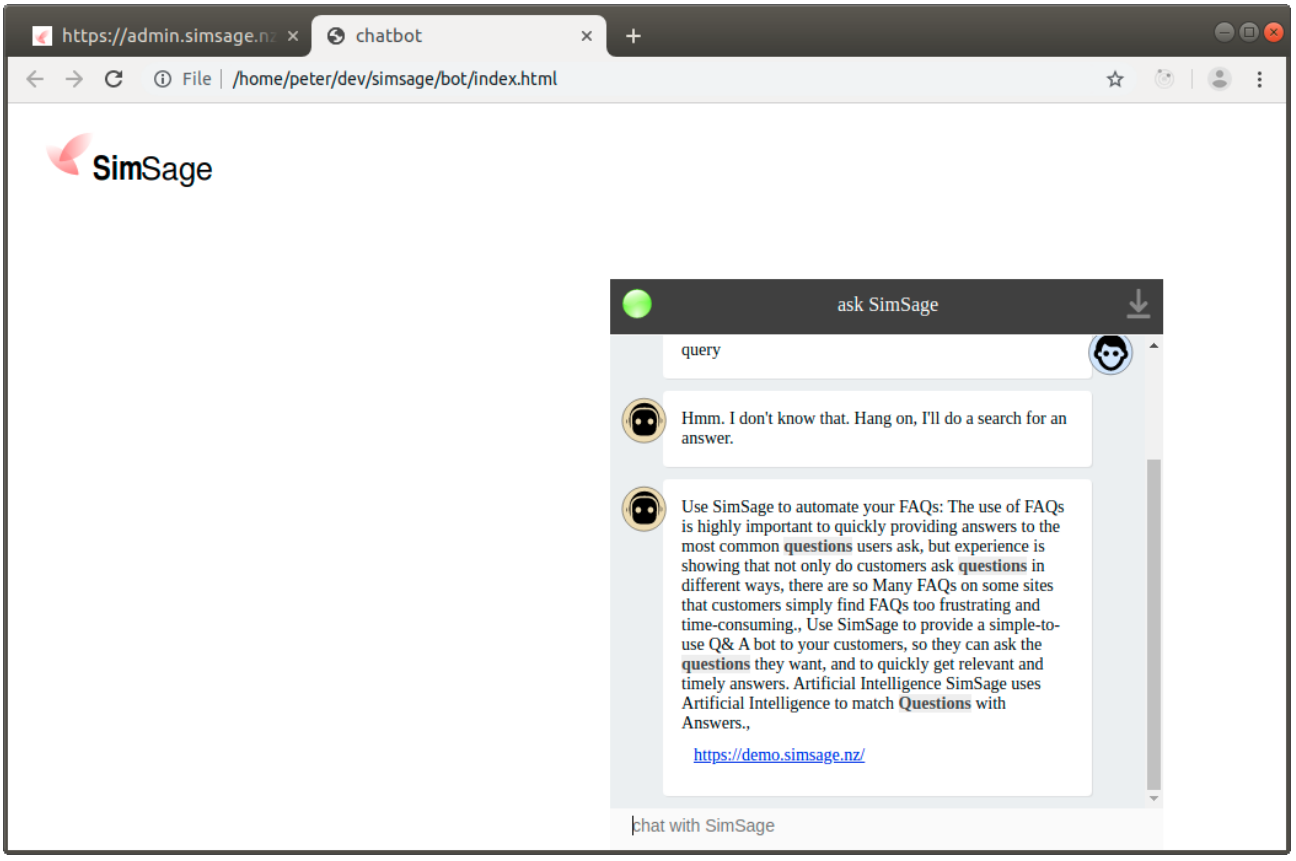
Open the settings.js file with a text editor and check the ids correspond to the ids viewed above.



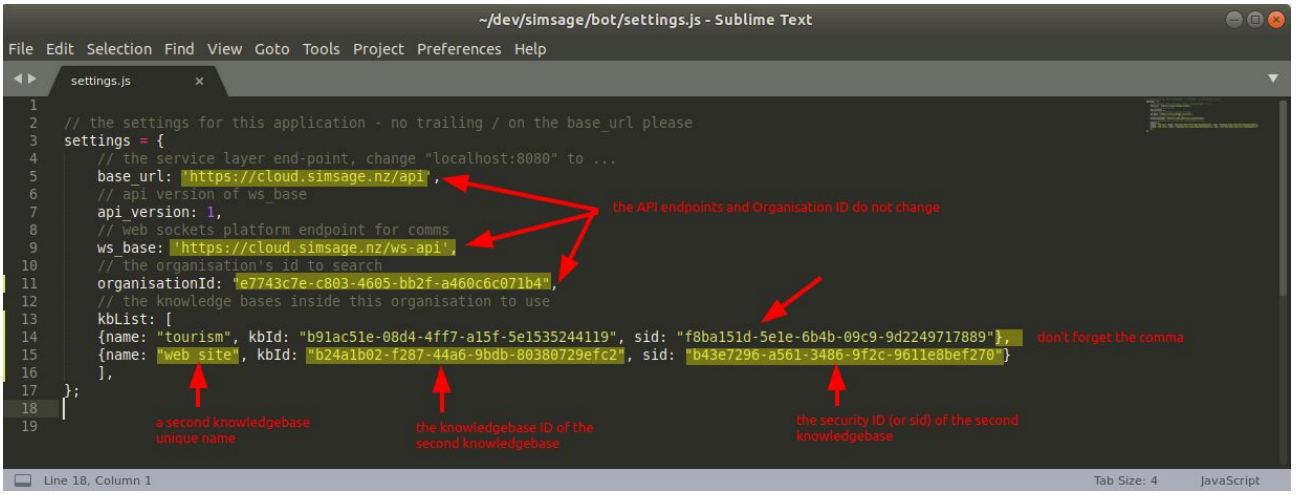
Double click the index.html file to open it in your favourite browser and query SimSage.



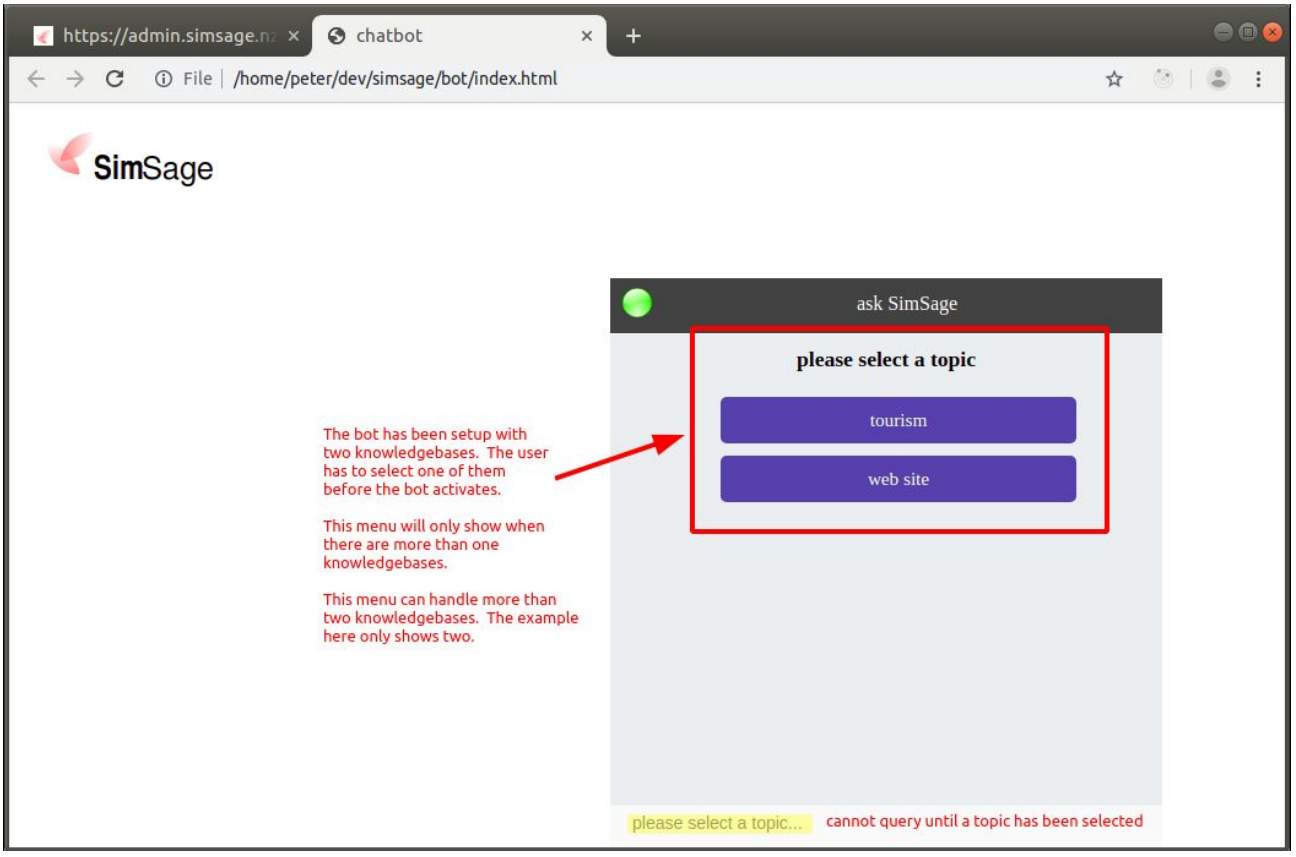
The bot UI is very powerful. It will fall back to semantic search if available for instance.



