Ghini Documentation

Реліз 1.0.x

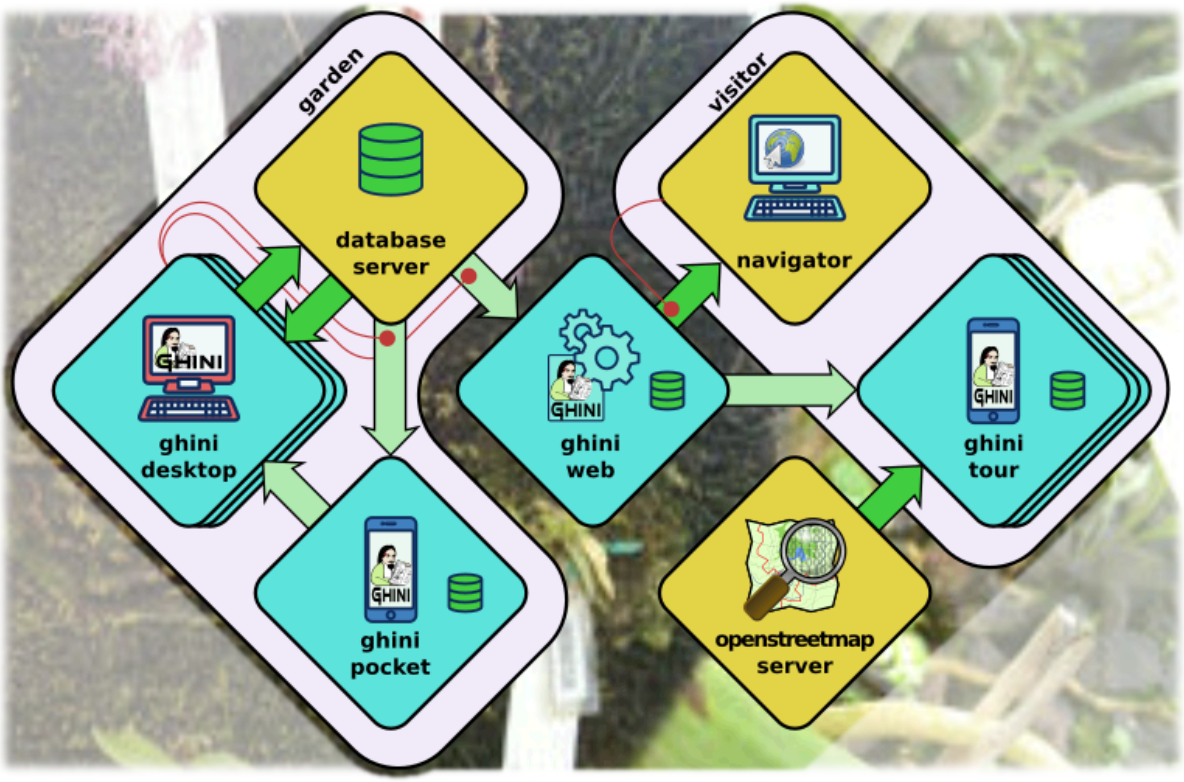
Mario Frasca

січня 11, 2023

Зміст

Ghini - це набір застосунків для управління колекціями ботанічних зразків.

- **ghini.desktop ** дозволяє створювати та запитувати базу даних, що представляє об'єкти та події у вашій колекції рослин.
- **ghini.web** publishes highlights from your database on the web.
- **ghini.pocket ** робить знімок вашої бази даних у вашому портативному пристрої.
- **ghini.tour** assists garden visitors with a map and spoken virtual panels.



The bulk of this documentation focuses on ghini.desktop. One final chapter presents the rest of the Ghini family: *ghini.pocket*, *ghini.web*, *ghini.tour*, and the *потоки даних між програмними компонентами*.

Все програмне забезпечення Ghini відкрите і безкоштовне . Наше автономне програмне забезпечення випускається під GNU Public License.

Розділ 1

Заяви

1.1 Цілі Ghini та основні моменти

Повинні ви використовувати це програмне забезпечення? Це питання для вас, щоб відповісти. Ми віримо, що якщо ви керуєте ботанічною колекцією, ви знайдете Ghini досить корисним, і ми сподіваємось, що ця сторінка переконає вас у цьому.

Ця сторінка показує, як Ghini робить програмне забезпечення задовольняє потреби ботанічного саду.

Якщо ви вже знаєте, і все, що ви хочете, це зробити щось практичне, просто встановіть програмне забезпечення, а потім перевірте наші користувацькі рецепти.

1.1.1 Ботанічний сад

Згідно з Вікіпедією, »Ботанічний сад - це сад, присвячений збиранню, вирощуванню та виведенню широкого спектру рослин, позначених їхніми ботанічними іменами«, і, як вважає Вікіпедія, »сад є запланованим місцем , як правило, на відкритому повітрі, відведеному для відображення, вирощування та насолоди рослинами та іншими формами природи.«

Таким чином, ми маємо в ботанічному саду як фізичний простір, сад, так і його динаміку, діяльність, до якої присвячений сад, діяльність, яка змушує нас називати сад ботанічним садом.

Fig. 1: **фізичний сад**

Fig. 2: **пов'язані з колекцією роботи в саду**

1.1.2 Ботанічний сад програмне забезпечення

З протилежного боку нашого міркування ми маємо прикладну програму Ghini, і знову цитуючи Вікіпедію »прикладна програма є комп'ютерною програмою, призначена для виконання групи узгоджених функцій, завдань або діяльності в інтересах користувача« або , коротше кажучи, »призначена, щоб допомогти людям виконувати свою діяльність«.

Дані та алгоритми в межах Ghini були розроблені, щоб відобразити фізичний простір та динаміку ботанічного саду.

Fig. 3: **основна структура бази даних Ghini**

На вищезгаданому малюнку, спрощений погляд на базу даних, виділені блоки є тими, що відносяться до об'єктів, які ви обов'язково повинні внести в базу даних.

Ми виділяємо три основні розділи в базі даних. Почніть читання графіка з правої сторони, з відповідною інформацією **Таксономія**, потім крок до адміністрування вашої **Колекції ** і, нарешті, розгляньте фізичний **Сад**.

Центральним елементом з точки зору Ghini є `Приєднання`. Після посилання на інші об'єкти бази даних ми зможемо краще зрозуміти структуру:

**Приєднання зв'язків Посадка до Видів **

> An `Accession` represents the action of receiving this specific plant material in the garden. As such, `Accession` is an abstract concept, it links physical living `Plantings` —groups of plants placed each at a `Location` in the garden— to the corresponding `Species`. It is not the same as an acquisition from a source, because in a single acquisition you can access material of more than one species. In other words: a single aquisition can embark multiple accessions. An `Accession` has zero or more `Plantings` associated to it (0..n), and it is at all times connected to exactly 1 `Species`. Each `Planting` belongs to exactly one `Accession`, each `Species` may have multiple `Accessions` relating to it.
>
> `Приєднання` залишається в базі даних, навіть якщо всі його `Насадження` були вилучені, продані або загинули. Визначення `Видів` з “ Приєднання“ послідовно пов'язує всі його `Насадження` до “ Видів“.

Accession at the base of the history of your plants

> `Розмноження` та `Контакти` забезпечують рослинний матеріал для саду; ця інформація є необов'язковою, і менші колекціонери можуть віддати перевагу залишити це в стороні. Процес `Розповсюдження` може виявитися невдалим, у більшості випадків це призведе точно до одного приєднання, але це також може призвести дещо інших таксонів, тому база даних дозволяє нуль або більше `Приєднань` за `Розповсюдження` (0..n). Також `Контакт` може надавати нуль або більше `Приєднання` (0..n).

Accession and Verification opinions

Спеціалісти можуть сформулювати свою думку щодо Видів, до яких належить Приєднання, шляхом надання Верифікації, підписання та встановлення відповідного рівня довіри.

Доступ до власних розповсюджень

Якщо Приєднання отримано з садового розсадника від успішного Розмноження, Розмноження``зв'язує ``Приєднання і всі його Насадження до батьківського Насадження, насіння або вегетативного батька.

Навіть після викладеного вище пояснення, нові користувачі, як правило, як і раніше запитують, чому їм потрібно пройти через екран Приєднання, тоді як все, що вони хочуть - це вставити Рослина у колекцію, і знову: що це річ «приєднання» в будь-якому випадку? Більшість дискусій у мережі не роблять поняття чіткішим. Один з наших користувачів дав приклад, який я радий включити в документацію Ghini.

варіант використання

1. На початку 2007 року ми отримали п'ять саджанців *Heliconia longa* (Вид рослини) від нашого сусіда (джерело Контакт). Оскільки це було перше придбання року, ми назвали їх 2007.0001 (ми надали їм єдиний унікальний код Приєднання, з кількістю 5) і ми посадили їх усіх разом в одному Розташування як одиничне Насадження, також з кількістю 5.

2. На момент написання, через дев'ять років, Приєднання 2007.0001 має 6 окремих Насаджень, кожне в іншому Розташуванні у нашому саду, отриманих вегетативно (безстатево) від оригінальних 5 рослин. Наше єдине втручання було поділ, переміщення, і, звичайно, записати цю інформацію в базі даних. Загальна кількість рослин перевищує 40.

3. Нові Насадження, отримані (допоміжним) половим Розповсюдженням, надходять в нашу базу даних за різними кодами Приєднання, де наш сад є джерелом Контакт, і де ми знаємо, який із наших Насаджень - батько сім'ї.

ці три випадки переводяться в кілька коротких історій використання:

1. Активізуйте меню Вставка → Приєднання, перевірте існування та правильність Види *Heliconia longa*, вкажіть початкову кількість Приєднання; додайте своє Насадження на бажане Розташування.

2. відредагуйте Насадження для виправлення кількості живих рослин - повторіть це так часто, як це необхідно.

3. відредагуйте Насадження, щоб розділити його на окремі Розташування - це створює інші Насадження під тим самим Приєднанням.

4. редагувати Насадження, щоб додати (насіння) Розповсюдження.

5. редагувати Насадження щоб оновити статус Розповсюдження.

6. активізуйте меню Вставити → Приєднання, щоб пов'язати приєднання до успішної пробної версії Розповсюдження; додайте Насадження у бажаному

`Розташуванні`.

Зокрема, можливість розділити `Насадження` в декількох різних `Розташуваннях` і тримати все рівномірно асоційованими з одним `Видом`, або можливість зберегти інформацію про `Насадження` які були вилучені з колекції, допоможе підтвердити наявність рівня абстракції `Приєднання`.

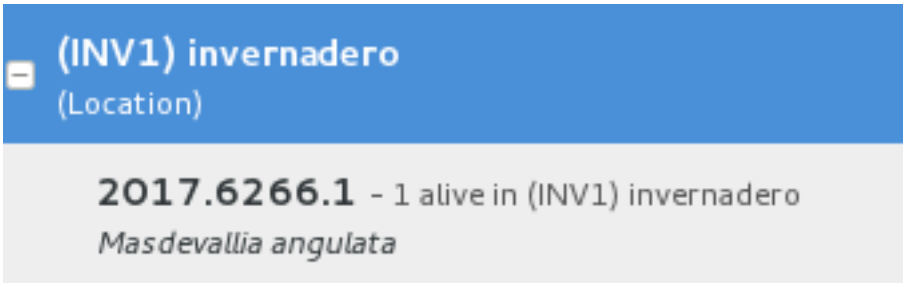
1.1.3 Hypersimplified view

People using Ghini only sporadically may prefer ignoring the database structure and look at it as two nested sequences of objects, each element of the sequence being necessary to add element at the next level.

In order to get down to an Accession, you will need four levels, as in this example:

A quite complete set of Families and Genera are inserted in your database at the moment Ghini initializes it. So all you need is adding Species and Accessions, in this order.

When placing a physical Plant (relative to an Accession) somewhere in the garden, you need to describe this «somewhere» digitally, as a Location in the garden.



1.1.4 Підсвічування

не такий короткий перелік основних моментів, мав на увазі, щоб підігріти апетит.

класифікаційна інформації

Коли ви вперше запустите Ghini, і підключитися до бази даних, Ghini буде ініціалізувати базу даних не тільки з усіма таблицями, які потрібно запустити, але і заповнювати таблиці таксонів для рангів сім'ї і роду, використовуючи дані з "RBG Kew's Family and Genera list from Vascular Plant Families and Genera compiled by R. K. Brummitt and published by the Royal Botanic Gardens, Kew in 1992". У 2015 році ми переглянули дані щодо Orchidaceae, використовуючи, як джерело, "Tropicos, botanical information system at the Missouri Botanical Garden - www.tropicos.org".

імпорт даних

Ghini дозволить вам імпортувати будь-які дані, які ви помістили в проміжному форматі json. Те, що ви імпортуєте, завершить те, що ви вже маєте в базі даних. Якщо вам потрібна допомога, ви можете попросити фахівця Ghini допомогти вам перетворити ваші дані в проміжний Ghini формат json.

синоніми

Ghini дозволить визначити синоніми для видів, родів, сімейств. Крім того, ця інформація може бути представлена в проміжному форматі json і імпортована в існуючу базу даних Ghini.

наукова відповідальність

Ghini реалізує концепцію „приєднань", як проміжну між фізичною рослиною (або їх групою) і абстрактним таксоном. Кожне приєднання може пов'язати ті ж рослини з різними таксонами, якщо два систематики не згодні з ідентифікації: кожен систематик може мати свою думку і не потрібно перезаписати роботу один одного. Всі перевірки можуть бути знайдені в базі даних, з відміткою часу і підписом.

допомагає офф-лайн ідентифікації

Ghini дозволяє асоціювати світлини з фізичними рослинами, це може допомогти розпізнати рослину в разі втрати етикетки, або допомогти таксономічній ідентифікації, якщо систематик не доступний весь час.

експорт і звіти

Ghini дозволить вам експортувати звіт в будь-якому текстовому форматі. Він використовує потужний шаблонний движок під назвою „mako", який дозволить експортувати дані вибравши будь-який формат, що вам потрібний. Після установки, кілька прикладів можна знайти в підкаталозі mako .

анотувати ваші дані

Ви можете додати примітки до рослин, приєднань, видів, ... Примітки можуть бути класифіковані і використані для пошуку або звітів.

сад або гербарій

Управління розташуванням рослин.

історія бази даних

Всі зміни в базі даних зберігаються в базі даних, як журнал історії. Всі зміни „підписані“ і з відміткою часу. Ghini дозволяє легко отримати список всіх змін в останній робочий день або тиждень, або в будь-який певний період в минулому.

простий і потужний пошук

Ghini дозволяє шукати в базі даних, використовуючи прості ключові слова, наприклад: назва місця або ім'я роду, або ви можете написати складніші запити, які не досягають складності SQL, але дозволять вам гідний рівень деталізації локалізованих даних.

агностична база даних

Ghini не є системою управління базами даних, щоб не вигадувати знову колесо. Вона працює зберігаючи свої дані в базі даних SQL, і буде підключатися до будь-якої системи управління базами даних, яка приймає роз'єм SQLAlchemy. Це означає будь-яку досить сучасну систему управління базами даних і включає в себе MySQL, PostgreSQL, Oracle. Вона також може працювати з sqlite, що для цілей окремих користувачів цілком достатньо і ефективно. Якщо підключити Ghini до реальної системи бази даних, ви можете розглянути можливість зробити базу даних частиною системи LAMP (Linux-Apache-MySQL-Php) і включити ваші дані в реальному часі на вашому веб-сайті установи.

агностична мова

The program was born in English and all its technical and user documentation is first written in that language. Both technical and user documentation use `gettext`, an advanced tool for semi-automatic translation.

The program has been translated and can be used in various other languages, including Spanish (97%), French (82%), Portuguese (71%), to name some Southern American languages, as well as Ukrainian (100%) and Czech (71%).

Translation of documentation goes a bit slower, with only Ukrainian, Spanish and Italian at more than 50%.

агностична платформа

Встановлення Ghini в Windows - це простий і лінійний процес, займе не більше 10 хвилин. Ghini народився в Linux і встановити його на ubuntu, fedora або debian ще простіше. MacOSX заснований на unix, можна успішно запустити процедуру встановлення Linux на будь-якому нещодавньому комп'ютері Apple, після кількох кроків підготовки.

легко оновлюється

Процес установки буде створювати оновлювану установку, де оновлення займе менше ніж одну хвилину. Залежно від величини зворотного зв'язку, яку ми отримуємо, ми будемо проводити оновлення через кожні кілька днів або один раз в певний час.

протестовані

Ghini постійно та екстенсивно перевіряється, дещо, що робить регрес функціональності близьким до неможливості. Кожне оновлення автоматично перевіряється, на Travis Continuous Integration службі. Інтеграція TravisCI з платформою github ускладнить для нас випуск будь-якого матеріалу, що має єдиний тест на несправність.

Більшість змін і доповнень, які ми робимо, приходять з деяким додатковим модульним тестуванням, яке визначає поведінку і робить будь-які небажані зміни добре видимі.

настроюємий/розширяємий

Ghini розширюється за допомогою плагінів і може бути налаштований відповідно до потреб установи.

1.2 Місія і Бачення

Тут ми визначаємо, хто ми, що ми думаємо про нашу роботу, що ви можете очікувати від цього проєкту.

1.2.1 Хто стоїть за Ghini

Ghini - це невеликий набір програм, призначених для того, щоб розпорядники колекцій могли керувати своєю колекцією також у цифровій формі.

Ghini народився ще в 2004 році як Bauble, в Ботанічному саду Белізу. Пізніше вона була адаптована до потреб ще декількох садів. Оригінальний програміст Brett Adams зробив це програмне забезпечення загальним, випустивши його за ліцензією GPL.

Після багатьох років застою Mario Frasca відновив цей проєкт і назвав його як Ghini на честь Luca Ghini, засновника першого європейського ботанічного саду та гербарію. Mario Frasca почав захищати, подорожувати, розповсюджувати, розвивати, розширювати, перевизначати, документувати, і зараз Mario Frasca це написав, шукаючи користувачів, з проханням зворотного зв'язку.

Отже, зараз на Ghini не один розробник, але невелика, але зростаюча глобальна спільнота користувачів.

Переклади надаються добровольцями, які в більшості своїй залишаються поза сценами, перекладаючи відсутні терміни або пропозиції, і знову зникають.

Щоб зробити речі чіткішими, коли ми говоримо про Ghini, але ми повинні вказати, чи це Ghini (програмне забезпечення), або Ghini (люди), якщо, очевидно, ми не маємо на увазі обидві речі.

1.2.2 Місія

Наша мета, як Ghini Software, - забезпечити безкоштовне програмне забезпечення, що має перевірену якість, і дозволити кожному встановити його, якщо вони прагнуть цього. Ми також прагнемо полегшити доступ до функціональних знань у формі документації або шляхом встановлення контактів між користувачами або між користувачами та професіоналами програмного забезпечення.

Усі наші джерела, програмне забезпечення та документація є відкритими та вільними, і ми вітаємо і стимулюємо людей використовувати та надавати свої послуги. Для полегшення формування спільноти, всі наші платформи можна протестувати без реєстрації. Реєстрація потрібна, якщо ви хочете внести свій внесок.

Ghini вітає формування груп користувачів, згрупування сил для визначення та фінансування подальшого розвитку, і ми вітаємо розробників, що надають програмне забезпечення, з будь-якого куточка світу, і ми стимулюємо та допомагаємо їм дотримуватися вимог високої якості, перш ніж ми приймемо внесок код у джерелах програмного забезпечення.