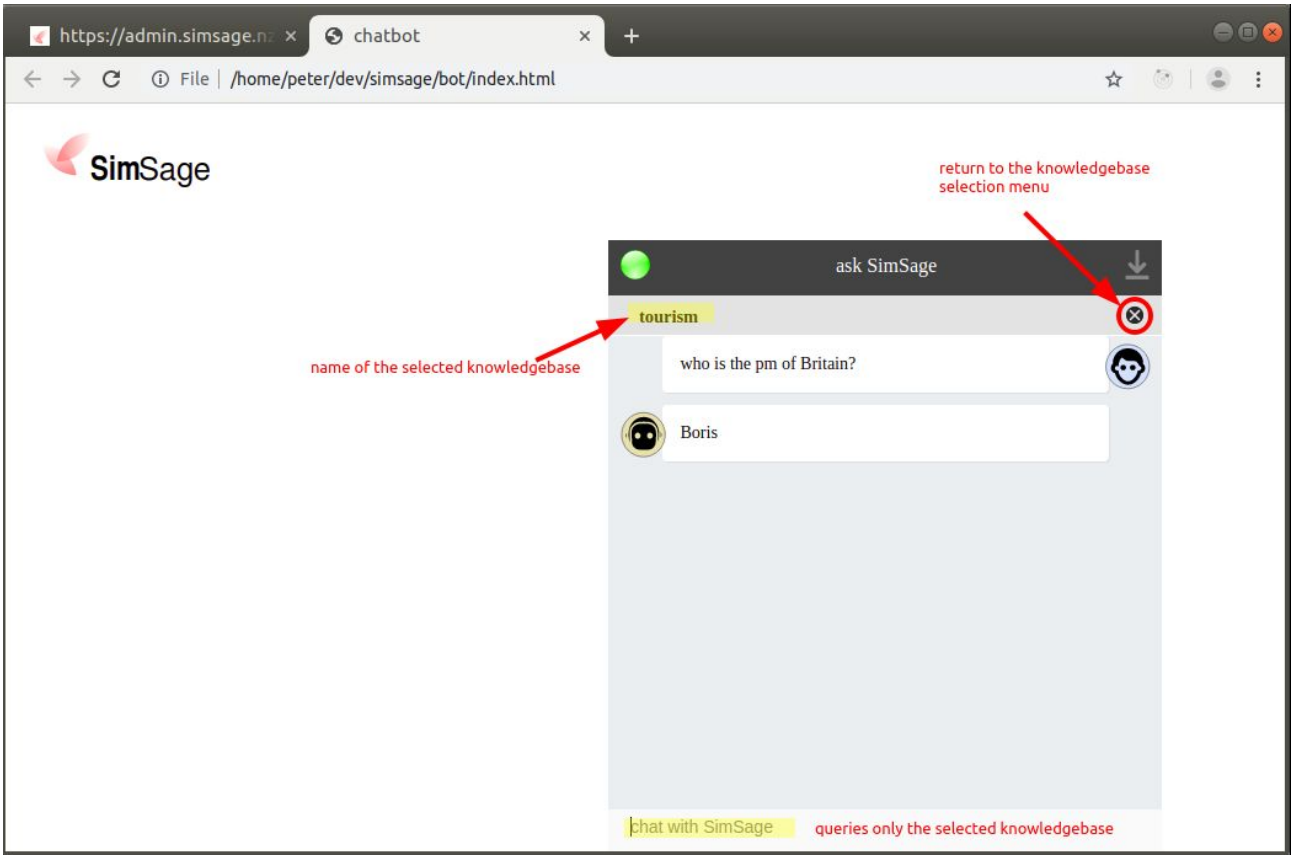
You can also add your other knowledge bases for the SAME organisation to the settings.js file as shown below (your Ids will differ!). Pay careful attention to the syntax of this JSON file before modifying it.



Then when you refresh the bot HTML in your browser, you will be presented with a menu where the user can switch between different knowledge bases to interact with your SimSage platform.



Selecting a topic then selects a specific knowledge base. Click the button will take the user back to the selection menu shown above.



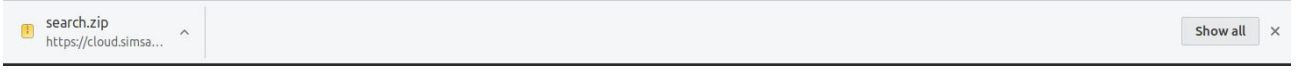
Note that this menu and selection mechanism will only show if there are more than one knowledge bases in the settings.js file.

4.2 Downloading the search interface

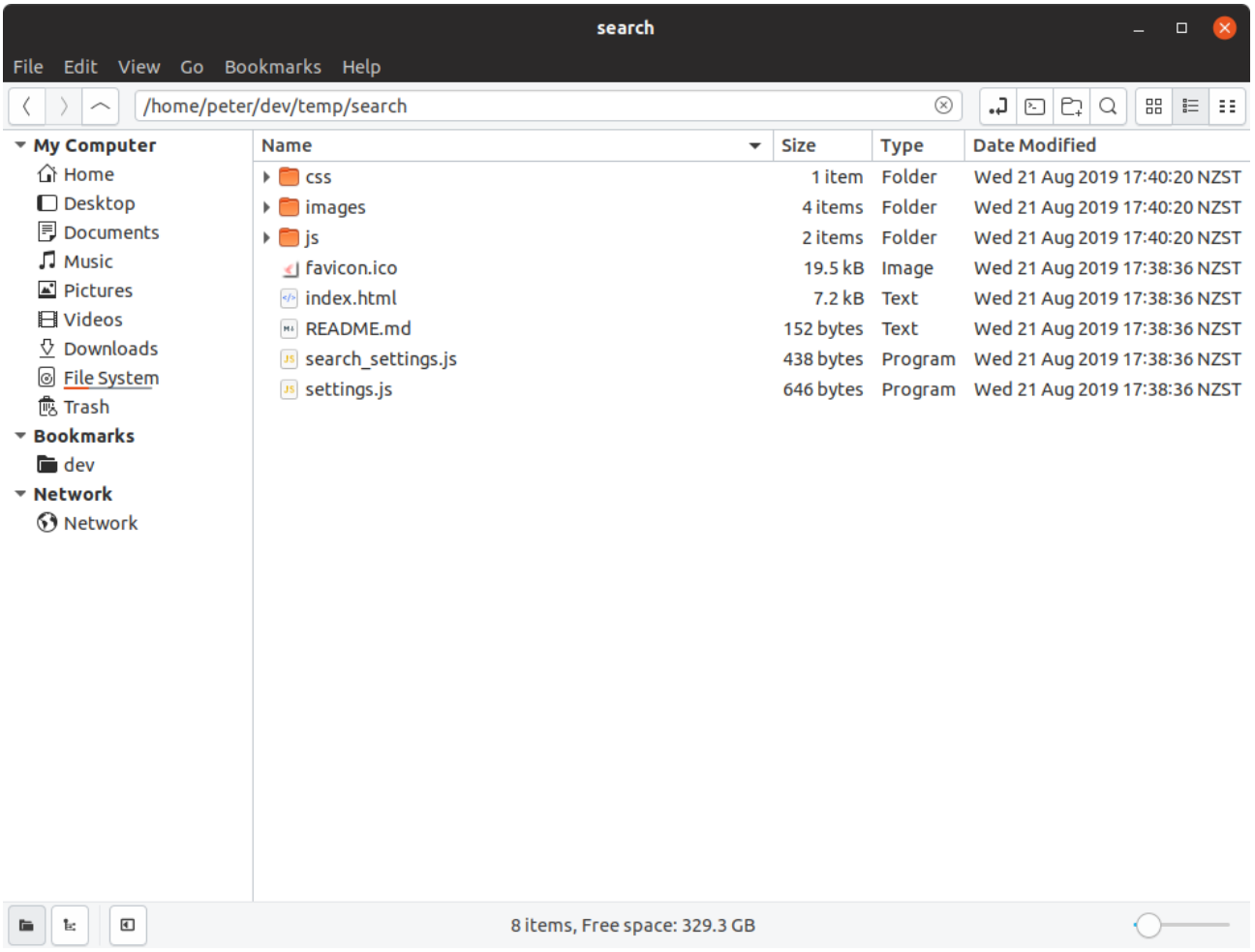
We now have a SimSage knowledge base loaded with a few questions and semantic search information. The admin interface is part of the SimSage platform. However, bot and search interfaces are for you to set up. We provide a download facility for skeleton implementations as examples. Please use these as you see fit.

You can also find the source code for these items on our GitHub page at https://github.com/simsage-nz

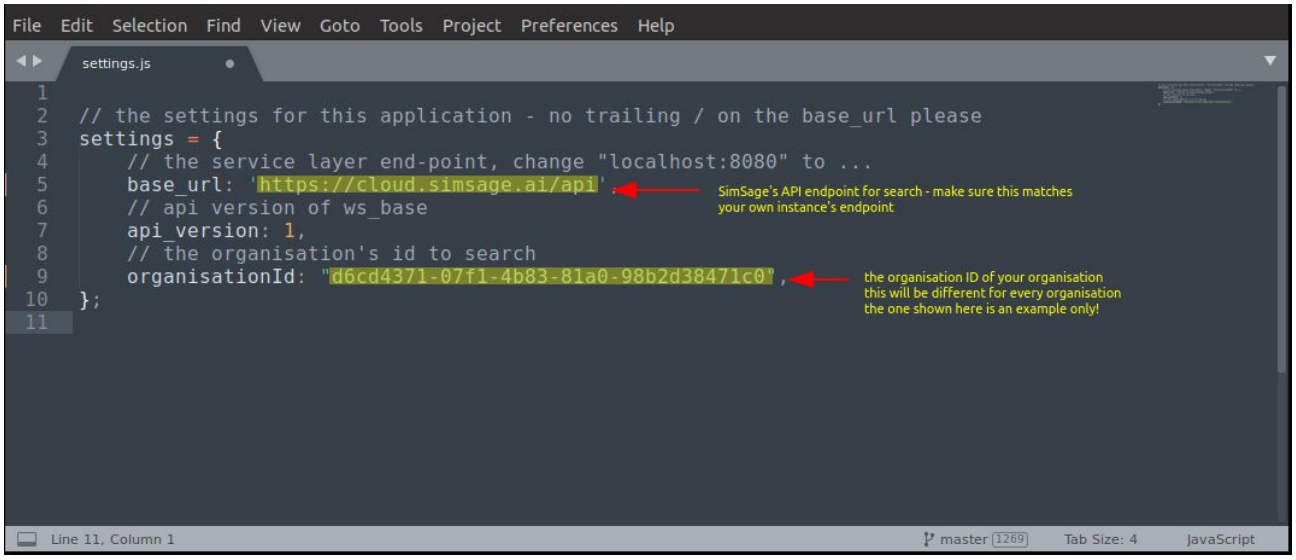
Click the download search HTML icon for the *tourism* knowledge base. This will download the HTML to your browser. The experience of this download varies between browsers. Chrome shows the downloaded file at the bottom of the page like so.



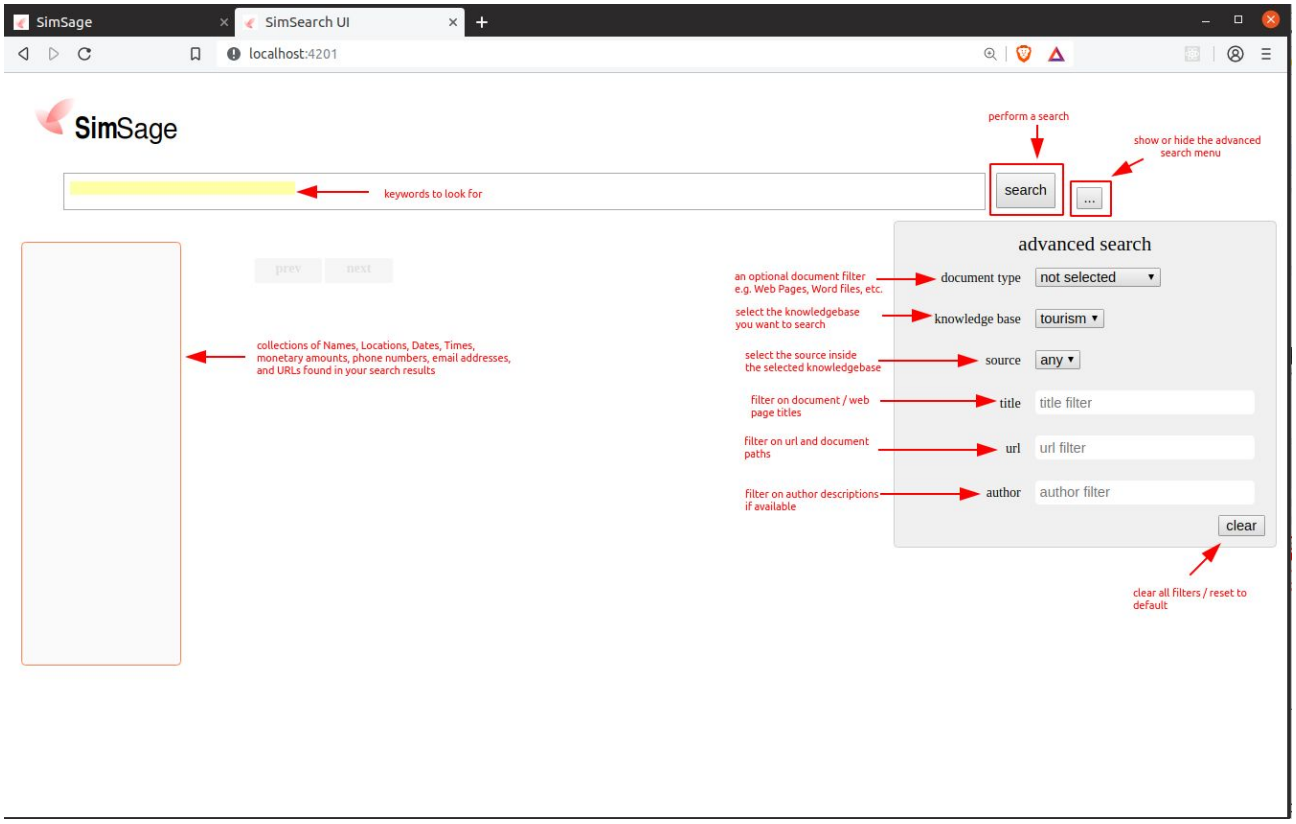
Unzip the file and inspect its contents.



Open the settings.js file with a text editor and check the ids correspond to the ids viewed above.



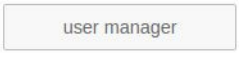
Double click the index.html file to open it in your favourite browser and query SimSage. Note that you will need an actual web-server (e.g. node's http-server) to use the search interface.



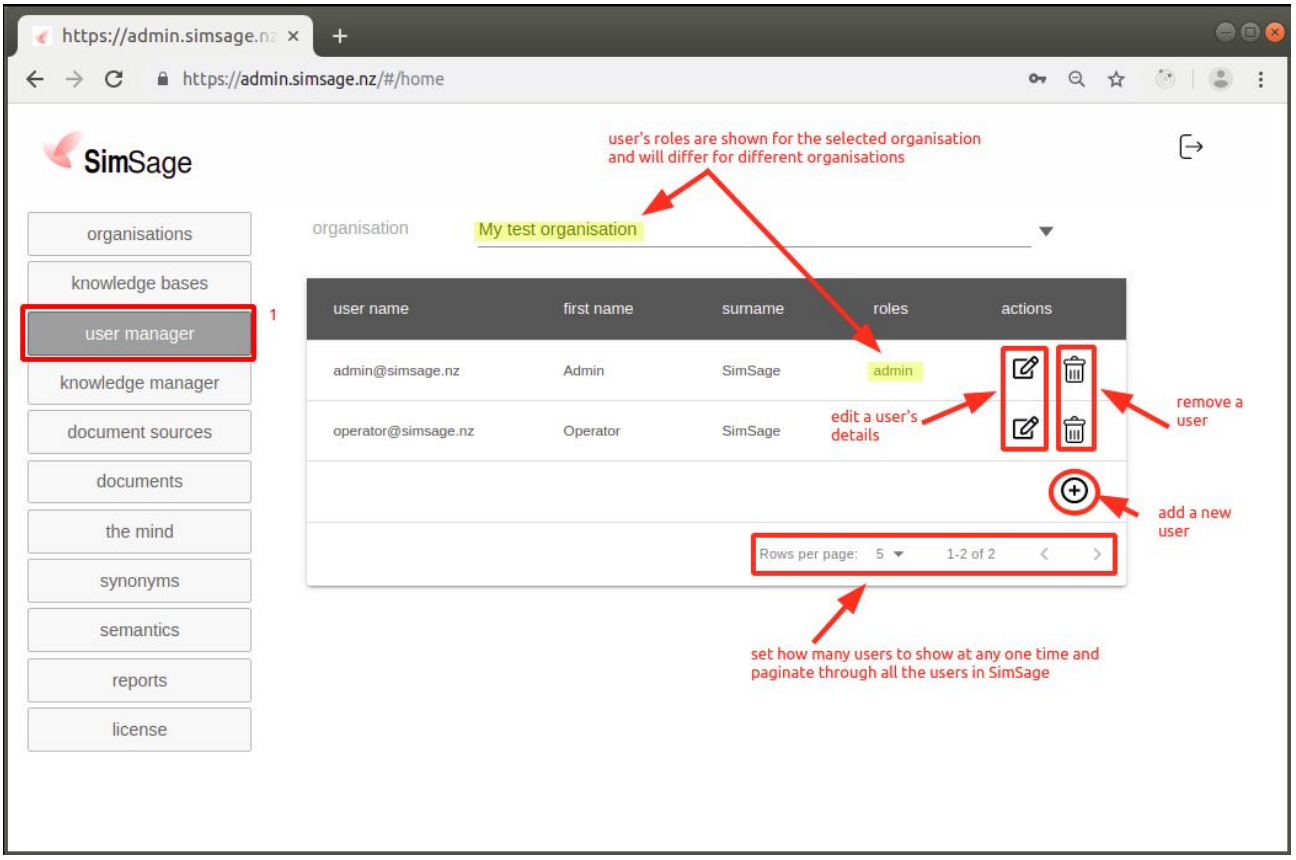
5. Users and Roles

SimSage provides several different kinds of roles for accessing the platform. These roles are:

Role name	Description
Admin	SimSage system administrator. Super user of the platform. Can login to the administrative interface and add users, knowledge bases and organisations.
Operator	SimSage operator. This role can sign-in to the operator / help desk interface. Operators must be assigned knowledge bases through the admin interface by an administrator.
Crawler	Reserved for external SimSage platform crawlers. Not used at present.
Manager	SimSage system manager. Much like an administrator. However, a manager cannot add new organisations nor modify existing users.
Reporter	Reserved user for generating reports. Not used at present.

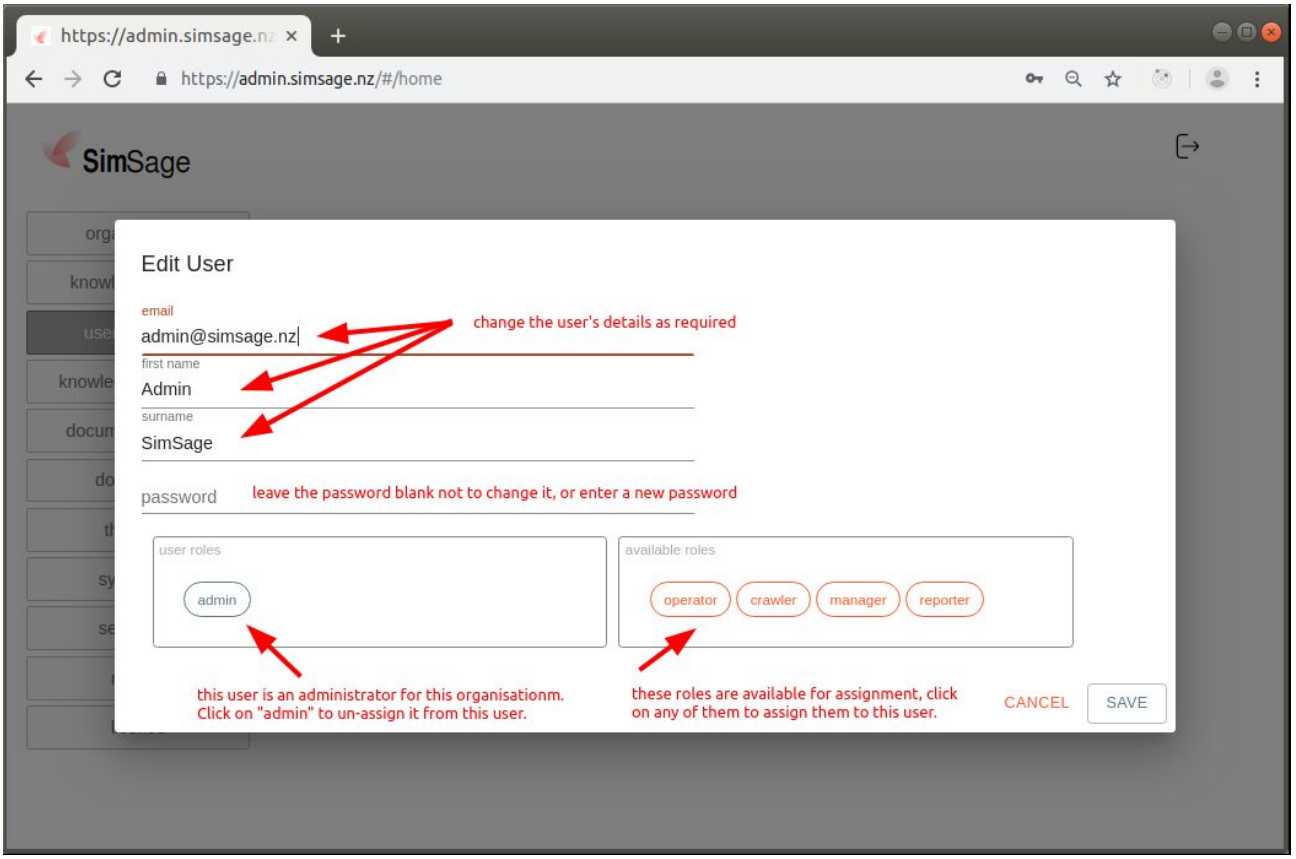
Click on  the user manager button in the administrative interface.