1.2.3 Бачення

Бачення служить для позначення майбутнього реалістичного та привабливого способу майбутнього іміджу того, що хочемо ми, наша організація. Вона слугує мотивацією, оскільки вона візуалізує виклик і напрямок необхідних змін, щоб рости і процвітати.

- до 2020 року
- контрольна точка
- спільнота
- розвиток
- інтеграція з веб-порталом

- географічна інформація

Розділ 2

Установка Ghini

2.1 Установка

Ghini.desktop - це крос-платформна програма, яка працюватиме на Unix-машинах, таких як GNU/Linux і MacOSX, а також у Windows.

одним рядком для поспішних користувачів.

> Користувачі Linux просто завантажують і запускають інсталяційний скрипт. Ви можете ознайомитися з документацією пізніше.
>
> Windows users in a real hurry don't the instructions and use a recent Windows installer. You do not miss any functional feature, but you have less chances to contribute to development.
>
> Mac users are never in a hurry, are they?

Ghini підтримується дуже небагатьма людьми, які зосереджують свою увагу на підвищенні його функціональних частин, ніж на написанні фантастичних інсталяторів. Замість декількох інсталяторів, ми пропонуємо єдину процедуру встановлення крос-платформ. Це має декілька великих переваг, які ви оцінити, як ми почнемо.

Встановлення засноване на запуску сценарію.

- Сценарій GNU/Linux дбає про все, від dependecies до інсталяції для користувачів у групі `ghini`.
- Сценарій Windows потребує, щоб ви спочатку встановите пару речей.
- У MacOSX ми використовуємо той самий сценарій, що і на GNU/Linux. Оскільки OSX не має менеджера пакетів за замовчуванням, ми встановлюємо його, і

використовуємо, перш ніж запускати скрипт.

Дотримуючись нашої процедури встановлення, по закінчені Ghini працює в віртуальному середовищі Python, всі Python залежності встановлено локально, не конфліктуючи з будь-якою іншою програмою Python, яку ви можете мати у вашій системі.

Залежність, яка не відповідає віртуальному середовищу Python, полягає в наступному: Python, virtualenv, GTK + і PyGTK. Їх установка залежить від платформи.

Якщо пізніше ви вирішите видалити Ghini, ви просто видалите віртуальне середовище, яке є каталогом, з усім його вмістом.

2.1.1 Встановлення на GNU/Linux

Відкрийте вікно терміналу оболонки та виконайте наведені нижче інструкції.

1. Завантажте *devinstall.sh* скрипт:

   devinstall.sh

2. Invoke the script from a terminal window, starting at the directory where you downloaded it, like this:

```
bash ./devinstall.sh
```

   The script will produce quite some output, which you can safely ignore.

глобальна установка

When almost ready, the installation script will ask you for your password. This lets it create a `ghini` user group, initialise it to just yourself, make the just created `ghini` script available to the whole `ghini` user group.

If feeling paranoid, you can safely not give your password and interrupt the script there.

Possibly the main advantage of a global installation is being able to find Ghini in the application menus of your graphic environment.

---

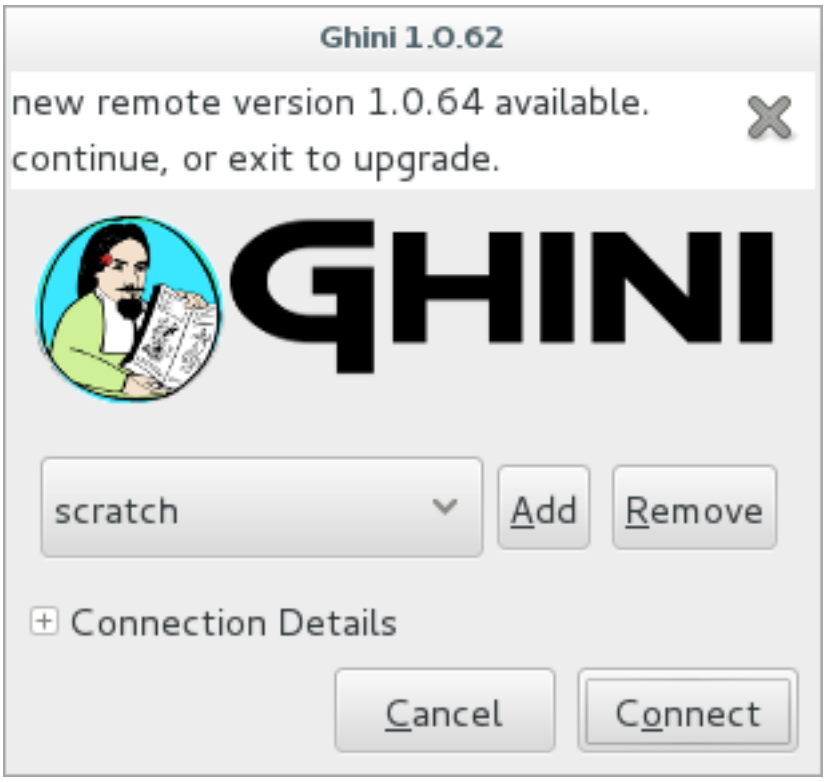
3. You can now start ghini by invoking the `ghini` script:

```
ghini
```

   1. You use the same `ghini` script to update ghini to the latest released production patch:

```
~/bin/ghini -u
```

      This is what you would do when ghini shows you something like this:



2. Users of the global installation will also type `ghini` to invoke the program, but they will get to a different script, located in `/usr/local/bin`. This globally available `ghini` script cannot be used to update a ghini installation.

3. Again the same `ghini` script lets you install the optional database connectors: option `-p` is for PostgreSQL, option `-m` is for MySQL/MariaDB, but you can also install both at the same time:

```
~/bin/ghini -pm
```

   Будьте обережні: може знадобитися вирішувати залежності. Як це зробити, залежить від того, який GNU/Linux ви використовуєте. Перевірте свою документацію про розповсюдження.

4. You can also use the `ghini` script to switch to a different production line. At the moment `1.0` is the stable one, but you can select `1.1` if you want to help us with its development:

```
~/bin/ghini -s 1.1
```

примітка новачка

Щоб запустити скрипт, спершу переконайтеся, що ви помітили назву каталогу, до якого ви завантажили скрипт, після чого ви відкриваєте вікно термінала, і в цьому вікні набираєте *bash*, після чого пробіл і повне ім'я сценарію включаючи ім'я каталогу, і натисніть клавішу Enter.

технічна примітка

You can study the script to see what steps if runs for you.

Коротше кажучи, він встановить залежності, які не можуть бути задоволені в віртуальному середовищі, то він створить віртуальне середовище з ім'ям `ghide`, використовуйте git для завантаження джерел до каталогу з назвою `~/Local/github/Ghini/ghini.desktop`, і підключіть git до гілки `ghini-1.0` (це ви можете розглянути виробничу лінію), він створює ghini, завантажує всі залишки залежностей у віртуальне середовище і, нарешті, створює сценарій запуску `ghini`.

If you have `sudo` permissions, it will be placed in `/usr/local/bin`, otherwise in your `~/bin` folder.

Далі...

Підключення до бази даних.

2.1.2 Установка на MacOSX

Будучи macOS в середовищі unix, більшість речей буде працювати так само, як на GNU/Linux (різновид).

Перш за все, вам потрібно речі, які є невід'ємною частиною unix середовища, але які відсутні в off-the-shelf mac:

1. інструменти розробника: xcode. перевірте сторінку Вікіпедії яка версія підтримується на mac.
2. менеджер пакетів: homebrew (tigerbrew для старих версій OSX).

Встановлення на старіші MacOS.

Every time we tested, we could only solve all dependencies on the two or three most recent macOS versions. In April 2015 this excluded macOS 10.6 and older. In September 2017 this excluded macOS 10.8 and older. We never had a problem with the lastest macOS.

The problem lies with homebrew and some of the packages we rely on. The message you have to fear looks like this:

```
Do not report this issue to Homebrew/brew or Homebrew/core!
```

The only solution I can offer is: please update your system.

On the bright side, if at any time in the past you did install ghini.desktop on your older and now unsupported macOS, you will always be able to update ghini.desktop to the latest version.

З встановленим вище, відкрийте вікно термінала і запустіть:

```
brew doctor
```

переконайтеся, що ви розумієте виявлені проблеми, і виправте їх. pygtk знадобиться xquartz і brew не дозволить залежність автоматично. або встановіть xquartz, використовуючи brew або якщо ви вважаєте за краще:

```
brew install Caskroom/cask/xquartz
```

тоді встановіть решту залежностей:

```
brew install git
brew install pygtk  # takes time and installs all dependencies
```

дотримуватися всіх інструкцій про те, як активувати то, що ви встановили.

In particular, make sure you read and understand all reports starting with `If you need to have this software`.

You will need at least the following four lines in your `~/.bash_profile`:

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
export PATH="/usr/local/opt/gettext/bin:$PATH"
export PATH="/usr/local/opt/python/libexec/bin:$PATH"
```

Activate the profile by sourcing it:

```
. ~/.bash_profile
```

Перш ніж ми можемо запустити `devinstall.sh` як на GNU / Linux, нам все ще потрібно встановити пару пакетів python, глобально. Зробити це:

```
sudo -H pip install virtualenv lxml
```

Інше - так само, як на звичайній машині Unix. Прочитайте наведені вище інструкції GNU/Linux, дотримуйтесь їх, насолоджуйтеся.

As an optional aesthetical step, consider packaging your `~/bin/ghini` script in a platypus application bundle. The `images` directory contains a 128×128 icon.

Далі...

Підключення до бази даних.

2.1.3 Установка на Windows

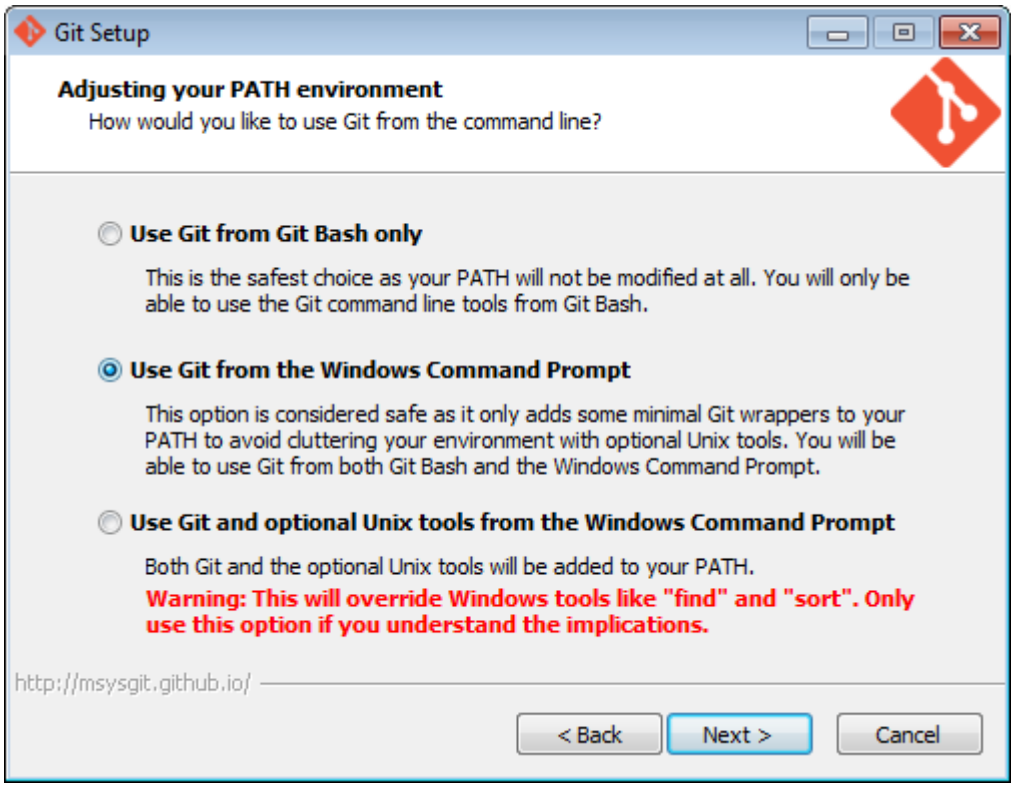
Нижче описані кроки допоможуть вам встановити Git, Gtk, Python та консолі бази даних Python. При правильному настроюванні цього середовища процедура встановлення Ghini працює як на GNU/Linux. Завершальні кроки знов є специфічними для Windows.

Примітка: Ghini був протестований і, як відомо, працює на W-XP, W-7 до W-10. Хоча це має працювати чудово на інших версіях Windows, він не був ретельно протестований.

Кроки по установці на Windows:

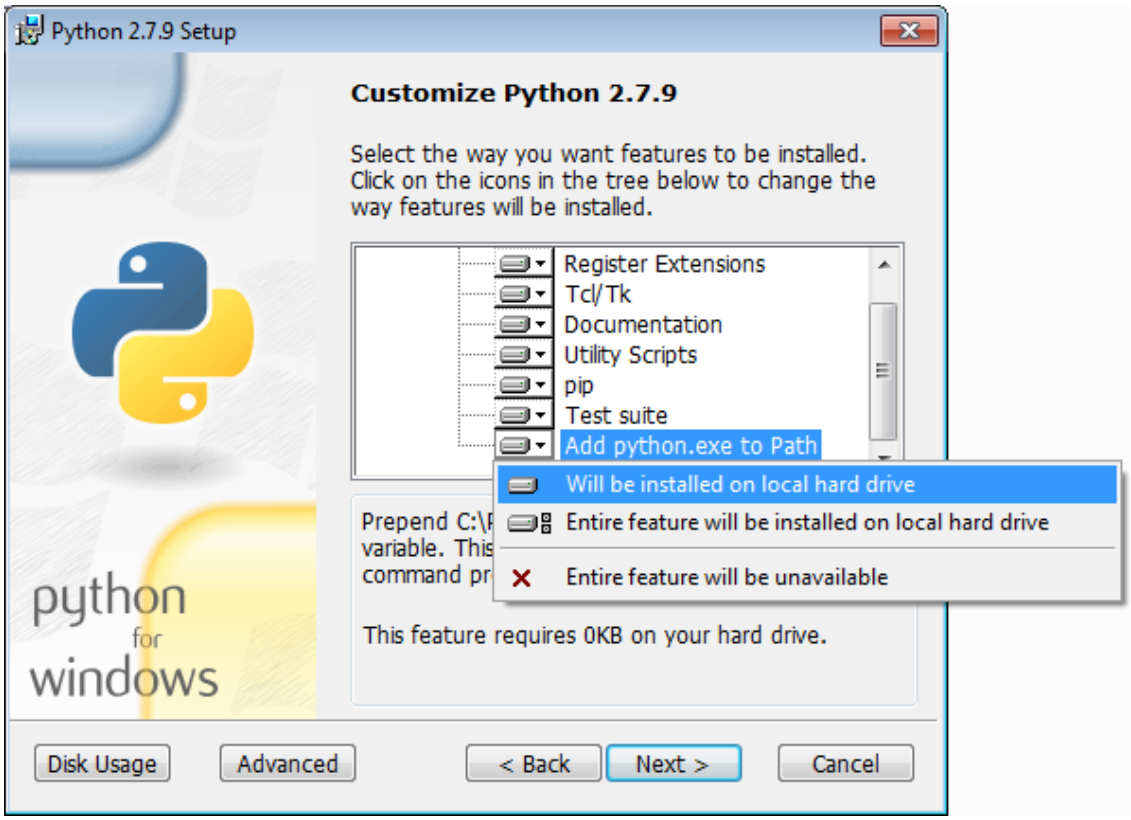
1. завантажити і встановити `git` (поставляється з unix-like `sh` і включає `vi`). Візьміть його з Git завантаження.

   всі опції за замовчуванням в порядку, за винятком git повинен бути виконуваним з командного рядка:



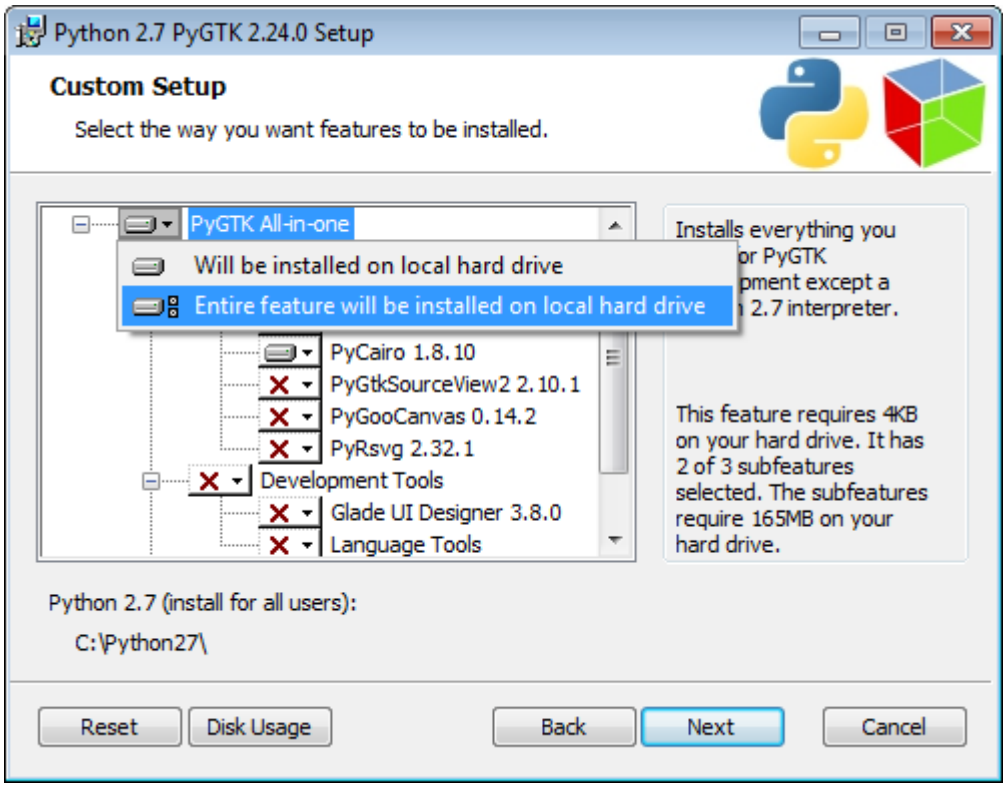
2. download and install Python 2.x (32bit). Grab it from the Python official site.

   Під час встановлення Python вставте Python в PATH:



3. завантажити `pygtk` з офіційного джерела. (для цього потрібно 32-бітний python). Переконайтеся що завантажте версію «все в одній».

   Зробіть повну установку, вибравши все:



4. (Possibly necessary, maybe superfluous) install lxml, you can grab this from the

pypi archives

Пам'ятайте, вам потрібна 32-бітова версія для Python 2.7.

5. (за бажанням), завантажте та встановіть конектор бази даних, відмінний від `sqlite3`.

   If you plan using PostgreSQL, the best Windows binary library for Python is psycopg and is Made in Italy.

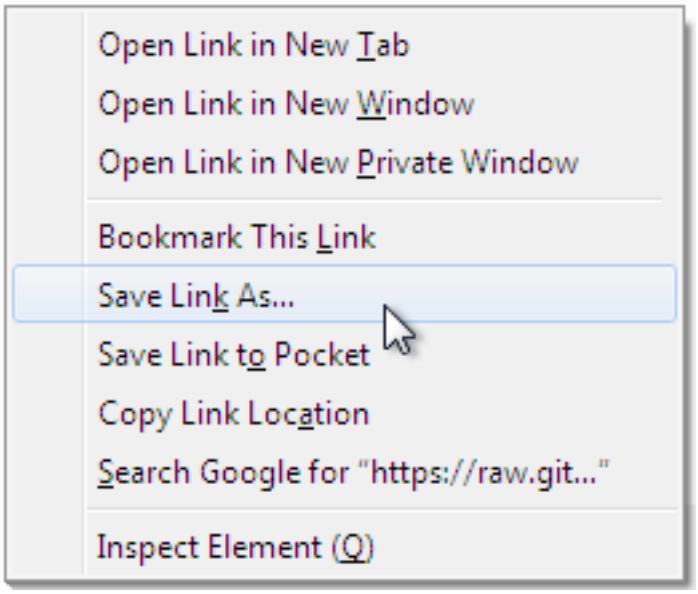
6. **ПЕРЕЗАВАНТАЖЕННЯ**

   Гей, це Windows, вам необхідно перезавантажити, щоб зміни вступили в силу!

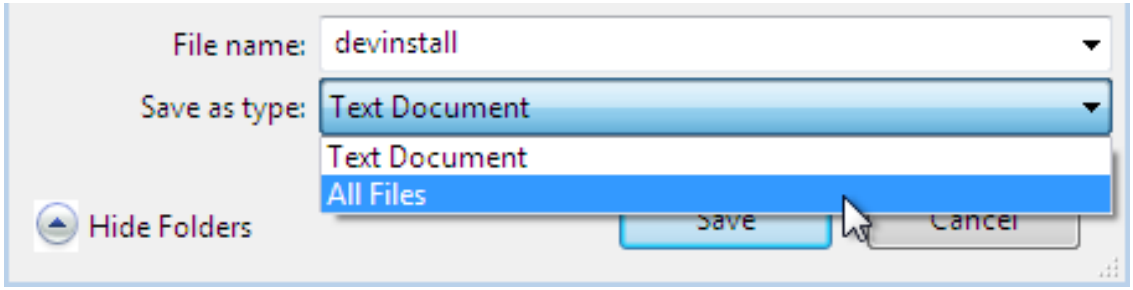
7. We're done with the dependecies, now we can download and run the batch file:

   devinstall.bat

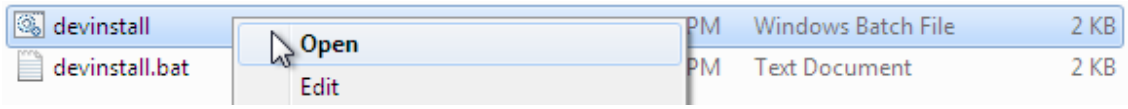
   Please don't just follow the above link. Instead: right click, save link as...



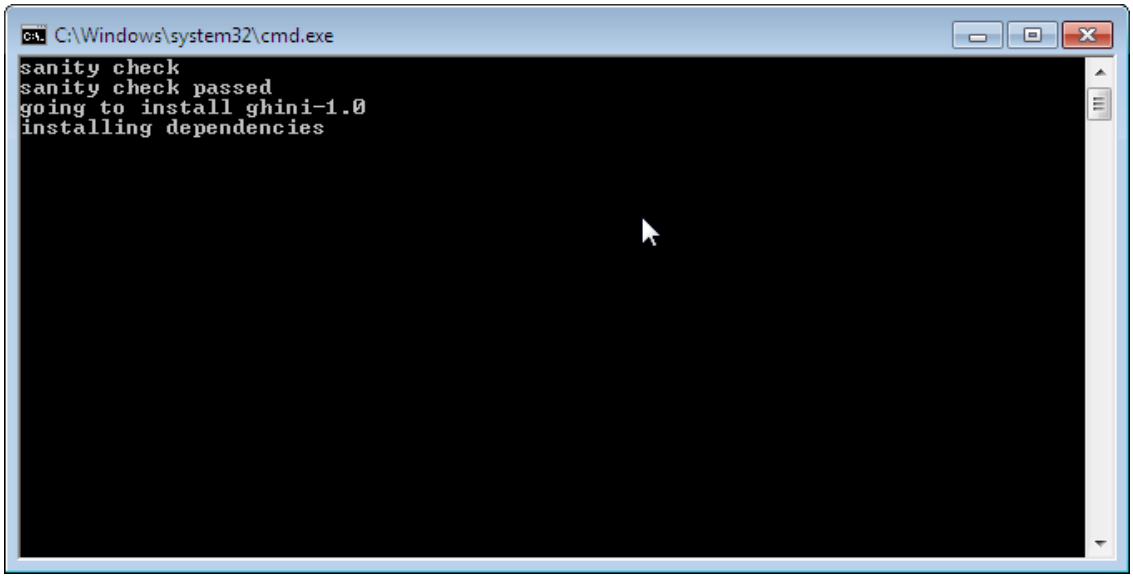
Also make sure you don't let Windows convert the script to a text document.



Now **Open** the script to run it. Please note: in the below image, we have saved the file twice, once letting Windows convert it to a text document, and again as a Windows Batch File. Opening the batch file will run the script. Opening the text document will show you the code of the batch file, which isn't going to lead us anywhere.



If you installed everything as described here, the first thing you should see when you start the installation script is a window like this, and your computer will be busy during a couple of minutes, showing you what it is doing.



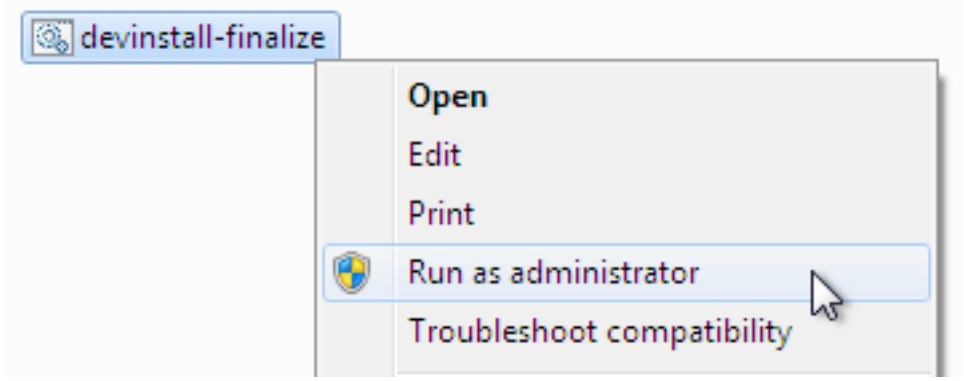
Запуск `devinstall.bat` перетягне сховище `ghini.desktop` з github у ваш домашній каталог, під `Local\github\Ghini`, перевірте виробничу лінію `ghini-1.0`, створіть віртуальне середовище і встановіть в нього ghini.

Ви також можете запустити `devinstall.bat`, передаючи його як аргумент числовій частині виробничої лінії, якої ви хочете дотримуватися.

Це останній етап інсталяції, який сильно залежить від робочого підключення до Інтернету.

Операція може зайняти декілька хвилин, залежно від швидкості вашого інтернет-з'єднання.

8. Останній етап установки створює групу Ghini і ярлики в меню Пуск Windows для всіх користувачів. Для цього вам потрібно запустити скрипт із правами адміністратора. Сценарій називається `devinstall-finalize.bat`, він прямо в папці HOME, і був створений на попередньому кроці.



Клацніть правою кнопкою миші на ньому, виберіть запустити як адміністратор, підтвердити, що хочете внести зміни в свій комп'ютер. Ці зміни знаходяться тільки в меню Пуск: створіть групу Ghini, помістіть ярлик Ghini.

9. download the batch file, it will help you staying up-to-date:

   ghini-update.bat

   If you are on a recent Ghini installation, each time you start the program, Ghini will check on the development site and alert you of any newer ghini release within your chosen production line.

   Any time you want to update your installation, just run the `ghini-update.bat` script, it will hardly take one minute.

   How to save a batch file, and how to run it: check the the quite detailed instructions given for `devinstall.bat`.

If you need to generate PDF reports, you can use the XLS based report generator and you will need to download and install Apache FOP. After extracting the FOP archive you will need to include the directory you extracted to in your PATH.

If you choose for PostScript reports, you can use the Mako based report generator and there are no further dependencies.

Далі...

Підключення до бази даних.

2.1.4 Installing on Android

`ghini.desktop` is a desktop program, obviously you don't install it on a handheld device, but we do offer the option, for your Android phone or tablet, to install `ghini.pocket`.

`ghini.pocket` is a small data viewer, it comes handy if you want to have a quick idea of a plant species, its source, and date it entered the garden, just by scanning a plant label.

Installation is as easy as it can be: just look for it in Google Play, and install it.

Export the data from `ghini.desktop` to pocket format, copy it to your device, enjoy.

Розділ 3

Керівництво користувача

3.1 Початкова конфігурація

Після успішного встановлення складніші організації повинні налаштувати свою базу даних і Ghini відповідно до їх конфігурації. Ця сторінка зосереджена на цьому завданні. Якщо ви не знаєте, чого це стосується, будь ласка, прочитайте частину щодо SQLite.

3.1.1 У вас SQLite?

Це вперше ви використовуєте Ghini, ви збираєтеся працювати в автономному режимі, у вас немає найменшої ідеї управління системою керування базою даних? Якщо ви відповіли так на будь-який з запитань, вам, напевно, краще дотримуватися бази даних на базі файлів SQLite, легкої, швидкої, з нульовою адміністрацію.

З SQLite, вам не потрібно ніякої підготовки, і ви можете продовжити *connecting*.

З іншого боку, якщо ви хочете підключити більше однієї робочої станції до тієї ж бази даних, або якщо ви хочете зробити свої дані доступними для інших клієнтів, як це може бути веб-сервер у налаштуваннях LAMP, слід враховувати збереження вашої бази даних В системі управління базами даних, як PostgreSQL або MySQL/MariaDB, обидва варіанти підтримуються Ghini.

When connecting to a database server as one of the above, you have to manually do the following: Create at least one user; Create your database; Give at least one user full permissions on your database; If you plan having more database users: Give one of your users the `CREATEROLE` privilege; Consider the user with the `CREATEROLE` privilege as a super-user, not meant to handle data directly; Keep your super-user credentials in a very safe place.

When this is done, Ghini will be able to proceed, creating the tables and importing the default data set. The process is database-dependent and it falls beyond the scope of this manual.

Якщо у вас вже є озноб або біль в животі, немає необхідності турбуватися, просто тримайтеся SQLite, ви не зазнаєте невдачі на особливостях, ні в продуктивності.

Some more hints if you need PostgreSQL

Start simple, don't do all at the same time. Review the online manual, or download and study the offline version.

As said above, create a database, a user, make this user the owner of the database, decide whether you're going to need multiple users, and preferably reserve a user for database and normal user creation. This super-user should be your only user with `CREATEROLE` privilege.

All normal users will need all privileges on all tables and sequences, something you can do from the *Tools→Users* menu. If you have any difficulty, please open an issue about it.

Connect using the `psql` interactive terminal. Create a `~/.pgpass` file (read more about it in the manual), tweak your `pg_hba.conf` and `postgresql.conf` files, until you can connect using the command:

```
psql <mydb> --username <myuser> --no-password --host <mydbhost>
```

With the above setup, connecting from ghini will be an obvious task.

3.1.2 Підключення до бази даних

При запуску Ghini вперше спливає діалогове вікно підключення.

Цілком очевидно, що, якщо це перший раз, коли ви запускаєте Ghini, у вас немає ще підключення і Ghini попередить вас про це.

Це сповіщення буде показано при першій активації, а також у майбутньому, якщо список ваших з'єднань стане порожнім. Як говориться: натисніть **Додати**, щоб створити своє перше з'єднання.

Просто вставте ім'я для підключення, щось значуще що ви асоціюєте з колекцією для представлення в базі даних (наприклад: «мій домашній сад») і натисніть **OK**. Ви повернетеся до попереднього екрана, але ваше ім'я з'єднання буде обрано і Деталі Підключення будуть розширені.

вказати параметри з'єднання

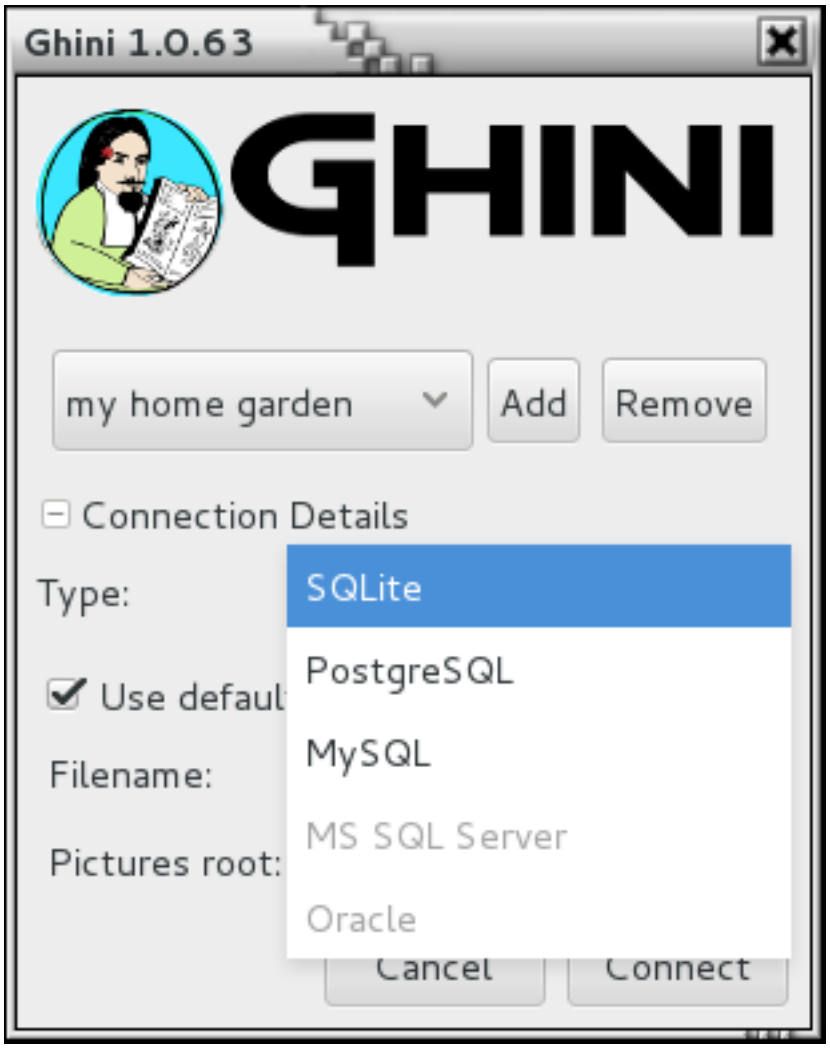
Якщо ви не знаєте, що робити тут, Ghini допоможе вам залишатися в безпеці. Активуйте **Використання розташування за замовчуванням** шляхом встановлення прапорця і створіть перше з'єднання, натиснувши на **Підключення**.

На даний момент ви можете сміливо пропустити решту цього розділу і продовжити читати наступний розділ.

точно налаштувати деталі з'єднання

За замовчуванням Ghini використовує файл бази даних на основі SQLite. В процесі установки у вас був вибір (і він у вас залишається і після установки), додати інші, ніж SQLite за замовчуванням з'єднувачі бази даних.

У цьому прикладі, Ghini може підключитися до SQLite, PostgreSQL і MySQL, але немає з'єднувачів доступних для Oracle або MS SQL Server.



Якщо ви використовуєте SQLite, все, що вам дійсно потрібно це вказати ім'я з'єднання. Якщо ви дозволите Ghini використовувати ім'я файлу за замовчуванням, то Ghini створює файл бази даних з тим же ім'ям, що і з'єднання і `.db` розширення, і теку зображень з таким же ім'ям і без розширення, як в `~/.bauble``в Linux/MacOSX або в ``AppData\Roaming\Bauble` в Windows.

Проте з SQLite, ви, можливо, отримали або завантажили bauble базу даних, і ви хочете підключитися до неї. В цьому випадку ви не дозволяєте Ghini використовувати ім'я файлу за замовчуванням, лише перегляньте на комп'ютері в тому місці, де ви зберегли Ghini SQLite файл бази даних.

Якщо ви використовуєте інше з'єднання бази даних, діалогове вікно буде виглядати по-іншому і вам запропонують варіант для точної настройки всіх параметрів необхідних для підключення до бази даних за вашим вибором.

If you are connecting to an existing database you can continue to *Редагування і вставка даних* and subsequently searching-in-ghini, otherwise read on to the following section on initializing a database for Ghini.

Якщо ви плануєте пов'язати світлини з установкою, вкажіть *pictures root* теку. Сенс цього детальніше пояснено в *Редагування і вставка даних*.

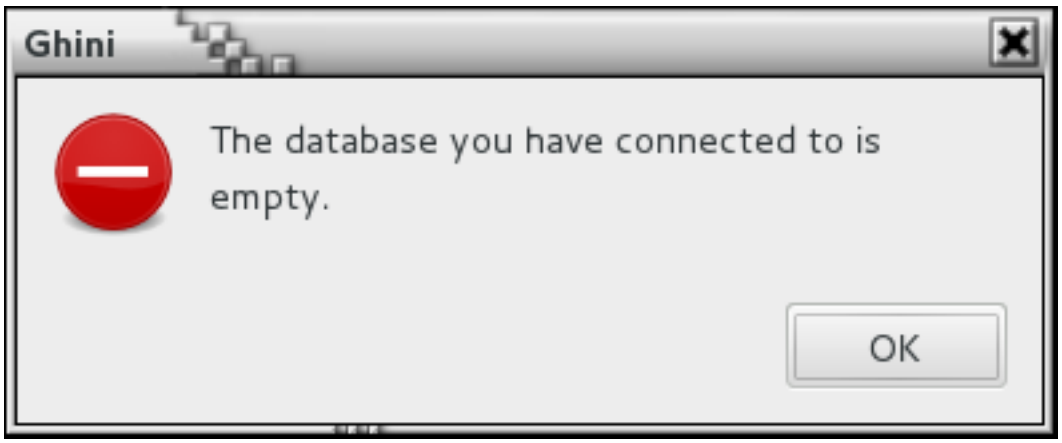
A sample SQLite database

Indeed we have a sample database, from our pilot garden «El Cuchubo», in Mompox, Colombia. We have a zipped sample database for ghini-1.0.

Download and unzip it to the location of your choice, then start Ghini, create a connection named possibly `cuchubo`, or `sample`, and edit the Connection Details. Keep the connection type at the default SQLite, but instead of using the default locations, make sure that Filename points to your unpacked `cuchubo.db` file.

3.1.3 Ініціалізувати базу даних

Коли ви відкриваєте з'єднання з базою даних, яка не відмічена в Ghini перший раз, Ghini спочатку відобразить оповіщення:



відразу слідує запитання:

Будьте обережні, коли вручну, вказуєте параметри з'єднання: значення, які ви ввели і відіслали до існуючої бази даних, можуть бути не призначені для використання в

Ghini. Даючи Ghini ініціалізувати базу даних, база даних буде вичерпана, і весь її вміст буде втрачено.

Якщо ви впевнені, що хочете створити базу даних у цьому зв'язку, виберіть «Так». Ghini почне створювати таблиці бази даних та імпортувати дані за замовчуванням. Це може зайняти хвилину або дві, тоді як всі дані за замовчуванням імпортуються в базу даних, тому будьте терплячі.

Після того, як база даних була створена, налаштована,ініціалізована, ви готові до запуску *Редагування і вставка даних* і згодом *Пошук в Ghini*.

3.2 Пошук в Ghini

Пошук дозволяє представляти, перегляд і створювати звіти з ваших даних. Ви можете виконувати пошук ввівши запит в основному компоненті пошуку або використати Query Builder для створення запитів. Результати пошуку Ghini виводяться в головному вікні.

3.2.1 Стратегії пошуку

Ghini пропонує чотири різні стратегії пошуку:

- за значенням - у всіх доменах;
- за виразом - в декількох неявних полях в одній явній області;
- за запитом - в одному домені;
- за біномічним ім'ям - пошук лише в домені Вид.

Усі стратегії пошуку - за винятком пошуку біноміальних назв - нечутливі до регістру.

Пошук за значенням

Пошук за значенням - це найпростіший спосіб пошуку. Ви вводите одне або декілька рядків і дізнаєтеся, які збіги. Результат включає об'єкти будь-якого типу (домену), де один або декілька його полів містять одне або кілька пошукових рядків.

Ви не вкажете пошуковий домен, всі вони включені, а також ви не вкажете, які поля потрібно підібрати, це неявно в пошуковому домені.

Наступна таблиця допоможе вам зрозуміти результати та допоможе вам у формулюванні ваших пошуків.

огляд пошукового домену		
ім'я та короткі терміни	поле	тип результату
family, fam	epithet (family)	`Family`
genus, gen	epithet (genus)	`Genus`
species, sp	epithet (sp) ×	`Species`
vernacular, common, vern	ім'я	`Species`
geography, geo	ім'я	`Geography`
accession, acc	code	`Accession`
planting, plant	code ×	`Plant`
location, loc	code, name	`Location`
contact, person, org, source	ім'я	`Contact`
collection, col, coll	locale	`Collection`
tag, tags	ім'я	`Tag`

Приклади пошуку за значенням будуть: Maxillaria, Acanth, 2008.1234, 2003.2.1,indica.

Рядок пошуку розділений пробілами. Наприклад, якщо ввести в рядок пошуку `Block 10` то Ghini буде шукати рядки Block і 10 та поверне всі результати, які відповідають цим рядкам. Якщо ви хочете знайти безроздільно Block 10 у рядку, то ви повинні ввести в рядок пошуку, як цитату `"Block 10"`.

× **складні первинні ключі**

Епітет **вид** означає мало, без відповідного роду, також код **рослин** унікальний тільки в рамках приєднання, до якого вони належать. У теорії термінологія бази даних, епітету і коду недостатньо для формування **первинного ключа** для відповідних видів та посадок. Ці домени потребують первинного ключа **composite**.

Пошук по значеннях дозволяє шукати **посадки** за їх повним кодом посадки, який включає код приєднання. Взагалі, код приєднання та код посадки забезпечують **складний первинний ключ** для посадок. Для **виду** ми представили біноміальний пошук, описаний нижче.

Пошук за виразом

Пошук за допомогою виразу дає вам трохи більше контролю над тим, що ви шукаєте. Ви обмежуєте пошук до певного домену, програмне забезпечення визначає, які поля потрібно шукати в зазначеному вами домені.

Вираз будується як `<domain> <operator> <value>`. Наприклад, пошук: `gen=Maxillaria` поверне усі роди, які відповідають імені Maxillaria. У цьому випадку домен `gen`, оператор `=` і значення `Maxillaria`.

У наведеній вище таблиці огляду пошукових доменів вказані імена пошукових доменів, а також для кожного пошукового домену, які поля шукаються.

Рядок пошуку `loc like block%` повертає всі місця, для яких ім'я або код починається з «block». У цьому випадку домен є `loc` (скорочення для `location`), оператор є `like` (це походить з SQL і дозволяє «fuzzy» пошук) значення є `block%`, поля, що неявно співпадають `ім'я` і `код`. Знак відсотка використовується як динамічна карта, тому якщо ви шукаєте `block%`, то він шукає всі значення, що починаються з `block`. Якщо ви шукаєте `%10`, він шукає всі значення, які закінчуються `10`. Рядок `%ck%10` буде шукати всі значення, яке містить `ck` і закінчується на `10`.

Скільки часу потрібно на запит

Ви даєте запит, це вимагає часу для обчислення, результат містить необґрунтовано багато записів. Це трапляється, коли ви маєте намір використати стратегію, але ваші рядки не утворюють дійсного виразу. У цьому випадку Ghini повертається до *пошуку за значенням*. Наприклад, рядок пошуку `gen lik maxillaria` буде шукати рядки `gen`, `lik` і `maxillaria`, повертаючи всі, що відповідають хоча б одному з трьох критеріїв.

Біноміальний пошук

Ви також можете виконати пошук у базі даних, якщо ви знаєте цей вид, просто розмістивши в пошуковій системі декілька початкових букв епітетів роду та виду, правильно надруковані, тобто: **Епітет роду** з однією заголовною великою буквою, **Епітет видів ** всі літери.

Таким чином ви можете виконати пошук `So ha`.

Це були б ініціали для Solanum hayesii, або Solanum havanense.

Біноміальний пошук приходить до компенсації обмеженої корисності вищезгаданого пошуку за виразом при спробі шукати вид.

Це правильна капіталізація **Xxxx xxxx **, яка інформує програмне забезпечення про свій намір виконувати пошук у біноміалі. Друга помилка програмного забезпечення - пошук за значенням, що, можливо, призведе до набагато більшої кількості значень, ніж ви очікували.

Аналогічний запит `so ha` поверне в новій установці понад 3000 об'єктів, починаючи з сім'ї «Acalyp(**ha**)ceae», закінчуючи географією «Western (**So**)uth America».

Пошук по запиту

Запити дозволяють максимально контролювати пошук. За допомогою запитів ви можете здійснювати пошук по відношенню до певних стовпчиків, комбінуючи критерії пошуку за допомогою логічних операторів, таких як `and`, `or`, `not` (і їх скорочення `&&`, `||`, `!`) додайте їх у дужки, та багато іншого.

Будь ласка, зв'яжіться з авторами, якщо ви хочете отримати додаткову інформацію, або якщо ви добровільно документуєте це докладніше. Тим часом ви можете почати ознайомитися з основною структурою бази даних Ghini.

Fig. 1: **основна структура бази даних Ghini**

Кілька прикладів:

- насадження сімейства Fabaceae в місці розташування Блок 10:

```
plant WHERE accession.species.genus.family.epithet=Fabaceae AND location.
↪description="Block 10"
```

- місця, де немає рослин:

```
location WHERE plants = Empty
```

- приєднання, пов'язані з видом відомих біномічних назв (наприклад, Mangifera indica):

```
accession WHERE species.genus.epithet=Mangifera AND species.epithet=indica
```

- приєднання ми поширювали в 2016 році:

```
accession WHERE plants.propagations._created BETWEEN |datetime|2016,1,1|␣
↪AND |datetime|2017,1,1|
```

- приєднання, які ми змінили за останні три дні:

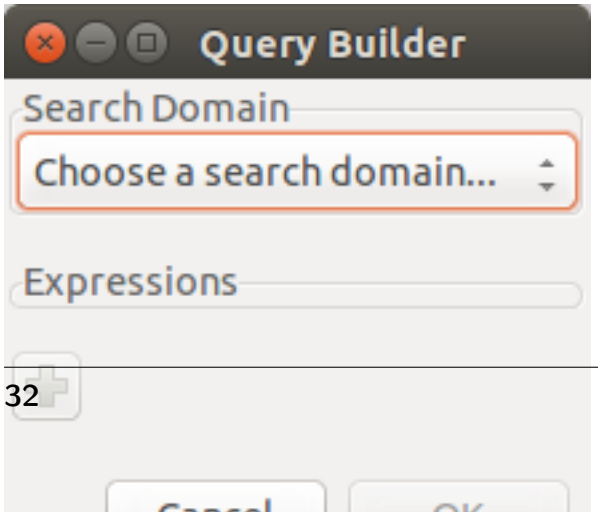
```
accession WHERE _last_updated>|datetime|-3|
```

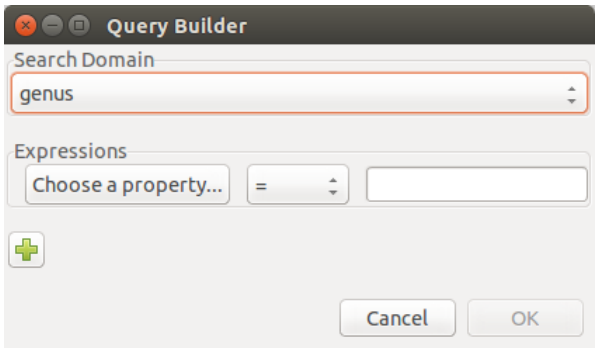
Пошук за запитами вимагає певних знань про маленький синтаксис та ідеї про велику структуру таблиці Ghini. Обидва ви придбаєте з практикою і за допомогою Query Builder.

3.2.2 Query Builder

Query Builder дозволяє будувати складні пошукові запити через вказування і натискання в інтерфейсі. Для того, щоб відкрити Query Builder натисніть  іконку зліва від входу пошуку або виберіть *Tools→Query Builder* в меню.

З'явиться вікно, яке веде вас до всіх кроків, необхідних для побудови правильного запиту, який розуміється в стратегії пошуку запиту Ghini.





Насамперед, ви вказуєте пошуковий домен, це дозволить Query Builder завершити свій графічний інтерфейс, після чого ви додасте стільки логічних статей, скільки потрібно, підключаючи їх `and` або `or` бінарним оператором.

Кожен розділ складається з трьох частин: властивості, які можна отримати з початкового домену пошуку, оператор зіставлення, який ви вибираєте з розкривного списку, значення, яке ви можете ввести або вибрати зі списку дійсних значень для поля.

Додайте якомога більше властивостей пошукових систем, натиснувши знак плюс. Виберіть та/або біля назви властивості, щоб вибрати, які об'єкти будуть об'єднані в пошуковому запиті.

Коли ви закінчите створення свого запиту натисніть кнопку OK, щоб виконати пошук.

На цьому етапі конструктор запитів записує запит у записі пошуку та виконує його. Тепер ви можете редагувати рядок так, ніби ви набрали його самостійно. Зверніть увагу, як значення інтер'єру лівої сторони інтерпретуються забудовником запиту та додаються в окремі лапки, якщо вони визнані як рядки, залишатись окремо, якщо вони виглядають як цифри або два зарезервовані слова `None` та `Empty`. Ви можете відредагувати запит і вставляти лапки, якщо вони вам потрібні, наприклад, якщо вам потрібно буквально шукати рядок `Empty`.

`None` - це значення порожнього поля. Це не те ж саме, що рядок нульової довжини `''`, ані числове `0`, ані логічне `False`, ані встановлення `Empty`, це означає, що поле не має значення Все.

`Empty` є порожнім набором. Будучи набором, він може бути узгоджений з наборами (наприклад: рослини приєднання або приєднання одного виду), не проти елементів (наприклад, кількість рослини або опис місця розташування). Тим не менш, Query Builder не дозволяє вибирати зупинку значень лівої сторони при встановленні, він сподівається, що ви виберете поле. Виберіть будь-яке поле: на момент створення запиту, коли Builder Query відповідає умовам з правою частиною значення буквеного

рядка `Empty`, він виведе поле назви і дозволить вам порівняти набір зліва з `Empty` справа.

У нас немає друкарських букв `False` і `True`. Це типи значення, і Query Builder не вміє їх випускати. Замість `False` введіть `0`, а замість `True` `1` тип.

3.2.3 Граматика запиту

Для тих, хто не боїться дещо формальної точності, наступний код BNF дає точне уявлення про граматику, яку реалізує Стратегія пошуку запитів. Деякі граматичні категорії визначаються неформально; будь-які зниклі з них залишаються у вашій родючій уяві; літери беруться в одинарні лапки; граматика в основному нечутлива, якщо не вказано інше:

```
query ::= domain 'WHERE' expression

domain ::= #( one of our search domains )
expression ::= signed_clause
             | signed_clause 'AND' expression
             | signed_clause 'OR' expression
             ;
signed_clause ::= clause
                | 'NOT' clause  #( not available in Query Builder)
                ;
clause ::= field_name binop value  #( available in Query Builder)
         | field_name set_binop value_list
         | aggregated binop value
         | field_name 'BETWEEN' value 'AND' value
         | '(' expression ')'
         ;
field_name ::= #( path to reach a database field or connected table )
aggregated ::= aggregating_func '(' field_name ')'
aggregating_func ::= 'SUM'
                   | 'MIN'
                   | 'MAX'
                   | 'COUNT'
                   ;
value ::= typed_value
        | numeric_value
        | none_token
        | empty_token
        | string_value
        ;
typed_value ::= '|' type_name '|' value_list '|'
numeric_value ::== #( just a number )
none_token ::= 'None'    #( case sensitive )
empty_token ::= 'Empty'  #( case sensitive )
string_value = quoted_string | unquoted_string
```

(continues on next page)

(continued from previous page)

```
type_name ::= 'datetime' | 'bool' ;  #( only ones for the time being )
quoted_string ::= '"' unquoted_string '"'
unquoted_string ::=  #( alphanumeric and more )

value_list ::= value ',' value_list
             | value
             ;
binop ::= '='
        | '=='
        | '!='
        | '<>'
        | '<'
        | '<='
        | '>'
        | '>='
        | 'LIKE'
        | 'CONTAINS'
        ;
set_binop ::= 'IN'
```

Будь ласка, пам'ятайте, що мова запиту Ghini є набагато складнішою, ніж те, що може створити Query Builder: запити, які ви можете побудувати за допомогою Query Builder, є правильною підмножиною запитів, визнаних програмним забезпеченням:

```
query ::= domain 'WHERE' expression

domain ::= #( one of our search domains )
expression ::= clause
             | clause 'AND' expression
             | clause 'OR' expression
             ;
clause ::= field_name binop value
         ;
field_name ::= #( path to reach a database field or connected table )
value ::= numeric_value
        | string_value
        ;
numeric_value ::== #( just a number )
string_value = quoted_string | unquoted_string ;

quoted_string ::= '"' unquoted_string '"'
unquoted_string ::=  #( alphanumeric and more )

binop ::= '='
        | '=='
        | '!='
        | '<>'
        | '<'
        | '<='
```

(continues on next page)

(continued from previous page)

```
        | '>'
        | '>='
        | 'LIKE'
        | 'CONTAINS'
        ;
```

3.3 Редагування і вставка даних

Основний спосіб, яким ми додамо або змінюємо інформацію в Ghini, - це використання редакторів. Кожен основний тип даних має свій власний редактор. Наприклад, є редактор сімейства, редактор роду, редактор приєднання тощо.

Для того, щоб створити новий запис натисніть на :menuselection:‘Insert‘в меню, і виберіть тип запису який ви хотіли б створити. Це відкриє новий порожній редактор типу.

Щоб змінити існуючий запис в базі даних натисніть правою кнопкою миші на елементі в результатах пошуку і виберіть *Edit* в спливаючому меню. Відкриється редактор, який дозволить вам змінити значення в записі, який ви вибрали.

Більшість видів є дочірніми, які можна додати, натиснувши правою кнопкою миші на батьківському виді і виберіть пункт «Додати ??? . . . » в контекстному меню. Наприклад, сім’я Genus дочірня: ви можете додати Genus до Сім’ї натиснувши правою кнопкою миші на сім’ї і вибравши пункт «Додати genus».

3.3.1 Нотатки

Майже у всіх редакторів Ghini є вкладка *Примітки* яка працюють однаково, незалежно від того, який редактор ви використовуєте.

Якщо ввести веб-адресу в примітці, то посилання буде відображатися у вікні Посилання, коли елемент вашого редагування обраний в результатах пошуку.

Ви можете переглянути нотатки для елемента в базі даних за допомогою вікна Нотатки в нижній частині екрана. Вікно Нотатки буде нечутливим, якщо обраний елемент не має записів.

3.3.2 Сім’я

Редактор сім’ї дозволяє додавати або змінювати ботанічну сім’ю.

Поле *Сім’я* у редакторі дозволяє змінити епітет сім’ї. Поле Сім’я обов’язкове.

Поле *Класифікатор* дозволяє вам змінити класифікатор сім’ї. Значення може мати *широке тлумачення*, *вузьке тлумачення* або ніякого.

Synonyms allow you to add other families that are synonyms with the family you are currently editing. To add a new synonyms type in a family name in the entry. You must select a family name from the list of completions. Once you have selected a family name that you want to add as a synonym click on the Add button next to the synonym list and the software adds the selected synonym to the list. To remove a synonym, select the synonym from the list and click on the Remove button.

Щоб відмітити зміни без збереження натисніть на кнопку *Скасувати*.

Для того, щоб зберегти сім'ю з якою ви працюєте, натисніть *OK*.

Для того, щоб зберегти сім'ю з якою ви працюєте і додати рід натисніть на кнопку *Додати рід*.

Для того, щоб додати ще одну сім'ю, коли ви закінчили редагування поточної натисніть на кнопку *Далі* внизу. Це дозволить зберегти поточну сім'ю і відкрити новий порожній редактор сім'ї.

3.3.3 Рід

Редактор родів дозволяє додавати або змінювати ботанічний рід.

Поле *Сім'я* в редакторі роду дозволяє вибрати сім'ю для роду. Коли ви починаєте вводити ім'я сім'ї він буде показувати список сімей, щоб вибрати з нього. Сім'я вже повинна існувати в базі даних, перш ніж ви можете встановити її в якості імені сім'ї для роду.

Поле *Рід* поле дозволяє встановити рід для цього запису.

Поле *Автор* дозволяє задати ім'я або абревіатуру автора (ів) для роду.

Synonyms allow you to add other genera that are synonyms with the genus you are currently editing. To add a new synonyms type in a genus name in the entry. You must select a genus name from the list of completions. Once you have selected a genus name that you want to add as a synonym click on the Add button next to the synonym list and it will add the selected synonym to the list. To remove a synonym select the synonym from the list and click on the Remove button.

Щоб відмітити зміни без збереження натисніть на кнопку *Скасувати*.

Для того, щоб зберегти рід з яким ви працюєте, натисніть *OK*.

Для того, щоб зберегти рід з яким ви працюєте і додати до нього вид натисніть на кнопку *Додати Вид*.

Для того, щоб додати ще один рід, коли ви закінчили редагування поточного натисніть на кнопку *Далі* в низу. Це дозволить зберегти поточний рід і відкрити новий порожній редактор роду.

3.3.4 Види/Таксон

З історичних причин називаються *види*, але ми маємо на увазі *таксон* в ранзі *видів* або нижче. Це являє собою унікальне ім'я в базі даних. Редактор видів дозволить вам створити ім'я, а також асоційовані метадані з таксоном, такі як розподіл, синоніми і інша інформація.

Внутрішньовидові частини в редакторі видів дозволять вам вказати додатково *таксон'ніж при 'види* рангу.

Щоб відмітити зміни без збереження натисніть на кнопку *Скасувати*.

Для збереження виду над яким ви працюєте натисніть *OK*.

Для збереження виду над яким ви працюєте і додаванням до нього приєднання натисніть на *Додати Приєднання*.

Для того, щоб додати ще один вид, коли ви закінчите редагування поточного натисніть на кнопку *Далі* внизу. Це дозволить зберегти поточні види і відкрити новий порожній редактор видів.

3.3.5 Приєднань

Редактор приєднання дозволяє додавати приєднання до певного виду. У Ghini приєднання представляє собою групу рослин або клонів, що мають один і той же таксон, мають один і той же тип ростка (або обробки), отримані з одного джерела, отримані одночасно.

Виберіть назву Таксона, додайте одну, якщо забули це зробити заздалегідь.

Ви можете відзначати невизначеність у ідентифікації, додавши класифікатор ідентифікації у відповідному ранзі, так що ви можете, наприклад, мати рослину, яка спочатку ідентифікувалась, як * Iris * cf. *florentina* вибравши *Iris florentina* в імені таксона, класифікатор ідентифікації „cf.“, кваліфікований ранг „species“.

Тип ID приєднання, бажано також Кількість отриманих.

Джерело приєднання

Джерело приєднань дає змогу додавати додаткову інформацію про те, звідки взяли приєднання. Виберіть контакт у розкривному списку або виберіть «Garden Propagation», який розміщений як перший пункт за замовчуванням в списку контактів.

Garden Propagation є результатом успішного розповсюдження.

При доступі до матеріалу з Garden Propagation, ви спочатку залиште першу вкладку (Загальні) і почніть з другої вкладки (Джерело). Виберіть в якості контакту «Garden Propagation», вкажіть, яка рослина є материнською рослиною і виберіть серед ще не повністю доступних розповсюджень ту, яку ви плануєте додати як приєднання до вашої бази даних.

Після вибору розповсюдження програмне забезпечення встановить декілька полів на вкладці Загальні, які ви можете переглянути. Таксон (можливо, вам вдалося отримати щось дещо інше, ніж материнську рослину). Початкова кількість (у випадку, якщо не всі рослини переходять до одного приєднання). Тип матеріалу, що випливає з типу розповсюдження.

3.3.6 Рослина

`Plant` в базі даних Ghini описує окрему рослину у вашій колекції. Рослина належить до приєднання, і вона має своє місце розташування.

Створення декількох рослин

Ви можете створити кілька рослин з використанням діапазонів в запису коду. Це дозволено тільки при створенні нових рослин, і це не можливо при редагуванні існуючих рослин в базі даних.

Наприклад, діапазон, 3-5 створить рослину з кодом 3,4,5. Діапазон 1,4-7,25 створюватиме рослини з кодами 1,4,5,6,7,25.

При вході в діапазон в запису коду рослин запис буде синіти щоб вказати, що ви зараз створюєте кілька рослин. Будь-які поля, які встановлені в цьому режимі будуть скопійовані для всіх рослин які створюються.

Зображення

Так само, як майже всі об'єкти в базі даних Ghini можуть мати пов'язані з ними *Примітки*, рослини можуть мати *Зображення*: поруч із закладкою Примітки, редактор рослин містить додаткову вкладку під назвою «Зображення». Ви можете пов'язати стільки зображень, скільки може знадобитися для рослини та виду об'єкта.

Коли ви асоціюєте зображення до об'єкта, файл копіюється в теку *pictures*, а мініатюра (500x500) створюється та копіюється в теку *thumbnails* всередині теки зображень.

З Ghini-1.0.62, зображення не зберігаються в базі даних. Для забезпечення доступності зображень на всіх терміналах, де ви встановили і налаштували Ghini, ви можете використовувати мережний диск або службу спільного використання файлів Tresorit або Dropbox.

Пам'ятайте, що ви налаштували кореневу теку зображень, коли вказали деталі налаштування підключення до бази даних. Знову ж таки, ви повинні переконатися, що коренева тека зображень доступна для спільного використання файлів сервіс вибору.

Коли рослину або вид в поточному відборі буде виділено, її зображення відображається в панелі зображень, панель зліва від інформаційної панелі. При приєднанні вибір підсвічуються, будь-яке зображення пов'язане з рослинами в підсвічених приєднаннях відображається в панелі зображень.

У Ghini-1.0 зображення є спеціальними нотатками, з категорією «<picture>» і текстом шляху до файлу, відносно кореневої теки зображень. На вкладці Нотатки, нотатки з зображеннями відображатимуться як звичайні нотатки, і ви можете редагувати їх без обмежень.

Рослина є фізичним об'єктом, тому ви можете приєднати до неї зображення, взяті цієї індивідуальної рослини, взяті на будь-якій відповідній стадії розвитку рослини, можливо, допомагаючи її ідентифікувати.

Species are abstract objects, so you would associate to it pictures showing the characteristic elements of the species, so it makes sense to associate a flora illustration to it. You can also do that by reference: go to the Notes tab, add a note and specify as category «<picture>», then in the text field you type the URL for the illustration of your choice.

3.3.7 Розташування

Редактор Розташування

небезпечна зона

Редактор розташування містить спочатку прихований розділ під назвою *небезпечна зона*. Віджети, що містяться в даному розділі, дозволяють користувачеві переводити поточне місце розташування в інше місце, виправляти орфографічні помилки або здійснювати зміну політики.

3.4 Робота з розповсюдженням

Ghini пропонує можливість асоціювати дослідження розповсюдження на рослини та документувати їх трактування та результати. Трактування є невід'ємною частиною опису розповсюдження. Якщо дослідження про розповсюдження є успішним, Ghini дозволяє пов'язати його з новим приєднанням. Ви можете приєднати лише одне приєднання до пробної версії.

Тут ми описуємо, як ви використовуєте цю частину інтерфейсу.

3.4.1 Створення розмноження

Розмноження (випробування) отримують із рослини. Ghini відображає це у своєму інтерфейсі: вибираєте рослину, відкривається редактор рослин, активуєте вкладку розповсюдження, натискаєте Додати.

Коли ви робите це вище, ви отримаєте вікно редактора розповсюдження. Ghini не вважає, що дослідження про розповсюдження є незалежними органами. В результаті, Ghini розглядає редактор розповсюдження як спеціальне вікно редактора, який ви можете отримати тільки з редактора рослин.

Для нового розповсюдження ви вибираєте тип розповсюдження (це стає незмінним властивістю розповсюдження), потім вставляйте дані, що описують його.

Ви зможете редагувати дані про розповсюдження одним шляхом: виберіть рослину, відкрийте редактор рослин, ідентифікуйте поширення, яке ви хочете редагувати, натисніть на відповідну кнопку Редагувати. Ви зможете редагувати всі властивості існуючого процесу розповсюдження, крім його типу.

У випадку пробного розмноження насіння, у вас є батьківський пилок, і батьківське насіння. Завжди слід зв'язувати розповсюдження проб до батьківського насіння.

Примітка: У Ghini-1.0 ви вказуєте пилок батьківської рослини в поле «Примітки», у той час як Ghini-1.1 має (відношення) поле для нього. За інформацією ITF2, можуть бути випадки випробувань на розмноження де невідомо, яку роль відіграє яке насіння . Знову ж таки, в Ghini-1.0 ви повинні використовувати примітку, щоб вказати, чи це так, Ghini-1.1 має (булеве) поле, що вказує, чи це так.

3.4.2 Використання розповсюдження

Процес розповсюдження може бути успішним і призвести до нового приєднання.

Ghini допоможе відобразити це в базі даних: створити нове приєднання, негайно перейдіть на вкладку Джерело і виберіть «Розповсюдження саду» в (правда, дещо помилково) контактному полі.

Почніть вводити номер рослини, і ви побачите список відповідних рослин із випробуваннями для розповсюдження.

Виберіть рослину, а список доступних і недоступних випробувань поширення з'явиться в нижній частині вікна.

Виберіть незадовільну пробну версію для розповсюдження зі списку та натисніть кнопку OK, щоб завершити операцію.

Використовуючи дані, отримані в процесі розповсюдження, Ghini завершує деякі поля на вкладці Загальні: назва Таксона, Тип матеріалу та, можливо, Походження. Ви зможете редагувати ці поля, але будь ласка, зверніть увагу, що програмне забезпечення не заважатиме введення концептуальних невідповідностей у вашій базі даних.

Ви можете приєднати пробну версію до одного приєднання.

3.5 Тегування

Тегування - це простий спосіб надати контекст об'єкту або створити колекцію об'єктів, які ви хочете згадати пізніше.

Потужність цієї дії з тегами полягає в тому, що ви можете поділитися цим вибором із колегами, які можуть діяти на ньому, без необхідності повторювати всю вашу колективну роботу.

Наприклад, якщо вам потрібно надрукувати прикріплені етикетки інших не пов'язаних рослин, ви можете об'єднати об'єкти, позначивши їх рядком «relabel». Ви або один з ваших колег може вибрати «relabel» з меню тегів, у вікні пошуку відображатимуться всі об'єкти, які ви позначили, а виконання звіту буде діяти на позначених об'єктах.

Позначення діє на активний вибір, тобто елементи в результатах пошуку, які ви вибрали.

Будь ласка, пам'ятайте: ви можете вибрати всі рядки результатів, натиснувши `Ctrl-A`, ви можете скасувати вибір, натиснувши `Ctrl-Shift-A`, ви можете змінити позначення одного рядка за допомогою ‘‘Ctrl-Клік мишкою ‘на ньому.

Після активного вибору, тегування може здійснюватися двома способами:

3.5.1 діалогове вікно тегування

Натисніть клавішу `Ctrl-T` або виберіть *Tag→Tag Selection* з меню, це активує вікно, в якому ви можете створювати нові теги і застосовувати будь-який існуючий тег до вибору.

Вікно тегів складається з трьох частин:

1. У верхній частині згадується список об'єктів у вашому активному виборі. Це список об'єкта, якому ви редагуєте теги;
2. У середній частині є список всіх доступних тегів, а також прапорець, який ви можете активувати для застосування тегу або вилучення тегу з вибору;
3. Нижня частина містить лише посилання на створення нового тегу, а кнопка OK для закриття діалогового вікна.

Якщо, відкриваючи діалогове вікно тегу, активний вибір містить кілька елементів, тоді біля нього буде встановлено лише теги, які є загальними для всіх вибраних елементів. Теги, які застосовуються лише до відповідної підмножини активного вибору, відображатимуться з статусом „не визначено“. Теги, які не застосовуються до жодного об'єкта в активному виборі, будуть відображатися пустими.

Нещодавно створений тег або найновіший вибраний тег стає активним тегом, і він з'являється з позначкою поруч із ним в меню тегів.

3.5.2 тегування без вікна

Після того як у вас є активна тег, натискання клавіші ‘‘ Ctrl-Y‘‘ застосовує активний тег до всіх об'єктів активного вибору. *Ctrl-Shift-Y‘* видаляє активний тег з усіх об'єктів у активному виділенні.

3.6 Створення звітів

A database without exporting facilities is of little use. Ghini lets you export your data in table format (open them in your spreadsheet editor of choice), as labels (to be printed or engraved), as html pages or pdf or postscript documents.

3.6.1 The Report Tool

You activate the Report Tool from the main menu: *Tools→Report*. The Report Tools acts on a selection, so first select something, then start the Report Tool.

Report on the whole collection.

> To produce a report on your whole plant collection, a shortcut would be from the home screen, to click on the `Families: in use` cell.
>
> If your focus is more on the garden location than on taxonomy and accessions, you would click on the `Locations: total` cell.

Reports are produced by a report engine, making use of a report template. Ghini relies upon two different report engines (Mako & XSL), and offers several report templates, meant as usable examples.

Choose the report you need, specify parameters if required, and produce the report. Ghini will open the report in the associated application.

Configuring report templates, that's a task for who installs and configures ghini at your institution. Basically, you create a template name, indicating the report engine and specifying the template. Configured templates are static, once configured you are not expected to alter them. Only the special `**scratch**` template can be modified on the fly.

The remainder of this page provides technical information and links regarding the formatter engines, and gives hints on writing report templates. Writing templates comes very close to writing a computer program, and that's beyond the scope of this manual, but we have hints that will definitely be useful to the interested reader.

3.6.2 Використання форматера звіту Mako

Форматер звіту Mako використовує мову шаблонів Mako для генерації звітів. Детальнішу інформацію про Мако і його мову можна знайти на сайті makotemplates.org.

Система шаблонів Mako вже повинна бути встановлена на вашому комп'ютері, якщо встановлено Ghini.

Створення звітів за допомогою Мако аналогічна тому, як ви можете створити вебсторінку за допомогою шаблону. Це набагато простіше, ніж XSL Formatter (див нижче) і відносно легко буде створити шаблон для тих, хто хоча б з невеликим досвідом програмування.

Генератор шаблону буде використовувати те саме розширення файлу, як шаблон, який повинен вказати тип виводу шаблону при створенні. Наприклад, щоб створити HTML-сторінку з вашого шаблону, ви маєте називати шаблон так: *report.html*. Якщо шаблон буде генерувати файл зі значеннями, розділеними комами, слід назвати шаблон *report.csv*.

Шаблон буде отримувати змінну *values*, яка буде містити список значень в поточному пошуку.

Тип кожного значення в *values* буде таким же, як область пошуку, що використовується в пошуковому запиті. Для отримання додаткової інформації про пошукові домени дивитися: search-domains.

Якщо запит не має пошукового домену, то всі значення можуть бути різного типу і шаблон Мако повинен бути готовий впоратися з ними.

3.6.3 Використання XSL Report Formatter

XSL форматер звітів вимагає представляти XSL в PDF для перетворення даних в PDF файл. Apache FOP є безкоштовним і з відкритим вихідним кодом XSL->PDF визуализатор що рекомендується.

При використанні Linux, Apache FOP повинен бути встановлений за допомогою менеджера пакетів. В Debian/Ubuntu це встановлювана як `fop` в Synaptic або за допомогою наступної команди