This interface will always show all users ordered alphabetically with their roles for the selected organisation only. Change organisation to see what roles a user has in that organisation.



The screenshot above shows that admin@simsage.nz has the “admin” role in My test organisation. It also shows another user operator@simsage.nz that holds no roles in this organisation.

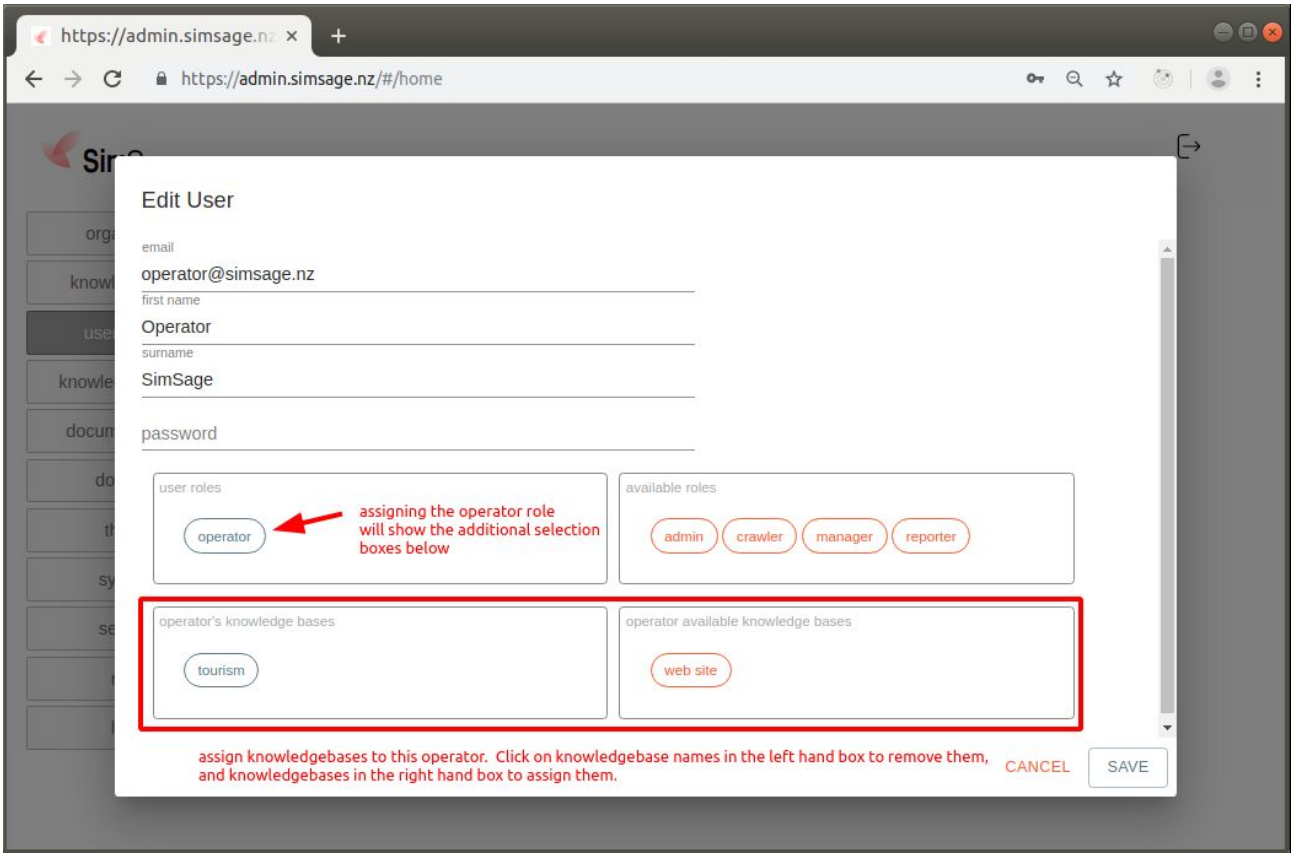
Click the edit icon to edit this user’s details. This includes changing their roles, email / login, first-name, surname and password.



This screenshot shows the admin@simsage.nz being edited. The available roles are to the right and the assigned roles to the left at the bottom of the dialog. Clicking any of these “button” roles moves them to the desired assigned / not assigned positions.

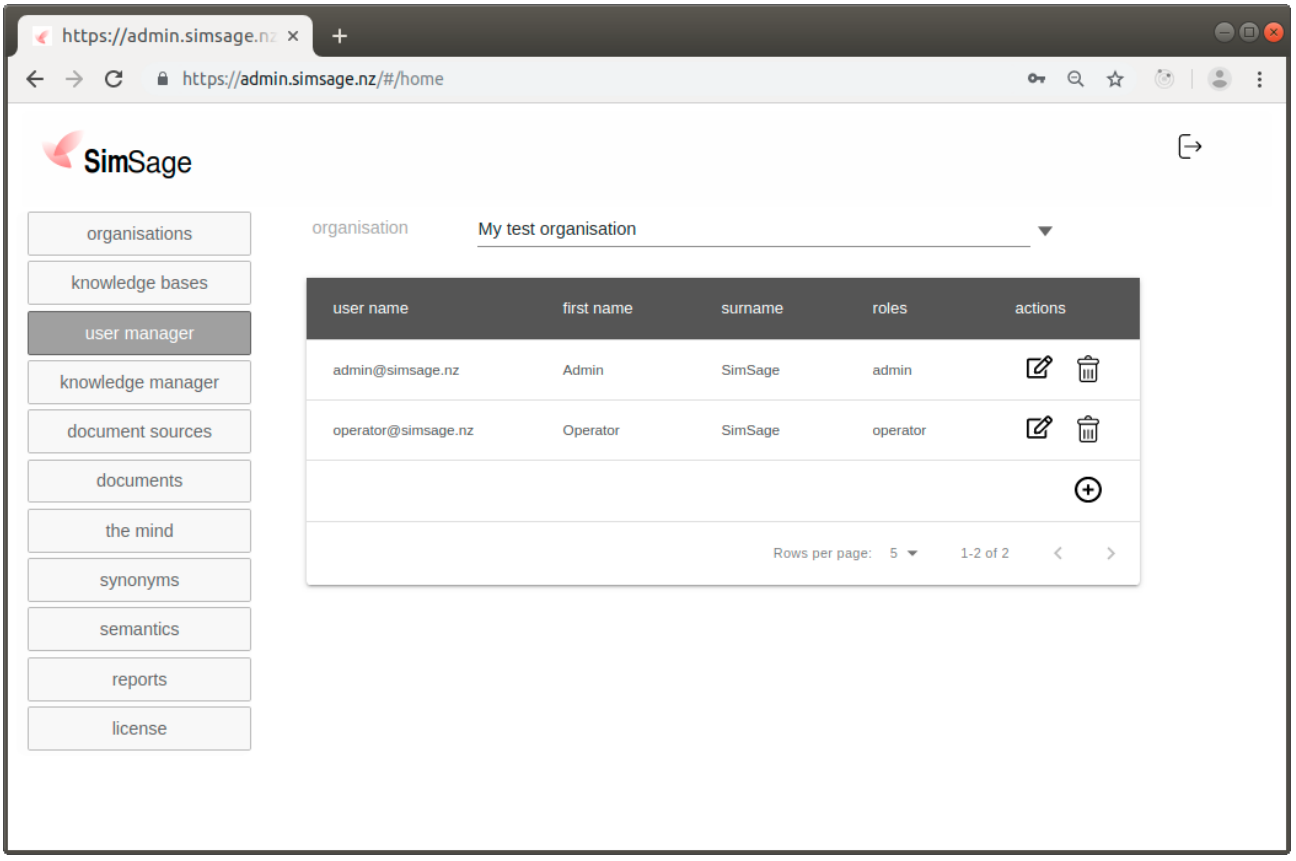
Note that a user cannot be saved without having any roles for the selected organisation. If you need to edit a user and change their details, make sure you do it for an organisation where they have assigned roles.

The operator role is a very special role. Selecting it brings up available knowledge bases as shown below.



You cannot save an operator role without it having been assigned a knowledge base first. This connects an operator to knowledge bases they can help with once signed in through the helpdesk interface.

Click the SAVE save button to make your changes. Leave the password field empty not to change it. Type in a new password to change this user's password. The interface now reflects the changes you made to the operator@simsage.nz user as shown below.



6. SimSage licensing

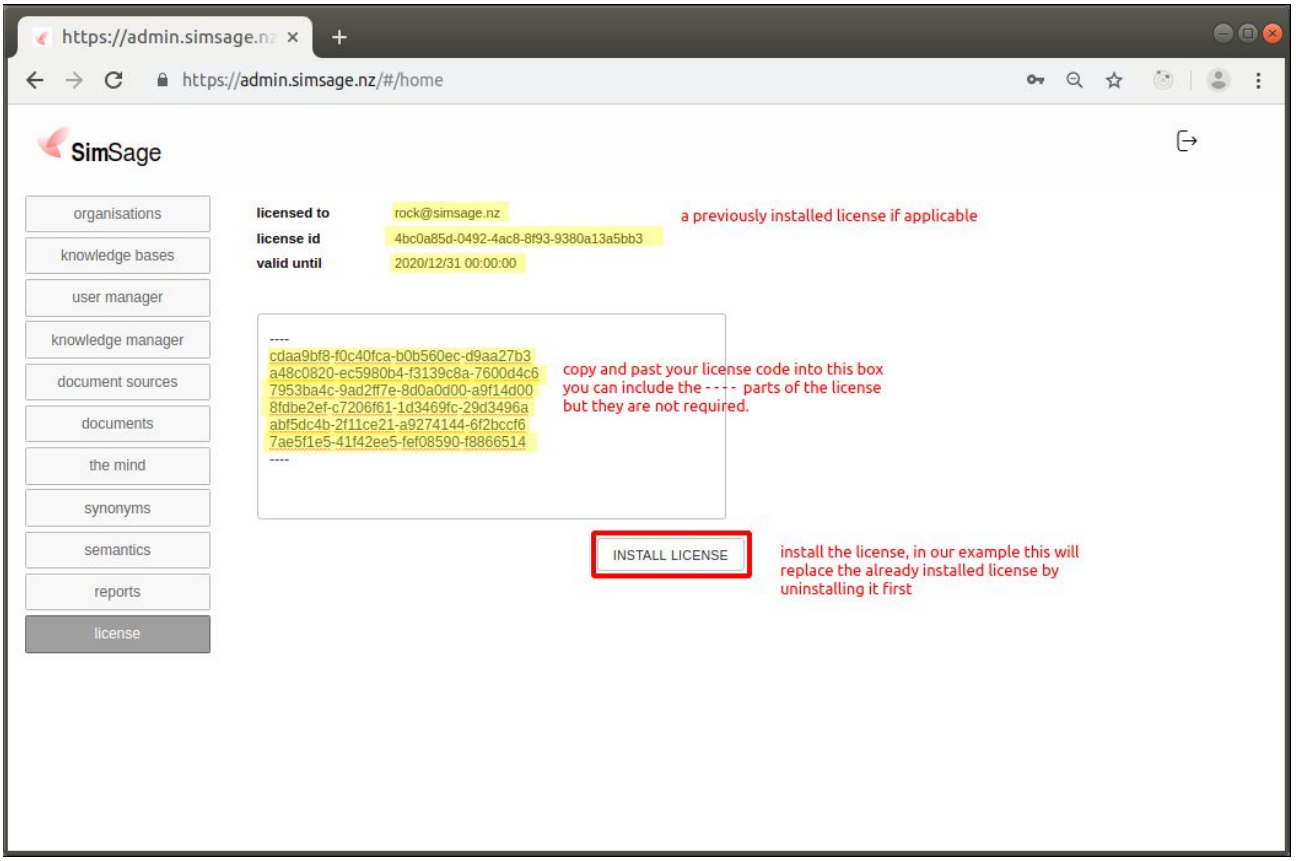
The SimSage software requires a license to operator. You should have been given a license along with your installation.

Click the license button in the administrative UI to view your current license.

A SimSage license looks like this;

---
cdaa9bf8-f0c40fca-b0b560ec-d9aa27b3
a48c0820-ec5980b4-f3139c8a-7600d4c6
7953ba4c-9ad2ff7e-8d0a0d00-a9f14d00
8fdbe2ef-c7206f61-1d3469fc-29d3496a
abf5dc4b-2f11ce21-a9274144-6f2bccf6
7ae5f1e5-41f42ee5-fef08590-f8866514

Copy and paste your license into the license box and click the INSTALL LICENSE install license button. Your SimSage screen should look something like this.



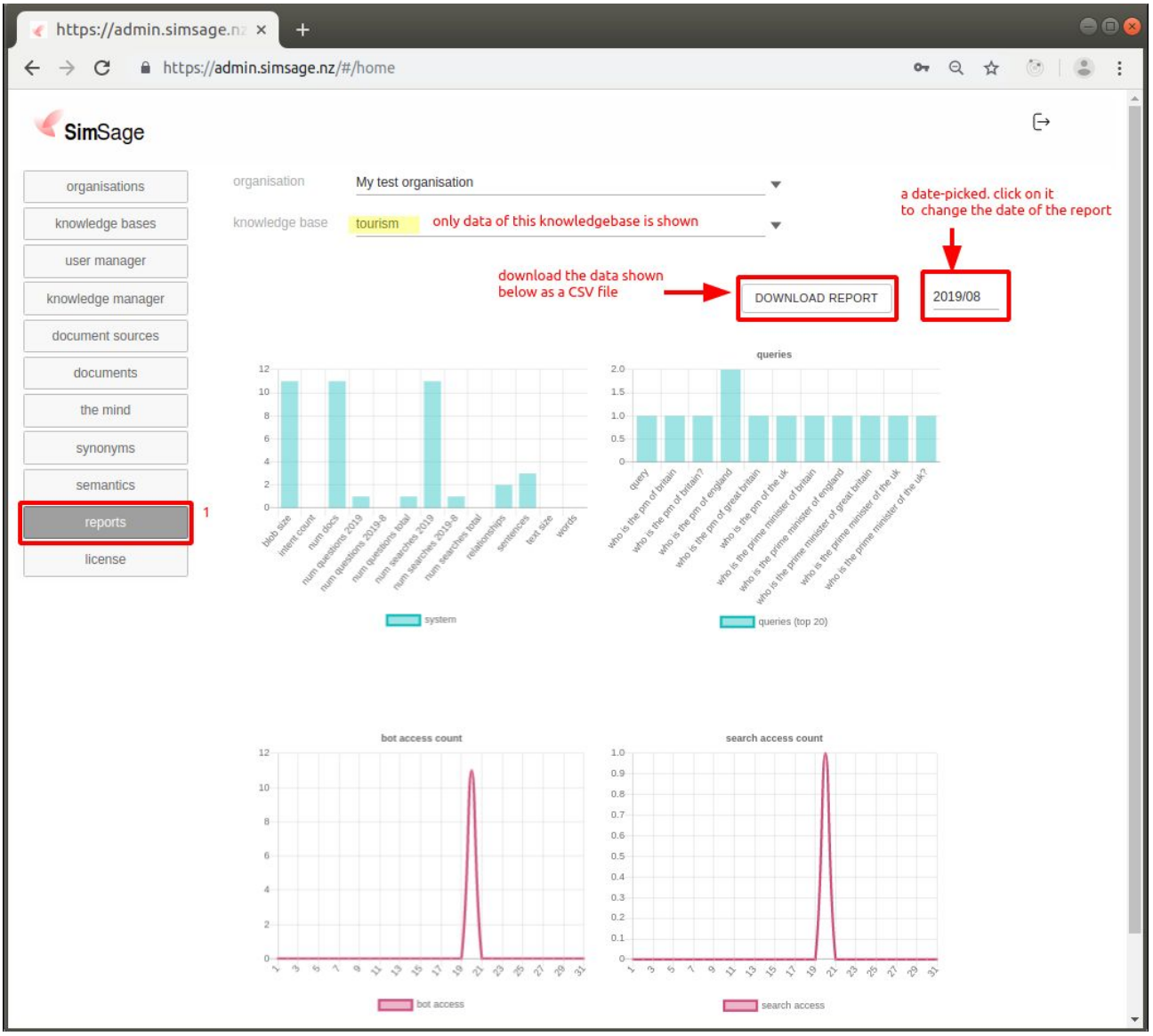
Clicking the install license button clears the license-code and changes the system from being an unlicensed system to a licensed system. Details of the license are displayed on this page, such as the owner of the license, the unique id of this license and the license's expiry date.

Any existing license will be uninstalled when you provide the system with a new license (i.e. you can try and re-install a license or try a different license if you have more than one). Make sure you keep your licenses in a safe place.

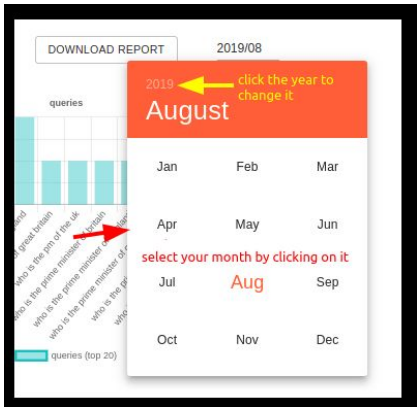
7. SimSage reporting

Click the reports button in the administrative interface to view the reports section. The reports section enables administrative users and manager to view SimSage system utilization and questions as well as performance.

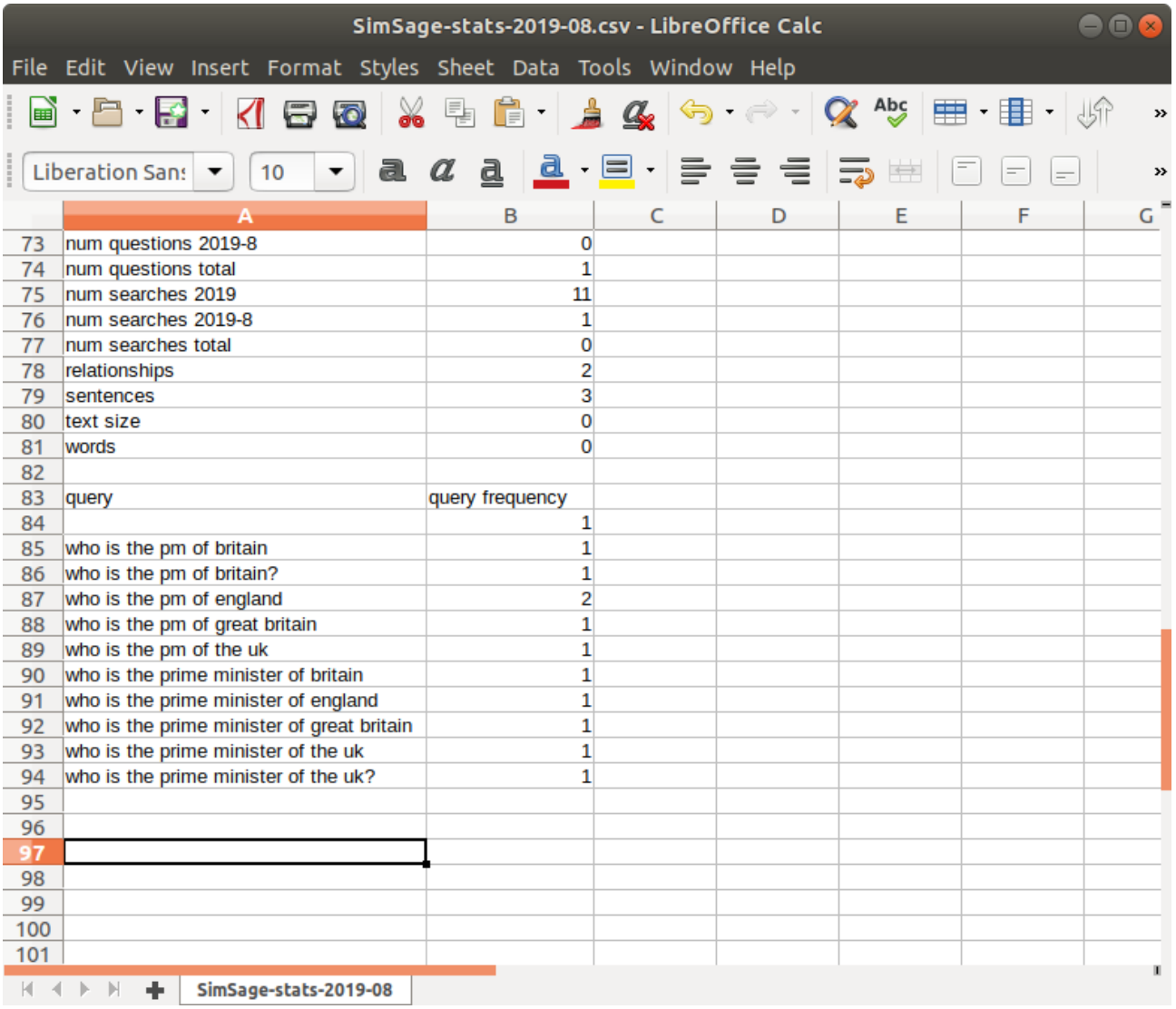
Utilization is displayed as the number of unique users per month. The top 20 questions can be viewed (top 20 by frequency). Performance of your question and answers can be downloaded by month.



Click the DOWNLOAD REPORT download report button or select the date picker next to it to download or change dates and view activity for the selected period.



SimSage reports are exported as text CSV files and can be opened with Excel or similar software tools for viewing and further analysis.



	A	B	C	D	E	F	G
73	num questions 2019-8	0					
74	num questions total	1					
75	num searches 2019	11					
76	num searches 2019-8	1					
77	num searches total	0					
78	relationships	2					
79	sentences	3					
80	text size	0					
81	words	0					
82							
83	query	query frequency					
84		1					
85	who is the pm of britain	1					
86	who is the pm of britain?	1					
87	who is the pm of england	2					
88	who is the pm of great britain	1					
89	who is the pm of the uk	1					
90	who is the prime minister of britain	1					
91	who is the prime minister of england	1					
92	who is the prime minister of great britain	1					
93	who is the prime minister of the uk	1					
94	who is the prime minister of the uk?	1					
95							
96							
97							
98							
99							
100							
101							

8. Semantics and Synonyms

Semantics and synonyms are important language constructs in SimSage. Synonyms are built into the semantic search system (along with other relationships) and can be fine-tuned for your industry for each knowledge base.

Synonyms define equivalents that can be used as substitutes for each other. Synonyms are idempotent, each word is considered the equivalent of the others.

Semantics define more general items. SimSage comes with 22,000 built in names with the “people” semantic. This includes common surnames, and first names (both masculine and feminine). Semantics can be used in SimSage for defining matching templates.

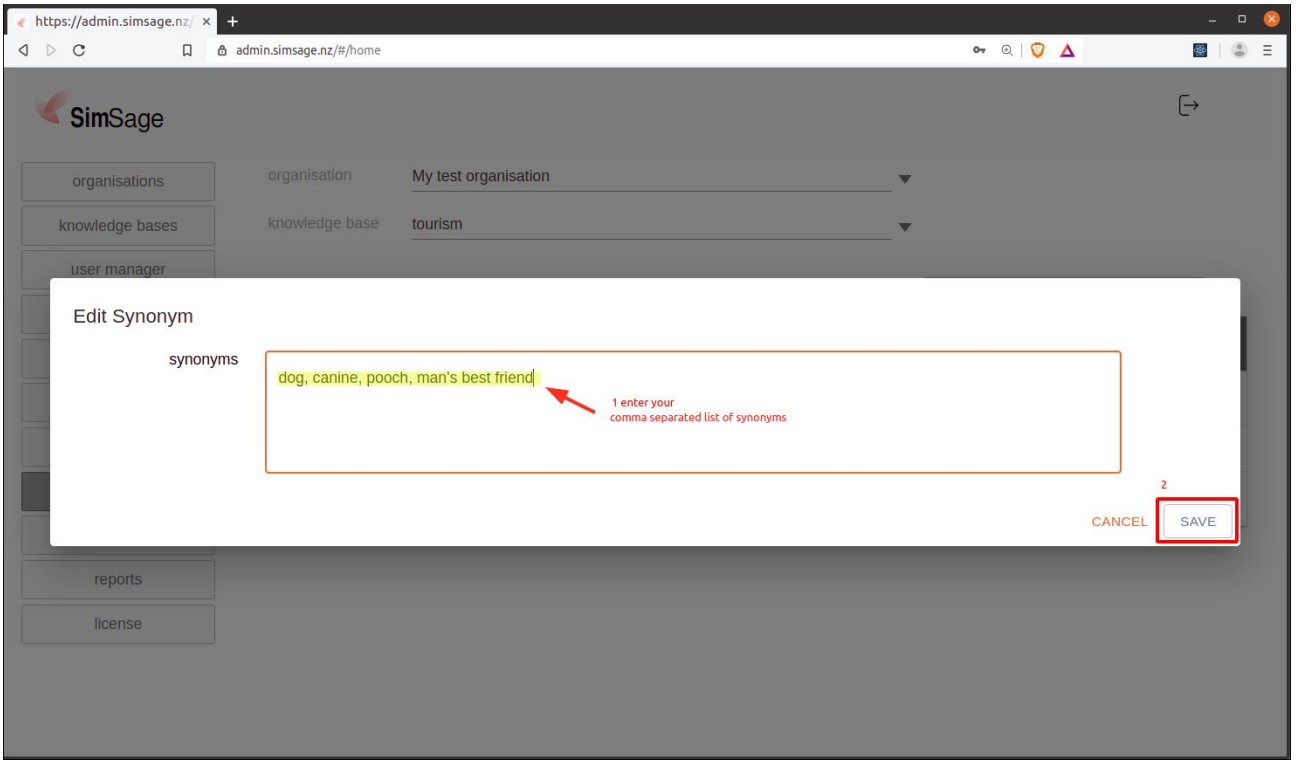
Suppose we define two semantics: Rock is a person, and Mary is a person. We can now create a question: who is person? And an answer: person1 is cool.

Person in the question will match “person”, “Rock” and “Mary” (and whoever else is define as a person out of the box). SimSage will take care of these matches automatically. SimSage will return the matched item in its question answer context. SimSage context is a powerful way for managing the matching system of SimSage. Context is beyond the scope of this document.

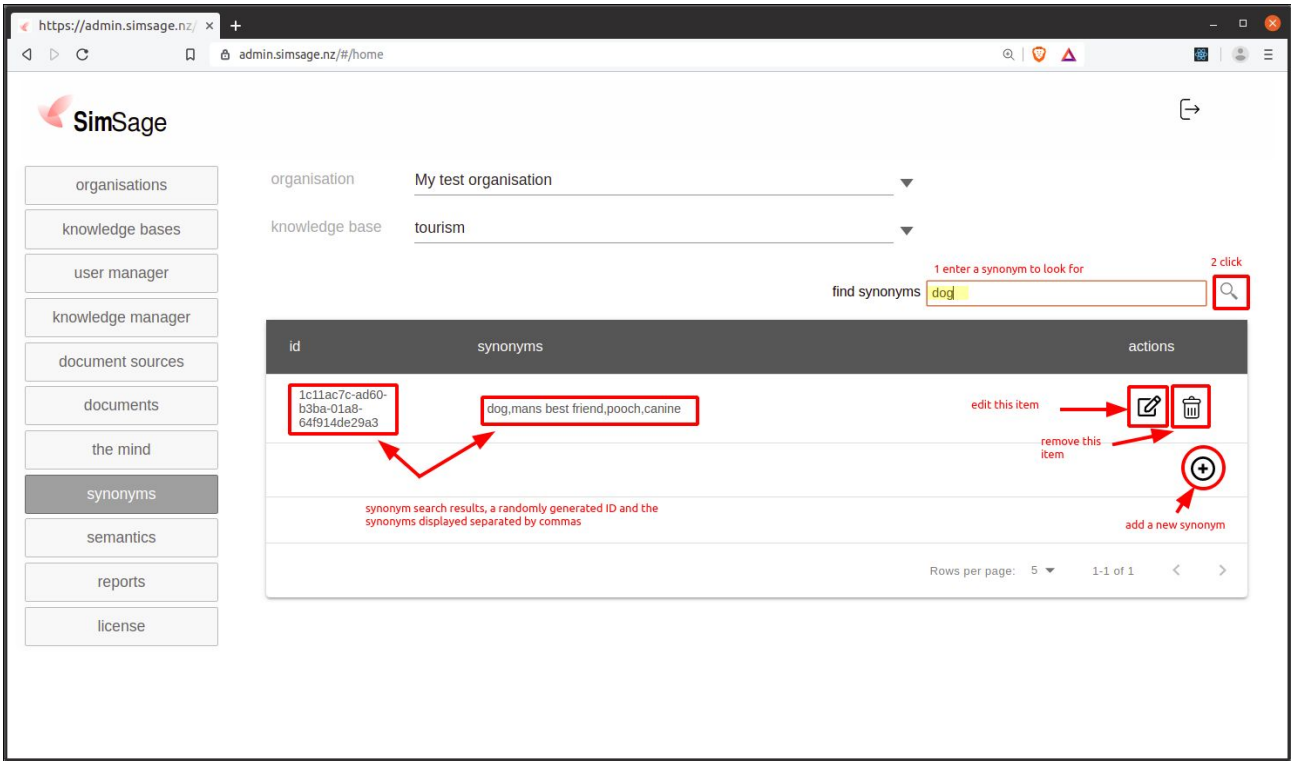
Ask SimSage: “who is Mary”, will match “who is person” and added a context item “person1 = Mary” (person1 refers to the first “person” matched in the question, looking from left to right).

8.1 SimSage Synonyms

Click the synonyms button to enter the synonyms section of the administrative UI. This interface is for finding, reviewing, adding and removing synonyms. You can filter / search for synonyms much like using the mind section.



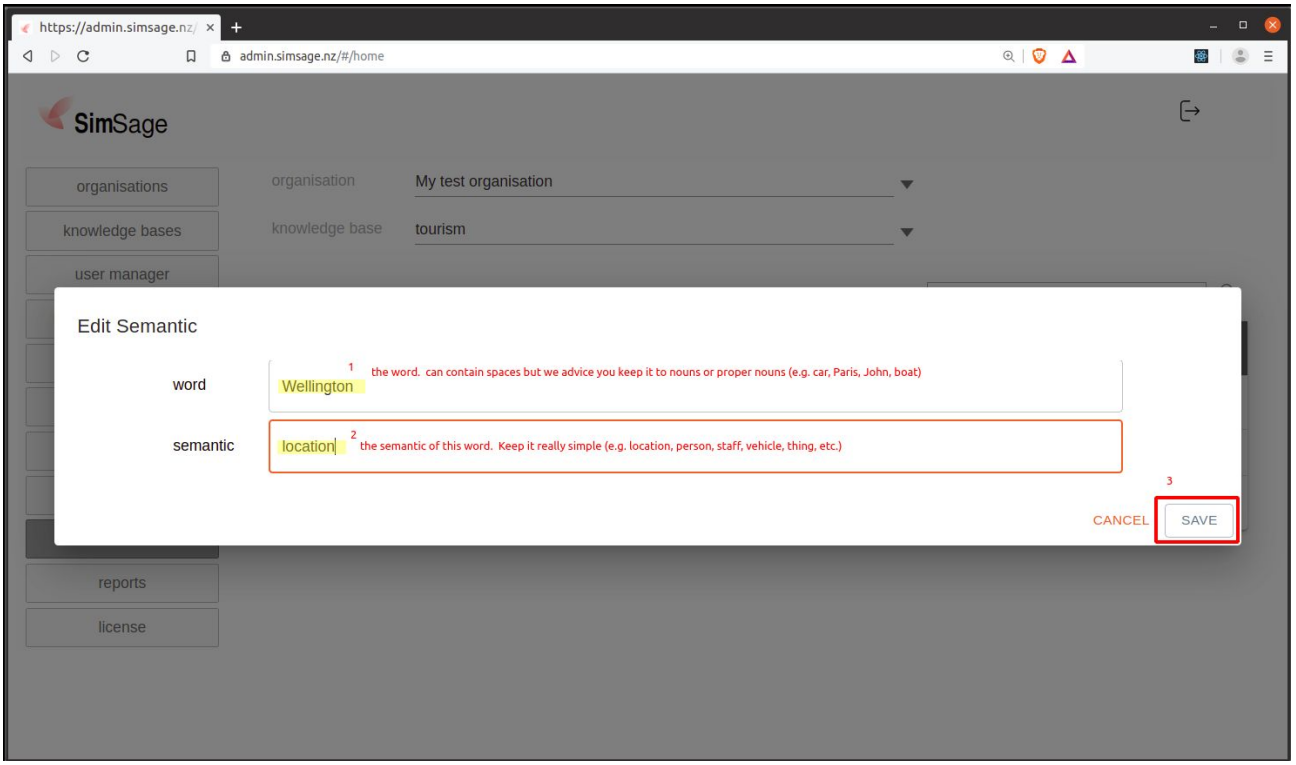
You create new synonyms by providing a comma separated list of items that are synonyms as shown in the example above. Synonyms created through the interface are automatically assigned unique ids.



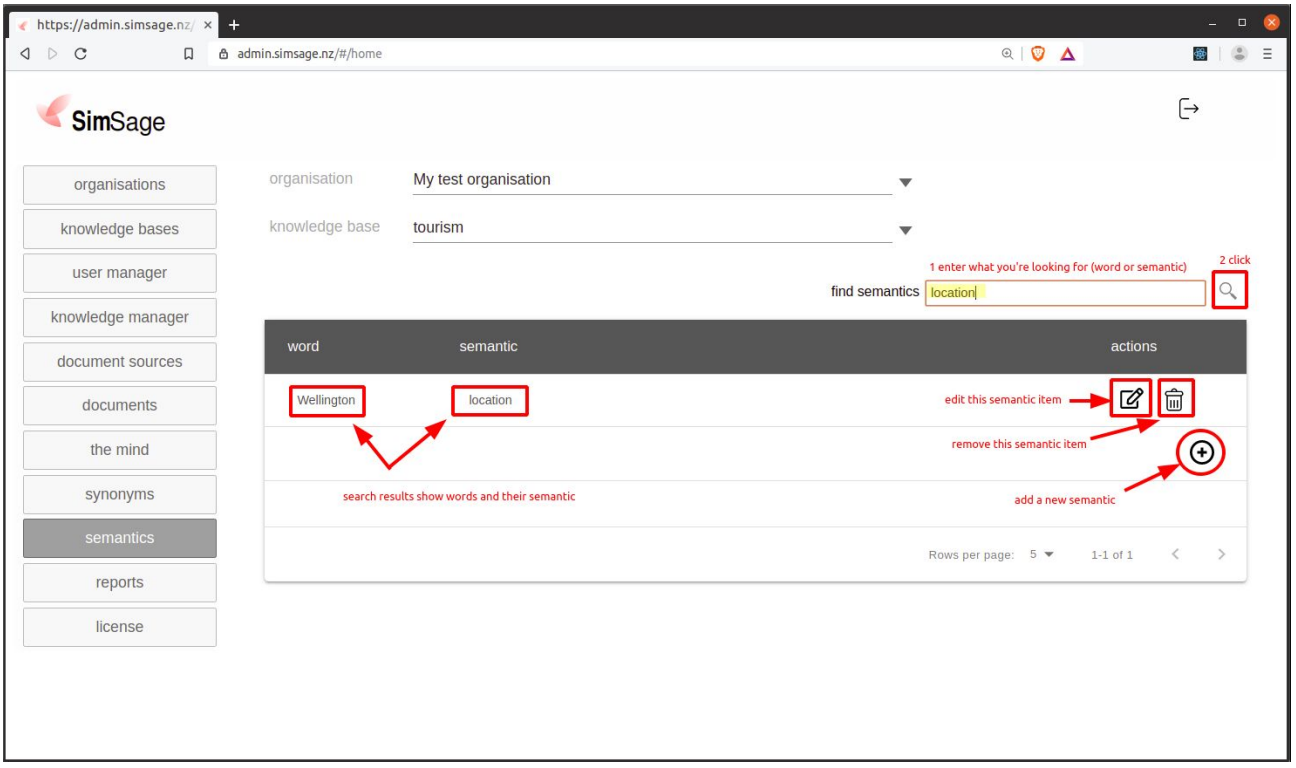
Searching for dog shows matching items and its id.

8.2 SimSage Semantics

Click the semantics button to enter the semantics section of the administrative UI. This interface is for finding, reviewing, adding and removing semantics. You can filter / search for synonyms much like using the mind section.



Semantics consist of a word and its more general “semantic”, as shown in the example above. Semantics do not have id numbers. The word itself is the id. In the above example, “Wellington” is the primary key. There can only be one “Wellington” in SimSage.



A search for either the word or its semantic will show results in the user interface as shown above.

9. Programming SimSage

This last section of this document is an advanced topic. This section goes deeper than any of the previous sections and can be skipped if you just wish to use SimSage through the administrative interface.

9.1 Bots and Natural Language

English is a natural language, a human language as opposed to a computer language. SimSage provides tools to greatly simplify the use of natural language. Our language-processors use what we call *user-intents*. At the highest level, a user-intent (aka. intent) is a question-answer pair. From the SimSage point of view, an intent is a language matching function with a series of associated actions for a given context.

A user-intent consists of a *question*, a *pre-context,* a *post-context*, and a series of *actions*. Intent language structures form the backbone of the SimSage programming language.

9.2 FAQs

Most of our customers use spreadsheets to capture their FAQ information. The SimSage platform can convert an FAQ like spreadsheet into a simple *SimSage program*.

A generated SimSage Program from a spreadsheet might look something like this.

```
%intent

["1","init","init"]
if ("what are you?" || "What is this?" || "Tell me about yourself") {
  metadata = "user data here";
  browser.write("I am SimSage, an artificial intelligence.");
  browser.url("https://simsage.nz");
}

["2","init","init"]
if ("who created you?" || "who was your creator?" || "Tell me about your creator") {
  metadata = "whatever you want";
  browser.write("Peter de Vocht created SimSage");
  browser.url("https://simsage.nz");
  browser.image("http://peter.nz/img/mockup/pmgdv.jpg");
}
```

We will get to what this all means in the next sections of this document. The column headers “id”, “question”, “answer”, “link” and “image” must start any spreadsheet in the positions shown.

9.3 Questions, synonyms and semantics

A question is the language matching part of SimSage. The SimSage language matcher is a very powerful.

A language matcher has contradicting requirements. It must be accurate to match the right response *yet* cover as much of English as possible to minimize effort. We want users and writers to use their own words to teach and use SimSage. We want their customers to be able to query SimSage using their words. We bridge that gap between you and your customer.

A question can be as simple as a simple English sentence you might expect a typical user to ask. We provide a "bracket notation" to have inline expansions of possibilities.

e.g.

what is the weather like (today | tomorrow |) ?

will internally expand to:

what is the weather like today?
what is the weather like tomorrow?
what is the weather like?

The bracket is treated like an "or" and can contain as many words as you like.

e.g.

what (is this | am I looking at)

Combinations of brackets quickly expand to many different possibilities internally.
e.g.

what (is this | am I looking at) (now | today |)

expands to six different possible questions

what is this now
what is this today
what is this
what am I looking at now
what am I looking at today
what am I looking at

Furthermore, you can separate different kinds of questions that don't fall nicely into bracket expansions using *question separators (double bar brackets ||)*.

What (is this | are you) || Tell me about (this | yourself)

expands to

What is this
What are you
Tell me about this
Tell me about yourself

Keep your questions simple and instruct your end-users to do the same. We are working on compound question handling and can deal with some more complex questions.

However, this is an evolving feature. One might appreciate that most of this work is cutting edge work in the current world of natural language -processing and -understanding.

9.4 Synonyms

Synonyms give you extra expansions for each word defined. Synonyms are defined in a separate worksheet of your spreadsheet. Synonyms should not be phrases but words.

e.g.

synonym ("1", "car, vroom vroom, automobile");

Defines "car", "vroom vroom", and "automobile" to be concepts that are interchangeable. You can now write a question like:

what is your car?

Which will get expanded to:

what is your car?
what is your vroom vroom?
what is your automobile?

Synonyms can form a powerful shortcut for words and cut-down the number of combinations you need to supply in your spreadsheet.

NB.

There are certain words that should never be synonyms in English. The verb "*to be*" and its conjugations (*be, being, been, was, is*) are too ambiguous for expansion, as are the verbs "*to do*" (*doing, done, did, does, do*) and "*to have*" (*have, had, having, has*). These verbs are commonly used as *auxiliary verbs* in English.

Stick to nouns and verbs. Other words like the question markers *what, who, where, when,* and *how* are too powerful and shouldn't have synonyms.

The same goes for other grammatical glue words like for instance conjunctions (*and, or, if, then, because*), and pronouns (*he, she, it*). SimSage is a sophisticated natural language processing system. ***Overriding English 'glue' words will confuse SimSage and potentially degrade its performance***.

All nouns (e.g. *car, train, water*) and proper nouns (*Iceland, John*) make good candidates. Adjectives are a good candidates too (e.g. *big, bigger, green, greenest*), but be careful with adjectives as they can be bonded to nouns to form unique concepts and can create unusual side-effects as a result (e.g. "green car" vs. "green light").

9.5 Semantics

Semantics are used for matching a set of concepts by its more general counterpart. Semantics are quite arbitrary and very powerful. Please be consistent with your concepts and pick whatever semantic hierarchy you want ahead of time. What concepts are

important to you? People (e.g. John, John Smith, Bob), Products (e.g. Desk Chair, Desk Light, spoon), Locations (e.g. London, Truro, Fallmouth), etc. Pick as many as you like but be consistent. For instance, don't confuse the concept of a "city" with a "location".

SimSage provides a rich set of semantics in the form of "is a" relationships (e.g. Peter **is a** person). These relationships form generalisations and equivalences at a higher level. Such relationships are called *hypernyms* in linguistics. SimSage comes preloaded with ~ 22,000 first names, and surnames with a "person" semantic.

By defining semantics, you can use the more general term (e.g. "city") to match instances of that semantic.

e.g. (using SimSage language statements)

semantic("London", "city");
semantic("Truro", "city");
semantic("Falmouth", "city");

and a question:

What is there to do in city

Will match

What is there to do in London

and set SimSage's matching context to: {"city1" = "London"}
(each semantic matched is numbered from left to right in the sentence).

9.6 Semantic Matching

The language matcher uses our own intelligent algorithms and neural networks to match language fragments. These are invisible to you as the developer and are rather complex as they related to Artificial Intelligence and Deep Learning. The team at SimSage is constantly working at tuning these networks and will keep you apprised of progress.

9.7 Answers and actions

Answers and actions are what are used after a user-intent match occurs. An intent statement consists of a matching part (the "if" part of the statement) and a set of actions to perform when a match occurs. In ordinary QA systems, the most common action is showing the user some text.

A single intent can have many actions, including multiple actions of the same kind. The current set of possible actions SimSage understands include:

4. browser.write: should be used to write text back to the user
5. browser.speak: intended to be used for speech synthesis
6. browser.image: intended to show an image from a URL
7. browser.url: anything that isn't an image, like a link to another site
8. user defined actions. Our out of the box bot examples won't understand these, but these are desgined to be used for your own interfaces.

A few examples of these actions using an excerpt of a SimSage program:

```
browser.write(“Hello there.”)
browser.write(“How are you?”)
browser.speak(“Hello there”)
browser.url(“www.simsage.nz”, “www.google.com”)
browser.image(“https://image.shutterstock.com/image-photo/macro-photo-male-wild-mallard
        -600w-1391625575.jpg”)
```

As the program’s creator you will be presented with these actions when SimSage matches an intent. How you use/interpret these actions is up to you.

9.8 IDs, Pre- and post-context

All human communication is based on context. The same information can mean different things in different contexts. A police officer asking someone “what have you been drinking?” requires a different response from a bartender asking the same question (although that would make for a bad bartender).

To facilitate context, SimSage implements a Finite-State-Machine. This adds extra context to the user’s session and the questions in the system. If the user has been interacting with SimSage regarding a certain product and asks the question “how much does it cost?” we can use the context of the product itself to resolve what price to get from a database.

SimSage will return the current context as part of any result returned. You, the developer, can use this information.

A context is just a label, a piece of text. A single word, or a bunch of words can form a context. These contexts “belong” to an intent. The initial context (the starting context in a conversation with SimSage) is called “init”. This is the only context that is hard coded into SimSage.

SimSage uses context to resolve intents when two different intents match one or more answers (with the same score). Perhaps this is best illustrated with an example.

Every intent has two “context” switches. One for matching (the pre-context) and one for changing the context after the response has been given (the post-context). In the next example we will be providing some context around the words “what do I do next?”. Imagine you are writing a SimSage program to replace a set of written instructions, like a manual. Our program starts when the user asks: “how do I begin?”. Our program then assumes that the user will repeatedly ask “what do I do next?”.

The following SimSage program illustrates this. This is the first time we’re putting together a few different concepts like “question”, “actions”, and “context”, so play careful attention to the syntax used.

```
[“1”, “init”, “second”]
if (“how do I begin”) {
        browser.write(“To start, you first you open the box.”);
}
```

```
["2", "second", "third"]
if ("what do I do next") {
        browser.write("The second step is to plug in your product to the wall socket.");
}

["3", "third", "fourth"]
if ("what do I do next") {
        browser.write("Switch on the product using the button at the front.");
}

["4", "fourth", "fourth"]
if ("what do I do next") {
        browser.write("Your product should be switched on now.");
}
```

Each *intent* is expressed as an "if" statement. The text inside the double quotes of the if statement's brackets is the *question*. The *actions* are contained within the curly brackets { ... }. The action shown above is a simple "browser.write" statement that can be used to show the user text.

The items within the square brackets are (from left to right) the **id**, **pre-context** and the **post-context**. You can have the pre- and post- contexts have identical values. This effectively means the context does not change when matched.

Each user-intent **must have a unique id**. The id is a string. This string can be any value, it can be numeric (inside the string), a guide, or whatever identification system you can come up with. **Intents with the same ID will overwrite each other**. Synonym Ids are separate from Intent Ids. They can have the same values but are not related.

You can write a program that doesn't use any context. In this case you can just use "init" for your contexts. This limits the sophistication of your bot's responses.

With this limited set of instructions, starting a new session in the SimSage Knowledge Base test section, the bot would respond as follows:

you> *how do I begin?*
SimSage> *To start, you first you open the box.*

you> *what do I do next?*
SimSage> *The second step is to plug in your product to the wall socket.*

you> *what do I do next?*
SimSage> *Switch on the product using the button at the front.*

you> *what do I do next?*
SimSage> *Your product should be switched on now.*

NB. This example is just to illustrate what it could look like to an end-user and does not represent any specific interface we provide

After this conversation, if you ask SimSage "what do I do next" again, it will repeat the last answer (since the last answer doesn't change *context*). If you ask SimSage "how do I begin", the journey will start from the beginning (the context resets to "second")

What if the user started a SimSage conversation with “what do I do next”? In this case, there is no context to match, but three matching answers. This is bad. To deal with this we should add another intent that reads:

```
[“5”, “init”, “second”]
if (“what do I do next”) {
   browser.write(“To start, you first you open the box.”);
}
```

This deals with the case where the user starts with “what do I do next” and then proceeds as it did before.

SimSage deals with multiple contexts for many different items. The user doesn’t need to worry about these. So, in effect, SimSage will keep track of multiple conversations and different topics and switch contexts as it needs to.

9.9 An example SimSage program

We leave you with a sample *SimSage program* that incorporates all we have discussed. None of these components are in any order. The SimSage parser uses a multi-pass process. We recommend you group synonyms and semantics before the intents for clarity.

```
%intent

// this is a comment that goes to the end of a line
synonym (“1”,  “program , technology, network,  equipment, service”)

semantic(“Peter de Vocht”,  “person”);   // Peter is a person
semantic(“Peter”,  “person”);
semantic(“de Vocht”,  “person”);

// id = 1, pre-context is “init” which is the default system context, post-context is “second”
[“1”, “init”, “second”]
if (“how do I begin”) {
        // you can of course put as many statements here as you like
        browser.write(“To start, you first you open the box.”);
        browser.speak(“Hello there”)
        browser.url(“www.peter.nz”, “www.google.com”)
        browser.image(“http://www.peter.nz/img/mockup/pmgdv.jpg”)
}

[“2”, “second”, “third”]
if (“what do I do next”) {
        browser.write(“The second step is to plug in your product to the wall socket.”);
}

[“3”, “third”, “fourth”]
if (“what do I do next”) {
        browser.write(“Switch on the product using the button at the front.”);
}

[“4”, “fourth”, “fourth”]
```

```
if (“what do I do next”) {
        browser.write(“Your product should be switched on now.”);
}


[“5”, “init”, “third”]
if (“what do I do next”) {
        write(“The second step is to plug in your product to the wall socket.”);
}
```

SimSage programs are text files. You can upload them as text files. See section 2.1 of this document on how to do just that.