```
apt-get install fop
```

Установка Apache FOP на Windows

У вас є два варіанти установки FOP на Windows. Найпростіший спосіб це завантажити скомпільований ApacheFOP-0.95-1-setup.exe installer.

В якості альтернативи ви можете скачати archive. Після вилучення архіву ви повинні додати каталог, розпакував архів в змінну PATH середовища.

3.7 Імпорт та експорт даних

Хоча Ghini може бути розширена за рахунок плагінів для підтримки альтернативних форматів імпорту та експорту, за замовчуванням він може тільки імпорт і експорт розділених комою значень файлів або CSV.

Існує деяка підтримка експорту в Access для біологічних колекцій даних але вона обмежена.

Існує також обмежена підтримка експорту в форматі XML, який більш-менш точно відображає таблиці та рядки бази даних.

Експорт ABCD і XML не будуть тут розглядатися.

Попередження: Імпорт файлів, швидше за все, знищить будь-які дані у вашій базі даних, переконайтеся, що ви створили резервну копію даних.

3.7.1 Імпорт з CSV

Загалом, краще імпортувати CSV файли в Ghini, які раніше були експортовані з Ghini. Можна імпортувати будь-який файл CSV, але це більше, чим цей документ буде охоплювати.

Щоб імпортувати CSV файли в Ghini виберіть: menuselection *Tools→Export→Comma Seperated Values* з меню.

Після натискання OK в діалоговому вікні запиту, якщо ви впевнені, що знаєте, що робите, вибрані файли будуть відкриті. У файловому браузері виберіть файли, які ви хочете імпортувати.

3.7.2 Експорт в CSV

Щоб експортувати дані Ghini в CSV виберіть *Tools→Export→Comma Seperated Values* з меню.

Цей інструмент запропонує вам вибрати каталог для експорту даних CSV. Всі таблиці в Ghini будуть експортовані до файлів у форматі tablename.txt, де tablename - це назва таблиці, з якої дані було експортовано.

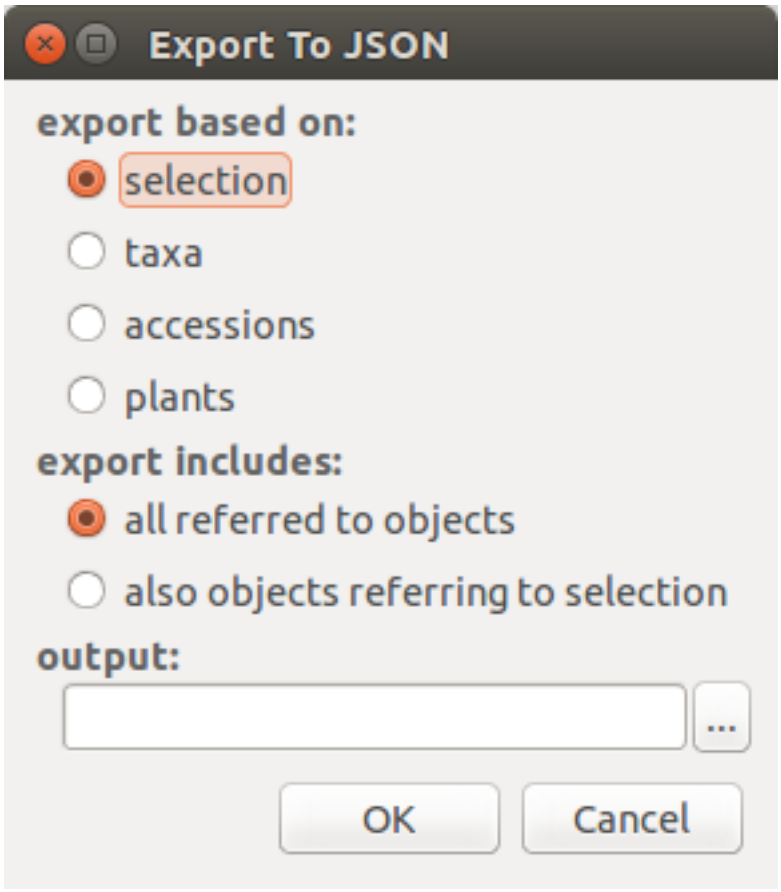
3.7.3 Імпорт з JSON

Цей *the * спосіб для імпорту даних в існуючу базу даних, не руйнуючи попередній вміст. Типовий приклад цієї функції буде імпорт вашої цифрової колекції в нові, тільки розміщені бази даних Ghini. Перетворення бази даних в bauble json формат обміну виходить за рамки даного керівництва, будь ласка, зв'яжіться з одним з авторів, якщо вам потрібна додаткова допомога.

Використовуючи формат обміну Ghini json, ви можете імпортувати дані, які ви експортували з іншої установки Ghini.

3.7.4 Експорт у JSON

Ця функція знаходиться в стадії розробки.



коли ви активуєте цей інструмент експорту, вам надається право вибору експорту. Ви можете використовувати поточний вибір щоб обмежити діапазон експорту, або ви можете почати з повним змістом домену, вибравши серед видів, приєднань, рослин.

Експорт *Вид* буде експортувати тільки повну класифікаційну інформацію в вашу базу даних. «Приєднання» буде експортувати всі приєднання плюс всю класифікаційну інформацію яка відноситься: необроблена для класифікації інформація не буде експортуватися. *Рослини* буде експортувати всі існуючі рослини (деякі приєднання можуть бути невключені), всі зазначені місця і класифіковані.

3.7.5 Імпортування з загальної бази даних

This functionality is the object of issue #127, for which we have no generic solution yet.

If you're interested in importing data from some flat file (e.g.: Excel spreadsheet) or from any database, contact the developers.

3.7.6 Importing a Pictures Collection

We can consider a collection of plant pictures as a particular form of botanical database, in which each picture is clearly associated with one specific plant.

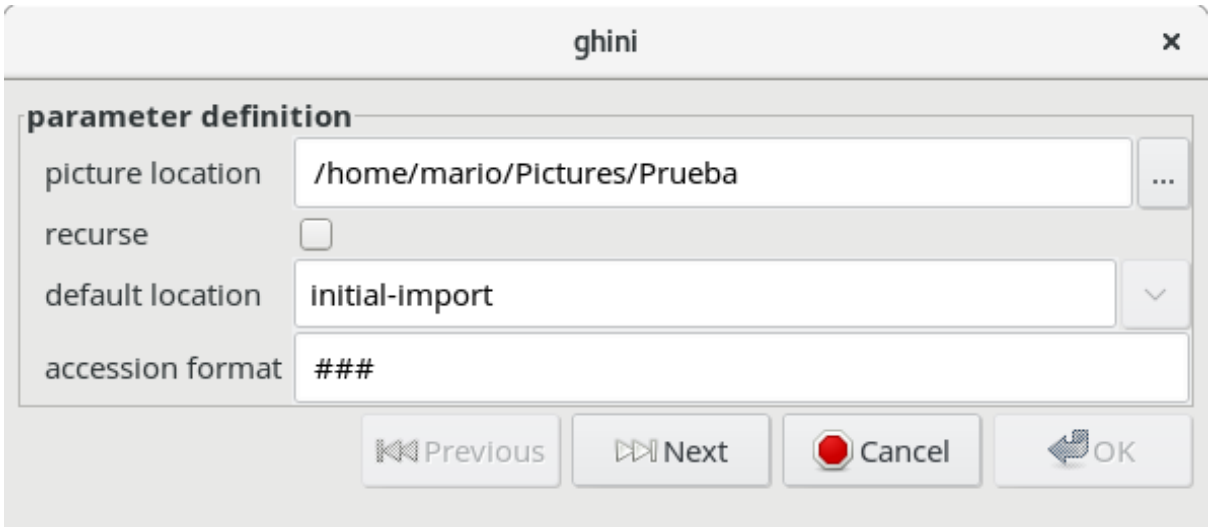
Even without using a photo collection software, you can associate pictures to accessions by following one and the same clear rule when naming picture files.

For example, `2018.0020.1 (4) Epidendrum.jpg` would be the name of the fourth picture for plant number 1 within accession 2018.0020, identified to rank genus as an Epidendrum.

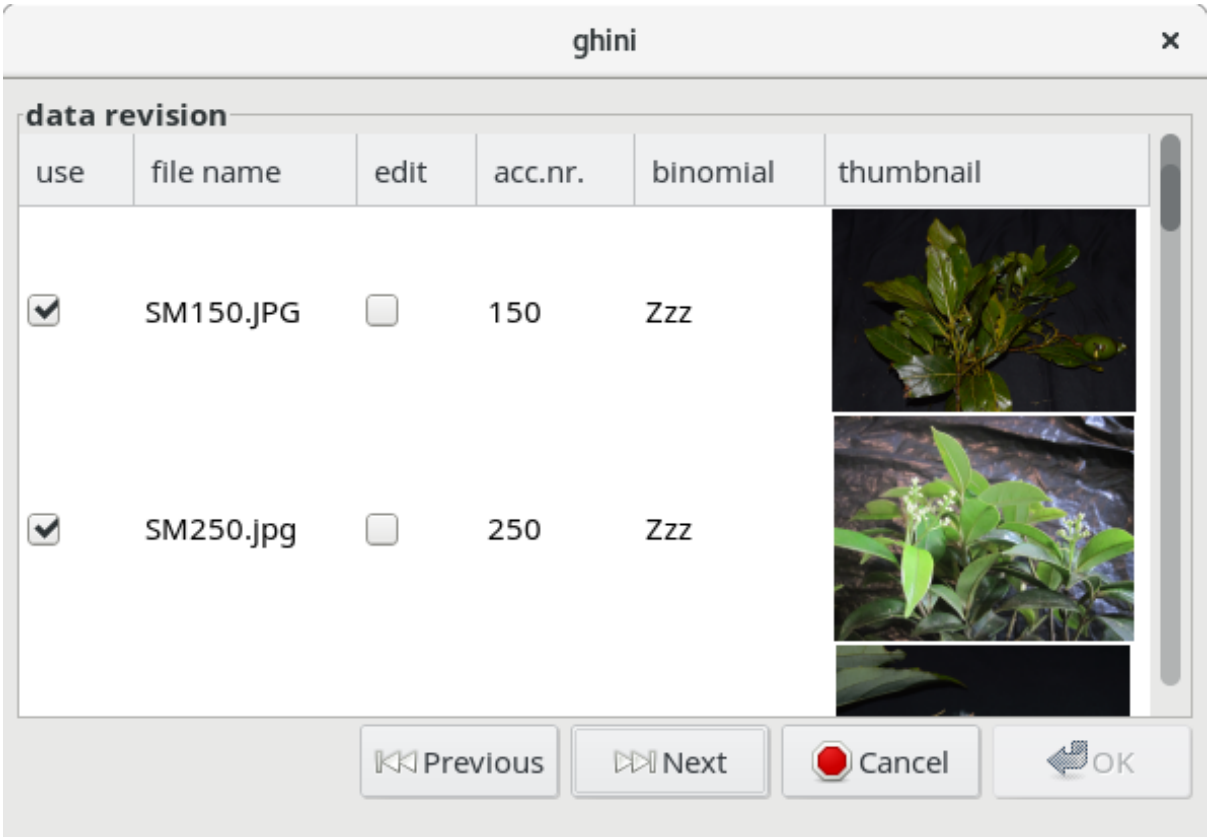
The *Tools→Import→Pictures* functionality here described is meant for importing an ordered collection of plant pictures either to initialize a ghini database, or for periodically adding to it.

Use *Tools→Import→Pictures* to activate this import tool. Import goes in several steps: parameter definition; data revision and confirmation; the import step proper; finally review the import log. At the first two steps you can confirm the data and go to the next step by clicking on the `next` button, or you can go back to the previous step by clicking on the `prev` button. Once the import is done and you're reviewing the log, you can only either confirm —or abort— the whole transaction.

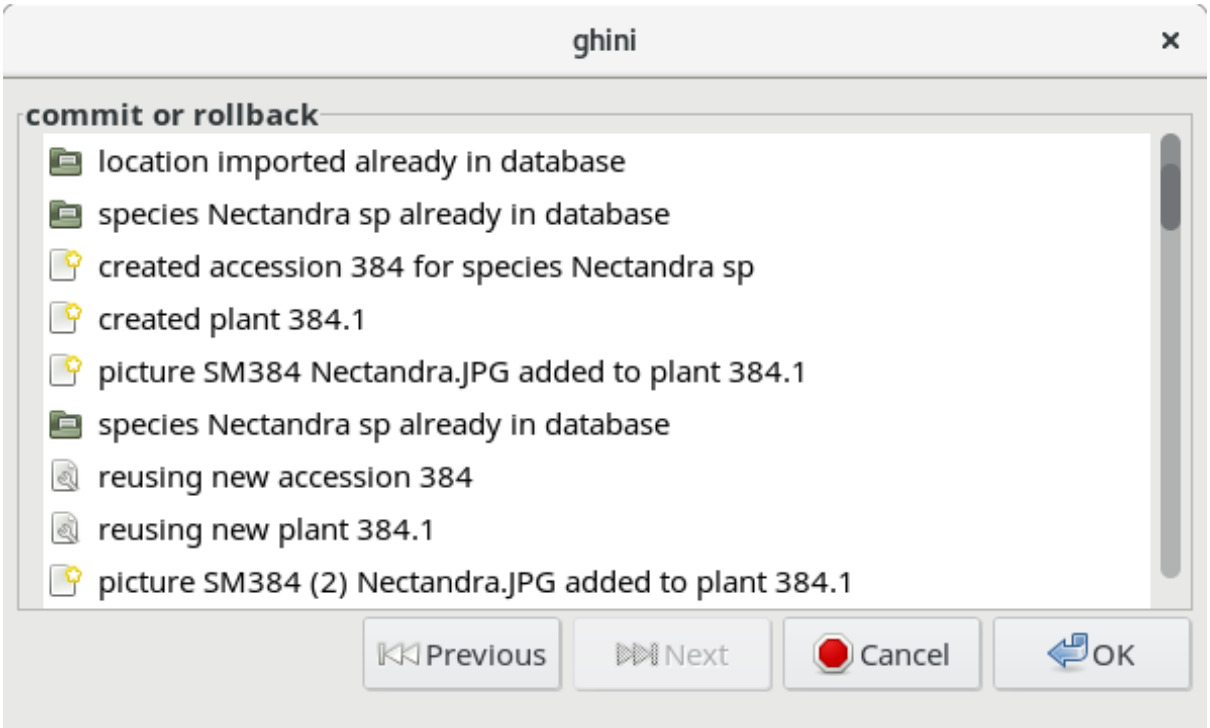
In the «parameter definition» pane you: select the directory from which you intend to import pictures; indicate whether to import pictures recursively; select or create a location which will be used as default location for new plants; inform the tool about the rule you've been following when naming picture files.



In the «data revision» pane you are shown a table with as many rows as the pictures you are importing. Each row holds as much information as the tool managed to extract from the picture name. You can review the information, correct or confirm, and indicate whether or not the row should be imported.



In the final «commit or rollback» pane you read the logs relative to your data import, and decide whether to keep them (commit them to the database), or undo them (rollback the transaction).



When the Picture Collection importer creates or updates objects, it also sets a Note that you can use for selecting the objects involved in the import, and for reviewing if needed.

3.8 Управління користувачами

Примітка: Плагін користувачів Ghini доступний лише на базі даних PostgreSQL.

The Ghini Users Plugin will allow you to create and manage the permissions of users for your Ghini database.

Ви повинні входити до своєї бази даних як користувач із привілеєм `CREATEROLE` для керування іншими користувачами.

3.8.1 Створення користувачів

Для створення нового користувача ...

3.8.2 Дозволи

Ghini дозволяє читати, писати і оформляти дозволи.

Розділ 4

Довідник програмних рішень

4.1 Внесені рецепти колекції

На цій сторінці представлені списки випадків використання. Якщо ви шукаєте пряму, практичну інформацію, ви знаходитесь у потрібному місці. Якщо ви віддаєте перевагу докладній презентації структури програмного забезпечення та бази даних, перегляньте розділ програмне забезпечення для ботанічних садів

Весь матеріал тут був внесений садами, використовуючи програмне забезпечення та обмін досвідом для користувачів.

Автори програмного забезпечення хочуть всім щиро подякувати.

4.1.1 Ботанічний сад Кіто

У квітні 2015 року у JBQ, Ботанічний сад Кіто, ми прийняли програмне забезпечення Ghini. З того часу ми накопичили досвід роботи з програмою, і ми самі повинні його документувати, щоб забезпечити знання в установі. Ми з радістю поділяємо це.

Технічний

- Ми працюємо над GNU/Linux, платформою, якою багато користувачів не оволодіють, і наша база даних знаходиться всередині віддаленої системи управління базами даних. Це означає кроки, які не є очевидними для звичайних кінцевих користувачів.

  Як запустити програму

Щоб запустити програму із зазначенням її назви, натисніть **|loose_png | Клавіша поруч з Alt або клацніть на |10000000000000300000002F89E0224ADF9EC09E_png|**, після чого почніть вводити назву програми, у нашому випадку “Ghini” або просто натиснути символ програми  , що з’являється біля лівих крапок вашого дисплея.

Сервер баз даних

Ми обрали для централізованого сервера баз даних PostgreSQL. Таким чином, ми захищені від одночасних конфліктуючих змін, і всі зміни одночасно доступні для всіх клієнтів ghini. Нам потрібно було використовувати аутсорсинг керування сервером бази даних.

додавання нового користувача

Ghini відстежує користувача, який виконує всі види редагування в базі даних, і в саду, крім постійних користувачів, у нас є всілякі тимчасові користувачі, які пишуть в базу даних, і ми вирішили дозволити Ghini допомогти нам стежити за подіями бази даних.

Оскільки ми працюємо з PostgreSQL, користувачі, яких зберігає Ghini в історії бази даних, є користувачі бази даних, а не користувачі системи.

Кожен користувач знає свій пароль, і тільки він знає його. Наш суперкористувач, який відповідає за вміст бази даних, також має `bauble` вигаданий пароль користувача, який ми використовуємо лише для створення інших користувачів.

Ми не використовуємо назви облікових записів, як `voluntario`, оскільки такі облікові записи не допомагають нам пов’язувати ім’я з особами.

— додавання нового користувача системи (linux/osx)

Додавання системного користувача не є обов’язково необхідним, оскільки ghini не використовує його в журналах, однак додавши, що системний користувач дозволяє розділити налаштування, налаштовані з’єднання, історію пошуку. У деяких наших системах ми маємо єдиний загальний обліковий запис з кількома налаштованими з’єднаннями, в інших системах ми маємо один обліковий запис для кожного користувача.

На системах з одним обліковим записом для кожного користувача наші користувачі мають одне налаштоване з’єднання, і ми зберігаємо

пароль бази даних у файлі /home/<account>/.pgpass. Цей файл є лише читабельним для власника <account>.

У системах із загальним обліковим записом користувач повинен вибрати своє власне з'єднання та ввести відповідний пароль.

Ось кроки для додавання користувачів системи:

```
sudo -k; sudo adduser test
sudo adduser test adm; sudo adduser test sudo
sudo adduser test sambashare; sudo adduser test ghini
```

— **додавання нового користувача бази даних**

Ghini має дуже мінімальний інтерфейс для управління користувачами, він працює тільки з postgresql, і його дуже не вистачає. Ми відкрили питання, які дозволять нам використовувати його, поки що ми використовуємо сценарій create-role.sh:

```
#!/bin/bash
USER=$1
PASSWD=$2
shift 2
cat <<EOF | psql bauble -U bauble $@
create role $USER with login password '$PASSWD';
alter role $USER with login password '$PASSWD';
grant all privileges on all tables in schema public to $USER;
grant all privileges on all sequences in schema public to $USER;
grant all privileges on all functions in schema public to $USER;
EOF
```

Надлишкова alter role після " create role" дозволяє застосовувати один і той же сценарій для виправлення існуючих облікових записів.

Наша база даних ghini називається bauble, і bauble також ім'я суперкористувача нашої бази даних, єдиного користувача з CREATEROLE привілеєм.

Наприклад, наступний виклик створить користувача willem з паролем orange у базі bauble, розміщених на 192.168.5.6:

```
./create-role.sh willem orange -h 192.168.5.6
```

- Розуміння того, коли оновлювати

Оновлення системи

Оновлення Ubuntu набагато легше і простіше, ніж у Windows. Тому коли система пропонує оновлення, ми дозволимо це зробити. Як пра-

вило, не потрібно чекати під час оновлення, ані перезавантажувати після завершення.

Оновлення ghini

Перше вікно, представлене Ghini, виглядає так. Зазвичай у цьому вікні вам нічого не потрібно робити, просто натисніть клавішу Enter і потрапите у головний екран програми.

Іноді у верхній частині екрана з'являється інформаційний текст, в якому повідомляється, що нова версія доступна в режимі онлайн.

Процедура оновлення проста, і це залежить від використовуваної вами операційної системи, ми не пояснюємо тут ще раз.

Як правило, хороша ідея оновлення програмного забезпечення. Якщо у вас сумніви, зв'яжіться з автором або напишіть у групу.

- розуміння початкового екрану ghini

Повний екран

На момент написання наш початковий екран виглядав так:

Крім основного меню додатків, Ghini пропонує три розділи спеціального інтерфейсу з інформацією та інструментами для вивчення бази даних.

Числовий огляд

Таблиця у правій половині екрана, яка містить зведення всіх зареєстрованих рослин, можна спостерігати. Кожна запис, надрукований жирним шрифтом, є посиланням на запит, вибравши відповідні об'єкти.

	total	in use	unused
Families:	511	7	504
Genera:	25394	158	25236
Species:	637	623	14
Accessions:	7722	7675	47
Plants:	7676	7676	0
Locations:	170	163	7

Збережені запити

Нижня половина правої сторони містить набір збережених запитів. Хоча ви можете редагувати їх за вашим бажанням, наші підказки включають вибір тих приєднань, які не були ідентифіковані за кате-

горіями видів. І один для історії бази даних.

Кнопки запиту та дії

У верхній частині цього екрана ви можете знайти поле, в яке потрібно ввести ваші пошуки.

– З кнопки, у вигляді будинку, ви можете повернутися з ваших пошуків на головний екран.

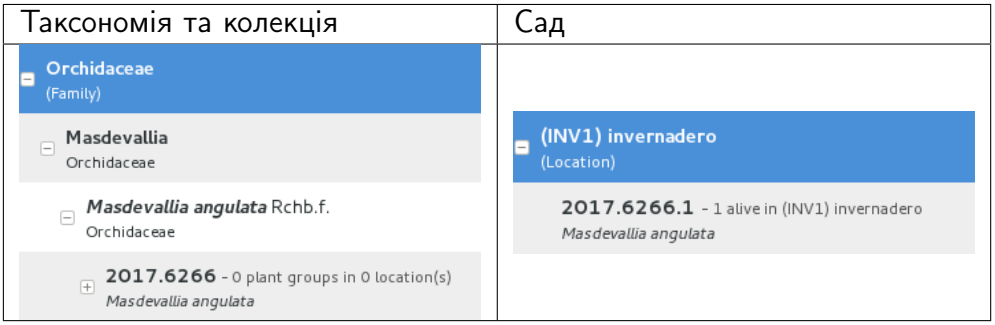
– З кнопки, у формі стрілки, ви можете повернутися до вашого останнього пошуку.

– З кнопки, у формі передач, ви можете запустити «Query Builder», який допоможе вам скласти складні пошуки в простий, графічний спосіб.

- У нас часто є добровольці, які працюють тільки в саду дуже короткий час. Саме це маючи на увазі, ми розробили hyper-simplified view на Ghini структури бази даних.

Деталі

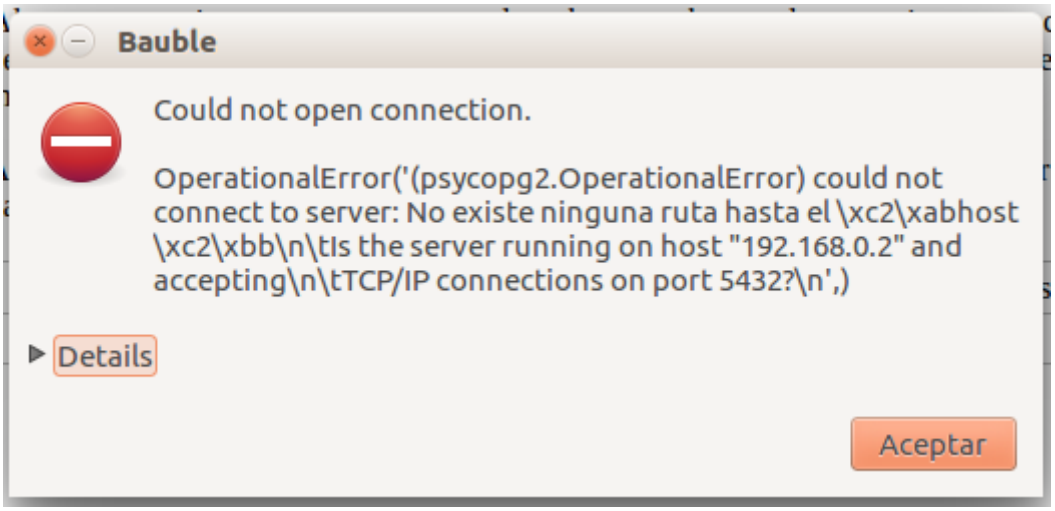
Дві цифри тут показують все, що наші тимчасові співробітники повинні знати.



- Іноді програма повідомляє про помилку. DON'T PANIC , повторіть спробу або повідомте розробникам.

Мережеві проблеми

Для роботи програма потребує стабільного мережевого підключення до сервера баз даних. Це може статися: ви запускаєте програму, і вона не може підключитися до нашого сервера баз даних. Потім ви отримаєте досить явне, але дуже погано набираєме повідомлення про помилку.

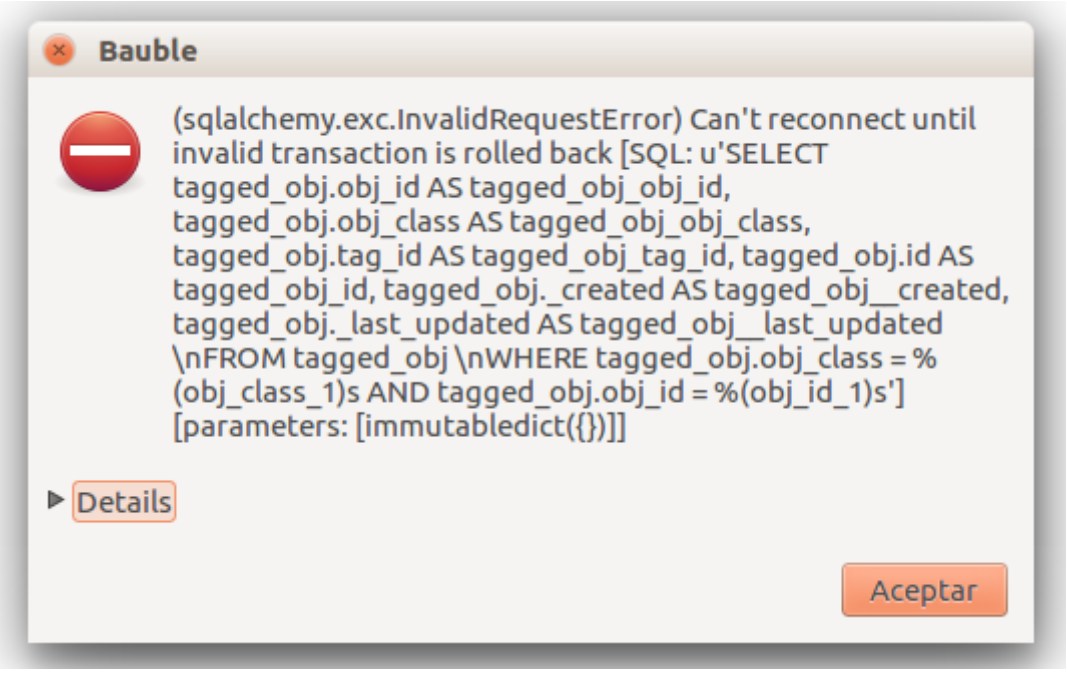


Просто ігноруйте його і повторіть спробу.

Пошук не вдається з помилкою

Іноді і без будь-якої очевидної причини пошук не буде успішно виконуватися, і з'явиться вікно з повідомленням про помилку. У цьому випадку вам доведеться спробувати виконати той самий пошук ще раз.

Приклад такого повідомлення про помилку:



Пошук не повертає те, що я щойно вставив

Коди приєднання, що починаються з нуля і складаються з чисел, наприклад, `016489` розглядаються програмним забезпеченням як числа, тому якщо ви не вкладете рядок пошуку в лапки, будь-який провідний 0 буде знятий і значення буде не знайдено.

Спробуйте ще раз, але додайте свій пошуковий рядок в одинарні або подвійні лапки.

Номер на етикетці	відповідний пошук
16489	„016489“

Будь ласка, запримітьте: коли ви шукаєте код рослини, а не приєднання, провідний нуль стає необов'язковим, тому в наведеному вище прикладі може бути простіше вводити `16489.1`.

- Серйозна ситуація трапляється один раз, і ми абсолютно хочемо запобігти цьому: знову користувач видалив рід, все, що було нижче, види і приєднання та синоніми.

Вирішення з правами користувачів

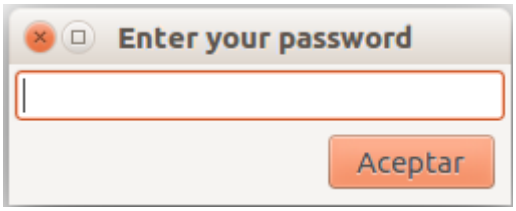
Ми пропонуємо мати різні профілі з'єднання, пов'язані з різними користувачами баз даних, кожен користувач із усіма необхідними правами доступу.

Повний дозвіл (BD-JBQ) Тільки кваліфікований персонал отримує такий доступ.

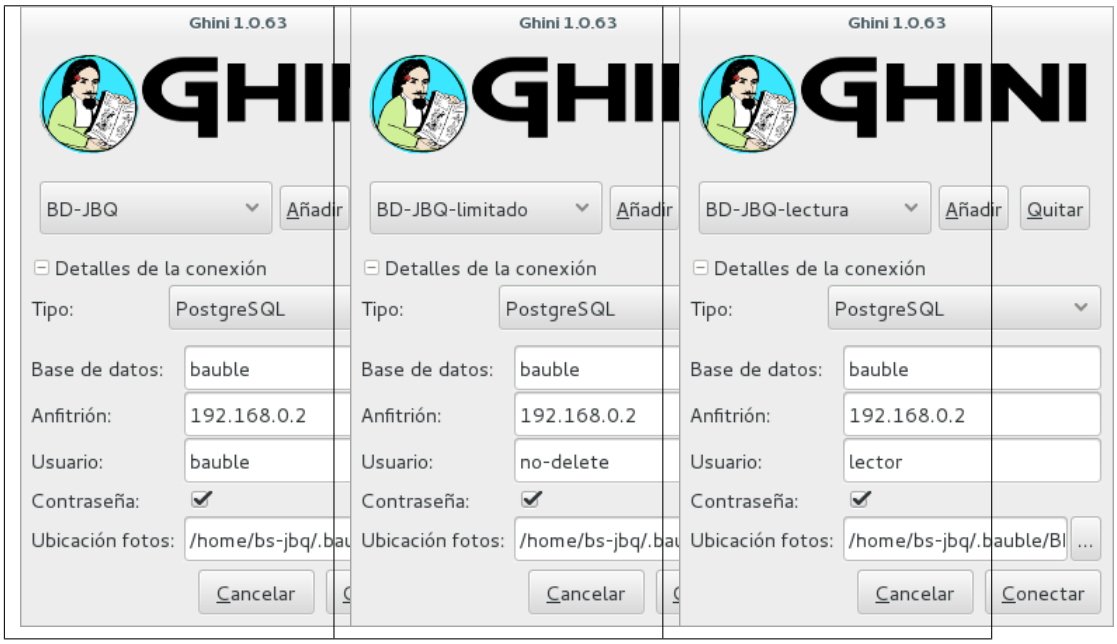
Вставити та оновити (BD-JBQ-limitado) Ми використовуємо цю програму для тих користувачів, які нам допомагають протягом обмеженого часу, і які не отримали повного введення в концепції бази даних. Вона призначена для запобігання серйозних помилок.

Тільки для читання (BD-JBQ-lectura) це може бути доступним для всіх, хто відвідує сад

Ви вибираєте підключення під час запуску, а програма запитає вас про пароль, що відповідає обраному вами з'єднанню.



Якщо ви хочете переглянути деталі підключення, натисніть кнопку поруч із пунктом «Подробиці підключення», вона змінюється на , і вікно з'єднання відображатиметься як одне з наступних:



Як бачимо, ми підключаємося до того ж сервера баз даних, кожен зв'язок використовує ту саму базу даних на сервері, але з іншим користувачем.

Думаю далі про це

З іншого боку, ми ставимо під сумнів, чи це взагалі доречно, дозволяючи будь-якому користувачеві видалити щось на такому високому

рівні, як сім'я або рід, або, у зв'язку з цим, щось пов'язане з приєднаннями до колекції.

Спосіб, за допомогою якого ghini поставив запитання про функції програмного забезпечення, полягає у відкритті відповідної проблеми <https://github.com/Ghini/ghini.desktop/issues/218> `_.

- При зверненні до розробників вони обов'язково просять технічну інформацію або, принаймні, побачити екранний знімок. Допоможіть їм допомогти вам.

Зйомка екрану

У Linux існує три способи створення знімка екрана, все включається натисканням клавіші „PrtSc". Найбільш практичний варіант, можливо, комбінація клавіш „PrtSc" з Ctrl і Shift. Запуститься інтерактивний інструмент копіювання екрана. Ви вибираєте прямокутник і область буде скопійована в буфер обміну. Вставте його в електронний лист, який ви пишете, або в чаті, де розробники намагатимуться вам допомогти.

Де знаходяться журнали

Ghini постійно зберігає дуже інформативний файл журналу у файлі `~/.bauble/bauble.log`. Не хвилюйтеся відкриваючи його, просто надішліть його. Він містить багато технічної інформації.

Безперервне безпілотне оповіщення

Інший варіант - активувати обробник sentry. Він повідомить наш сторожовий сервер про будь-які серйозні ситуації в програмному забезпеченні. Якщо ви зареєструвалися, розробники будуть знати, як з вами зв'язуватися, якщо це необхідно.

Для здорових песимістів: ми не стежимо за тим, що ви робите, ми стежимо за тим, як працює наше програмне забезпечення. Ви завжди можете відмовитися.

Ви активуєте обробник Sentry на сторінці `: prefs`: шукайте рядок з назвою `bauble.use_sentry_handler`, якщо значення не є тим, що ви хочете, двічі натисніть по лінії, і вона зміниться на інше значення.

Таксономія

- Введення

Таксономічна складність орхідей

У JBQ ми найбільше працюємо з орхідеями сімейства Orchidaceae, однією з найбільших рослинних сімейств, з не менш 850 родів, організованих - згідно з Dressler - приблизно 70 субтрибами, 22 трибами, 5 підродин. Як ми представляємо цю інформацію, це не є очевидним і потребує пояснення.

Таксономія сімейства Orchidaceae постійно переглядається. Роди додаються, відмовляються, реорганізуються, визнаються синонімами, деякі таксономісти віддають перевагу групі видів або пологів поновому, інші розділяють їх знову і по-різному, ботаніки різних національностей можуть мати різні погляди на цю справу. Все це звучить дуже складно і спеціально, але це є частиною нашої повсякденної роботи, і все це можна зберегти в нашій базі даних Ghini.

- Визначення по рангу роду або сім'ї

По родовому рангу

Ghini-1.0 передбачає, що в усіх випадках приєднання виявляється за рівнем виду. Нинішній супровідник визнає, що це помилка, починаючи з ранніх днів Bauble, і яка Ghini-1.0 має спільне з іншим ботанічним програмним забезпеченням. Доки це не буде зафіксовано, ми покладаємось на встановлені практики.

Якщо приєднання визначається по родовому рангу, ми додаємо вигадані види в цьому роді, ми не вказуємо епітет свого виду (ми цього не знаємо), і ми додаємо незареєстрований епітет у розділі інфрачервоної інформації, як це:

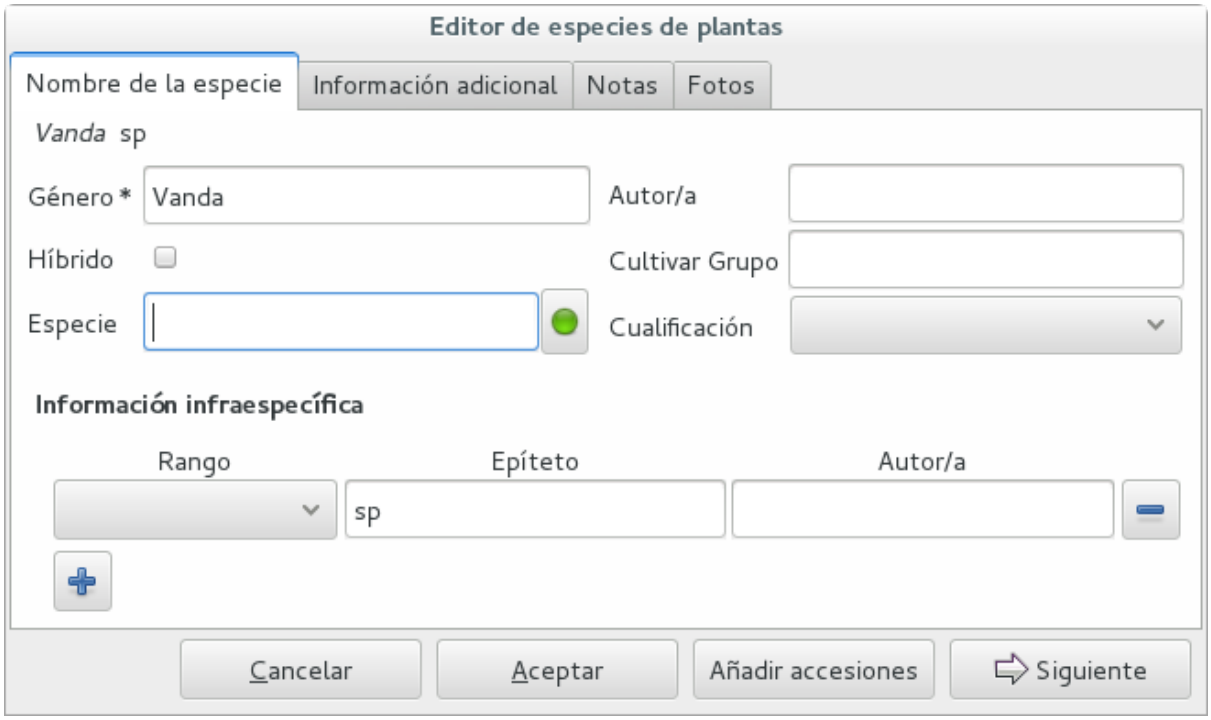
Коли він відображається в результатах пошуку, він виглядає так:

По сімейному рангу

Якщо приєднання визначається лише по рангу сім'ї, нам потрібен вигаданий рід, до якого ми можемо додати фіктивні види. Оскільки наш сад в першу чергу орієнтується на Orchidaceae, ми використовуємо дуже коротке ім'я **Zzz** для вигаданого роду в сім'ї, як це:

Поточний супровідник пропонує використовувати префікс **Zzz-** і за префіксом для написання імені сім'ї, можливо, зняття закінчення **e**. Видалення пробілу **e** корисно для того, щоб не отримувати результатів, які містять назви родів, коли ви, як для речовини, що закінчується **aceae**.

Окрім вищезазначеного **Zzz** у сімействі Orchidaceae, ми дотримуємося запропонованої практики, тому, наприклад, наша колекція вклю-



чатиме *Zzz-cactacea* або *Zzz-bromeliacea*.

Пам'ятайте: наш **Zzz** рід є вигаданим родом у сім'ї **Orchidaceae**, не використовуйте його як невизначений рід в інших сім'ях.

- Визначення за рангом, яке не дозволено програмним забезпеченням (наприклад: Subtribe або підсемейство)

По рангу субтриб

Ми іноді не можемо визначити таксонів у родовому класі, але нам вдається бути точнішим, ніж просто «це орхідея». Часто ми можемо вказати субтриб, це корисно, коли ви хочете виготовити гібриди.

Програмне забезпечення не дозволяє нам зберігати ряди, які є проміжними між сім'єю та родом, тому нам потрібно щось винайти, і це ми робимо:

Ми вставляємо вигаданий рід, називаючи його сабтриб, передусім називаючи його „Zzx-“, як у цьому прикладі:

Це Zzx-Laeliinae є деяким родом в Laeliinae субтрибу.

Щоб мати можливість вибирати рід за допомогою subtribe, ми також додаємо нотатку до вигаданого роду Zzx-Laeliinae, а також для всіх справжніх родів у цьому підзаголовоку, примітку категорії subtribus, відзначимо значення імені субтиба.

Це дозволяє виконувати такі запити, як:

```
genus where notes.note=Laeliinae
```

Ми дуже з нетерпінням чекаємо цього issue-9 вирішено!

По рангу підродини, триба

Оскільки ми зарезервували префікс Zzx- для субтрибу, ми зберігаємо префікси Zzy для трибу, Zzw- для підродини.

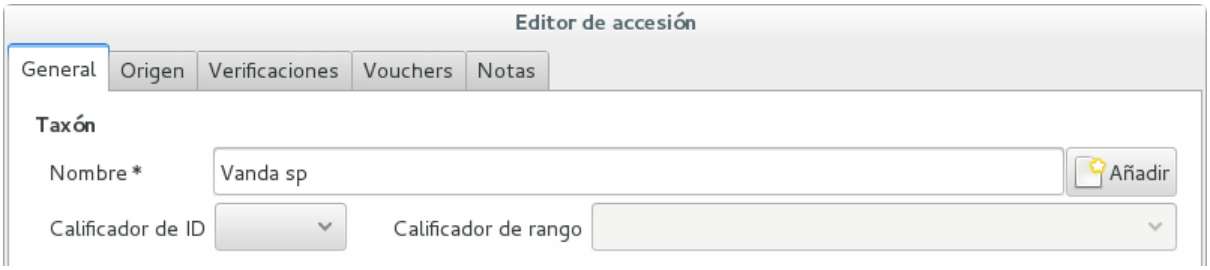
Зокрема інформація підродини є актуальною, оскільки в сімействах Orchidaceae є підродини, які ще не розділені.

- Редагування ідентифікації приєднання - відомості виду

Покажчик місця заповнення видів для окремих приєднань

Сценарій описує ідентифікацію єдиного приєднання, яке було пов'язане з «родовим», заповнювачем видів, щось типу "*Zzz* sp" або "*Vanda* sp";

У цьому випадку, коли виявляється вид рослини, ми змінюємо асоціацію при приєднанні, вибираючи інший вид.

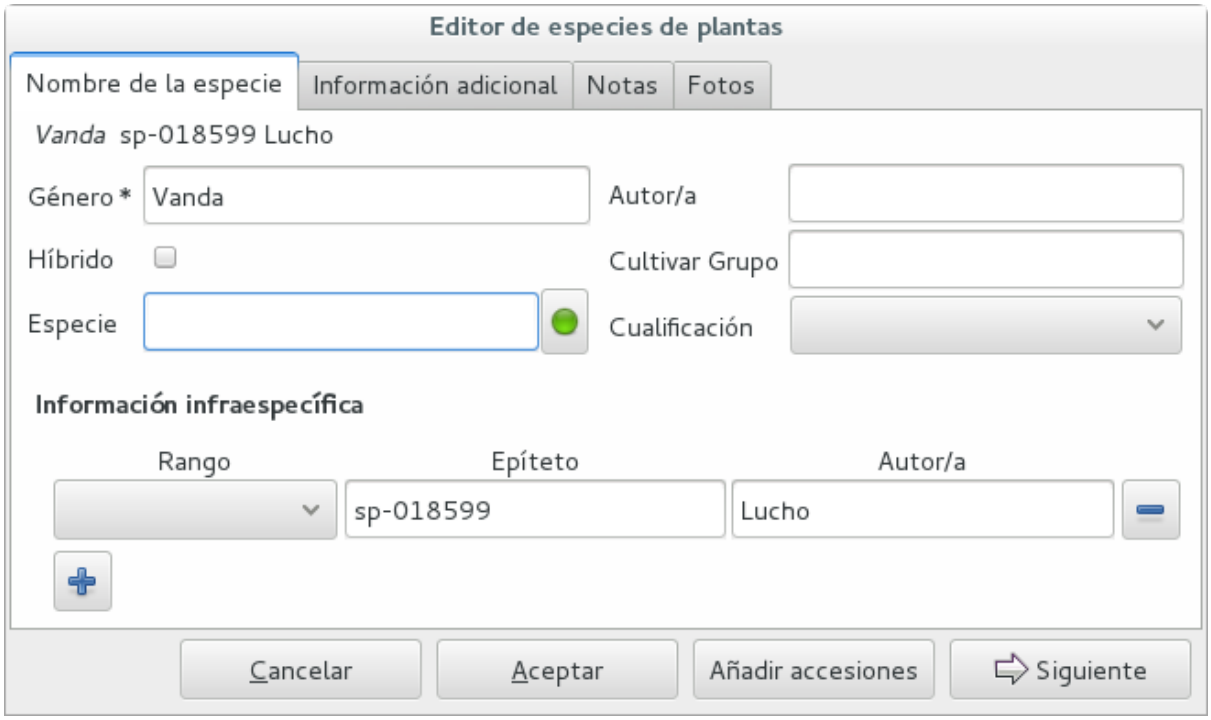


Ми не редагуємо види, оскільки можуть бути абсолютно незв'язані приєднання, пов'язані з тими ж типами заповнювачів.

Невідомі види для численних приєднань

Інша справа - коли ми маємо цілу партію приєднань, все це, очевидно, є одним і тим же видом, але ми не змогли його ідентифікувати. У цьому випадку ми пов'язуємо приєднання з неповним зазначеним видом, щось на кшталт "*Zzz* sp-59", бажано додати ім'я таксономіста, який створив асоціацію.

Вид, як "*Vanda* sp-018599", не є заповнювачем, це дуже конкретний вид, який ми ще не визначили.



У цьому випадку, коли вид ідентифікується (і це може бути навіть видом nova), ми безпосередньо редагуємо вид, тому всі приєднання, які належать до нього, отримують зміни.

- Нові рослини відносяться до виду, ще не в нашій колекції.

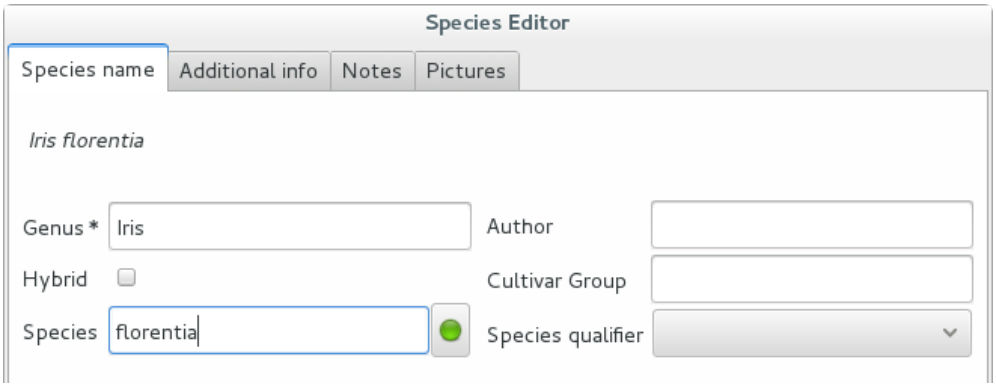
Остання хвилина видів

Ми починаємо це з вікна приєднання, і це дуже просто, просто натисніть + поруч із назвою виду, ми потрапляємо у вікно види.

- Додавання виду та використання онлайн-таксономічних послуг

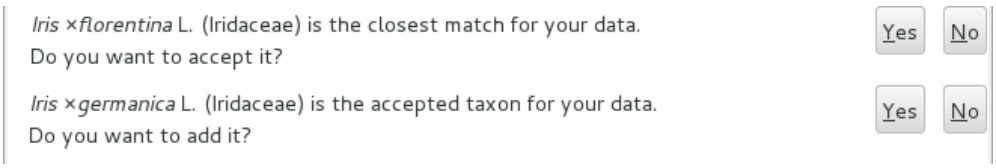
Додавання нового виду - список рослин.

Ми починаємо очевидний спосіб: наберіть епітет родини, можливо, виділіть його з списку завершень, потім введіть епітет виду або, принаймні, найкращий припущення.

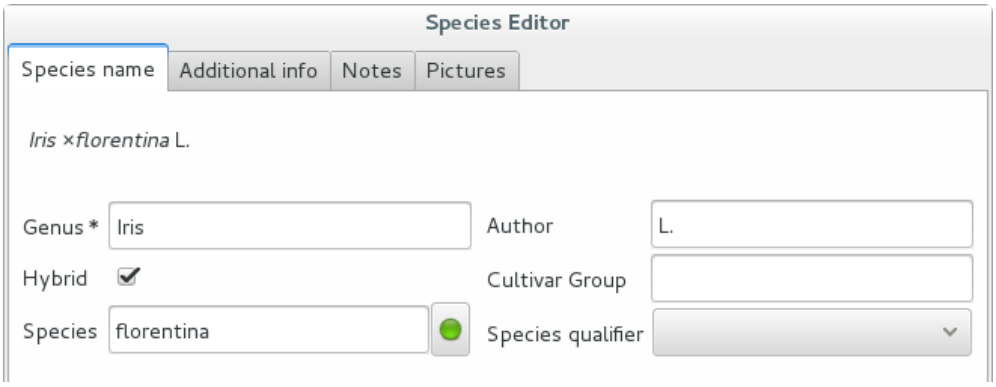


Поруч із видом епітетного поля є маленька кнопка,  , яка з'єднує нас із списком рослин. Натисніть на неї, у верхній частині вікна з'явиться повідомлення.

Залежно від швидкості вашого інтернет-з'єднання, а також від того, наскільки правильно опубліковане ім'я, верхня область зміниться на щось подібне:



Прийміть підказку, і це буде так, ніби ви самі набрали дані.



Перегляд цілого вибору - TNRS.

Це описано в посібнику, це надзвичайно корисно, не забувайте про це.

Нехай база даних підходить до саду

- Невичерпне завдання - переглянути те, що ми маємо в саду, і поєднати його з базами.

Початковий статус і змінні ресурси

Коли ми прийняли ghini, ми імпортували в це все, що правильно описано в базі даних файлових виробників. Ця база даних орієнтувалася виключно на Орхідеї, і навіть це було далеко не повне. На практиці ми все ще зустрічаємо помічені рослини в саду, які ніколи не були вставлені в базу даних.

Час від часу нам вдається отримати ресурси для перегляду саду, порівняння його з колекцією в базі даних, а головна діяльність полягає в тому, щоб вставити коди приєднання до бази даних, сфотографувати цю установку та помітити її місце розташування, Всі завдання, описані в решті частини цього розділу.

Невеликий застосунок ghini.pocket для Android був доданий до сім'ї Ghini, коли програміст Ghini був тут у Кіто. Це допомагає нам зробити знімок бази даних на місці під час прогулянки в саду, також це дозволяє дуже швидку інвентаризацію.

Процедури інвентаризації

Запускаючи ghini.pocket, ми записуємо назву місця, де ми будемо проводити інвентаризацію, наприклад (INV 1) оранжереї 1. Ми вводимо (тип або сканування, якщо на рослині є штрих-код або QR-код) код приєднання, і ми розглянемо його в ghini.pocket.

Побічним ефектом виконання пошуку є те, що ghini.pocket записує дату з часом, розташуванням та пошуковим кодом у текстовому файлі, який пізніше можна буде імпортувати до бази даних.

Для оранжереї приблизно з 1000 рослин наші оцінки вказують на те, що вам доведеться спокійно працювати два дні з 8:00 до 17:00.

Після імпорту файлу, створеного ghini.pocket, легко з'ясувати, які рослини відсутні. Наприклад: якщо ми провели інвентаризацію INV3 з 4 по 5 вересня, це відповідний пошук:

```
plant where location.code = 'INV3' and not notes.note like
↪'2017090%'
```

Усі ці рослини можуть бути позначені як мертві або втрачені відповідно до садової політики.

Візуалізуючи потребу таксономічної уваги

Наш протокол містить ще одну деталь, призначену для візуального виділення рослин, які потребують уваги таксономіста.

Це рослина, що з'являється лише в нашій базі даних, ідентифікованій на рівні сім'ї, або це ще не база даних отримує візуальний сигнал (наприклад, дерев'яна або пластмасова палиця, для морозива або фрі), щоб підкреслити, що це не ідентифіковано. Таким чином, відповідальний таксоном, під час екскурсії по оранжереї, може швидко їх помістити і, можливо, продовжувати додавати свою ідентифікацію в базу даних.

- Найменування конвенції в місцях саду

Деталі

code	опис
CAC-B *x*	Резервний для рослин кактуса поряд з експозицією орхідей теплиць.
CRV:	Виявлення непентаних
IC-*xx*:	теплі орхідеї в саду (від 1A до 9C є конкретні місця між орхідею)
IF-xx:	холодні орхідеї в саду (від 1A до 5I - конкретні місця в саду орхідей)
INV1:	теплиця 1 (тепло)
INV2:	теплиця 2 (холодно)
INV3:	теплиця 3 (тепло)

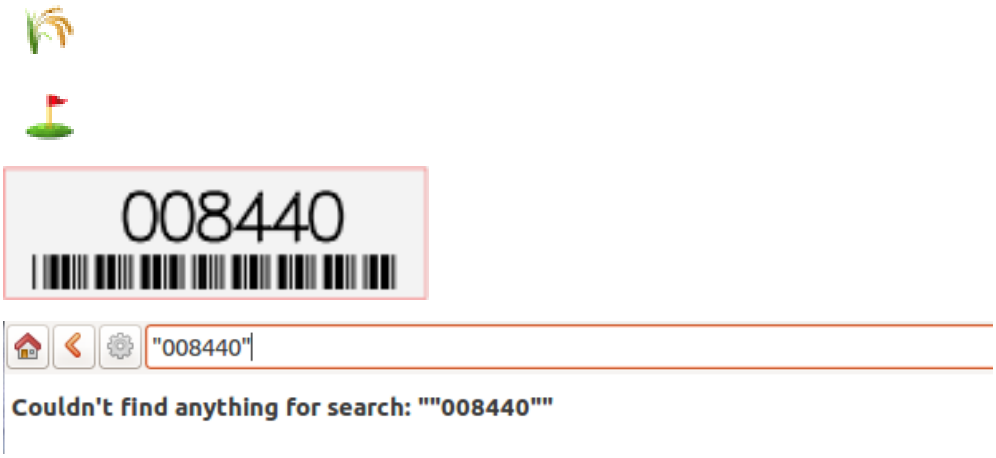
- Додавання приєднання для рослини

Очевидно, що ми продовжуємо збільшувати свою колекцію, коли рослини надходять із комерційних джерел або збираються з дикої природи, рідше виходять з експедицій у віддалені райони нашої країни, або ми отримуємо рослини, які були незаконно зібрані.

Іноді доводиться додавати рослини до цифрової колекції, лише тому, що ми їх фізично знаходимо в саду з міткою або без неї, але без їхнього цифрового аналога.

Існуюча рослина, знайдена в саду з власною етикеткою

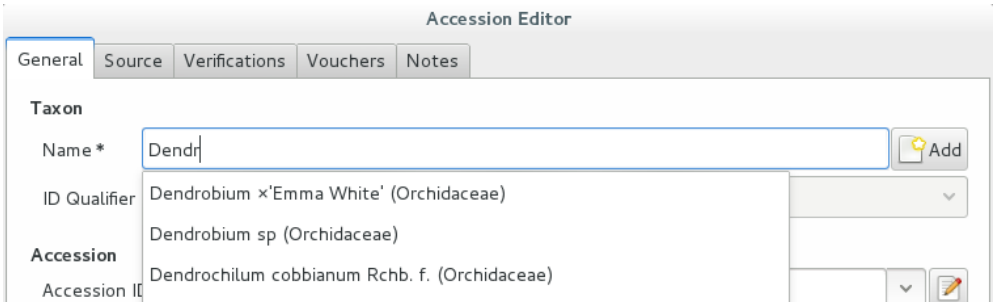
Ця діяльність починається з рослини, яка була знайдена у певному місці саду, а також знак приєднання та знання про те, що код входження не знаходиться в базі даних.



Для цього прикладу, припустимо, ми будемо вставляти цю інформацію в базу даних.

Приєднання	Вид	Розташування
008440	*Dendrobium* ×“Emma White“	Теплиця 1 (тепло)

Ми йдемо прямо в редактор приєднання, починаємо вводити назву виду у відповідному полі. На щастя, цей вид вже був у базі даних, інакше ми використали б кнопку **Додати** біля поля введення.



Ми вибираємо правильний вид, і ми заповнюємо ще пару полів, залишивши решту до значень за замовчуванням:

ID приєднання	Тип матеріалу	Кількість	Походження
008440	Рослина	1	Невідомо

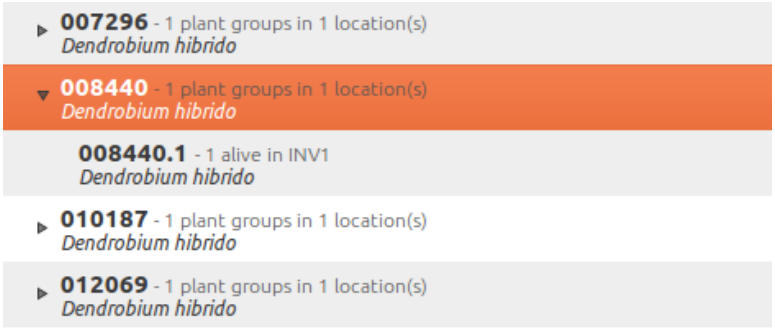
Після цього ми продовжуємо в редакторі рослин, натиснувши кнопку **Додати рослини**.

Ми не заповнюємо «**Призначені місця розташування**» приєднання, тому що ми не знаємо, що було первісним наміром, коли рослина

була вперше придбана.

У редакторі рослин ми вставляємо кількість та місце. І ми закінчили.

Рослина тепер є частиною бази даних:



Нове приєднання: рослина просто входить у сад

Ця діяльність починається з нової рослини, яка тільки що придбана від відомого джерела, ярлик рослини та передбачуваного розташування в саду.

Ми в основному робимо те ж саме, що і в тому випадку, коли рослина знаходиться в саду, є дві відмінності: (1) ми знаємо джерело рослини; (2) придбання цієї рослини було плановим заходом, і ми маємо намір розмістити його в певному місці в саду.

Знову ж таки ми йдемо прямо в редактор приєднання, починаємо вводити вид, і ми обираємо з списку завершень або додаємо його на льоту.

ID приєднання	Тип матеріалу	Кількість	Джерело
033724	Рослина	1	точно визначений

Після цього ми продовжуємо в редакторі рослин, натиснувши кнопку **Додати рослини**.

У редакторі рослин ми вставляємо кількість та місце.

Завбачте, що рослина може бути спочатку розміщена у теплиці, перш ніж вона досягне передбачуваного місця розташування в саду.

Існуюча рослина, знайдена в саду без етикетки

Коли це станеться, ми не можемо бути впевненими, що рослина ніколи не була в колекції, тому ми діємо так, наче ми переписали рослину. Це обговорюється в наступному розділі, але ми повернемося до випадку нового приєднання.

- Коли ми фізично пов'язуємо етикетку на рослину, завжди існує шанс, що щось трапиться або з рослиною (вона може зів'янути), або на етикетці (може стати нечитабельною) або з асоціацію (її можна розділити). У нас є програмні протоколи для цих подій.

Ми знаходимо мертву рослину

Кожного разу, коли рослина виявлена мертвою, ми забираємо її мітку та поміщаємо її в поле поруч із основним терміналом введення даних, поле позначено як "мертві рослини".

Безумовно, щонайменше раз на тиждень, бокс вичерпується, а база даних оновлюється з цією інформацією.

Мертві рослини не *видалені* з бази даних, вони залишаються там, але отримують **кількість** нуль. Якщо відомо про причину смерті, це також написано в базі даних.

Будь ласка, ще раз пам'ятайте, що **Рослина** не є **Приєднання**, і, будь ласка, пам'ятайте, що ми не видаляємо об'єкти з бази даних, ми просто додаємо їхню історію.

Вставте повний код рослини (щось на кшталт `012345.1` або `2017.0001.3`, і вам не потрібні провідні нулі чи лапки), натисніть правою кнопкою миші на відповідному рядку та натисніть **змінити** . Змінити кількість на 0, вказати причину і, бажано, також дату зміни.

Якщо вам потрібно додати будь-яку інформацію про смерть рослини, будь-ласка, використовуйте **примітку** та повторно використовуйте категорію примітки «death_cause».

Рослини з **кількістю** нуль показані у різному кольорі у вікні результатів. Це допомагає відрізняти їх від живих рослин.

Ми знаходимо рослину без етикетки

Ми не можемо бути впевнені, що рослина коли-набудь була в колекції чи ні. Ми припускаємо, що це було, і що її етикетка була втрачена.

Невдала втрата етикетки рослин, але це буває лише іноді. Що ми робимо, це ставимо на рослину нову етикетку і чітко говоримо, що етикетка є заміною оригіналу.

Потім ми розглядаємо справу так, наче це було нове приєднання, а також ми додали примітку до приєднання, категорію "етикетка", текст "перемарковані".

- Слідкуйте за різними джерелами рослинного матеріалу

Які різні джерела ми можемо мати

У цьому ботанічному саду ми одержуємо рослини з різних видів походження. Це може бути з експедицій (рослини виходячи з природи, зібрані з правовим дозволом MAE - Еквадорського Міністерства навколишнього середовища), подаровані рослини в основному приходять в якості подарунків від колекціонерів або орхідеї комерціалізації підприємств, придбаних або конфіскованих рослин (зазвичай відбуваються з MAE рейди по всій країні).

Якщо рослина походить з дикоростучого джерела

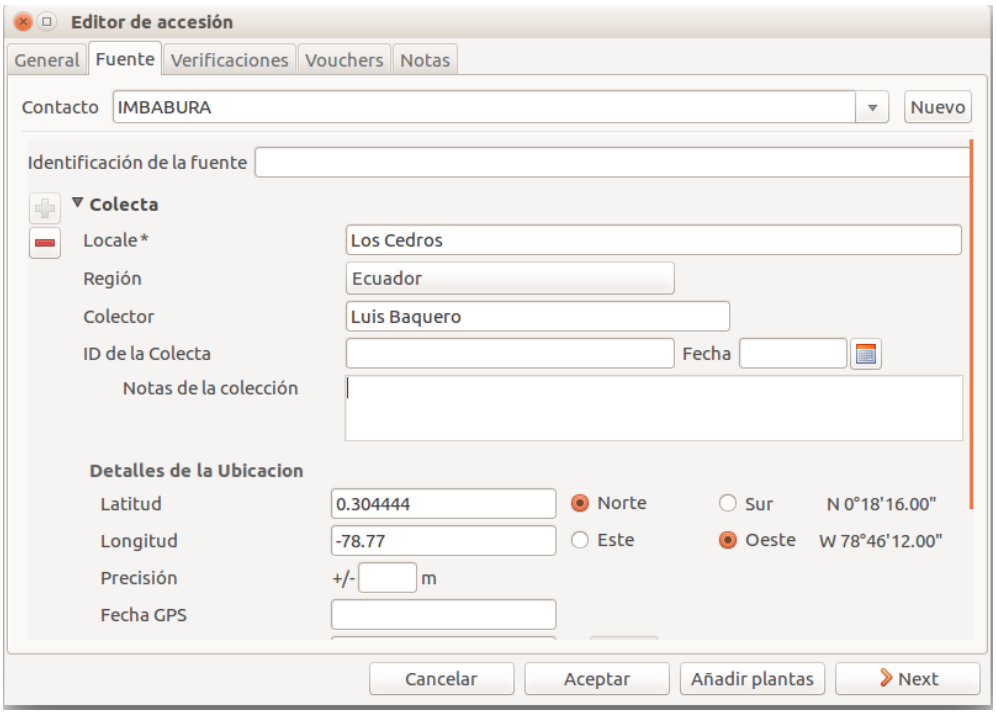
Редактор приєднання пропонує варіант «джерело». Коли рослина простежується до дикоростучого джерела, ми можемо вказати його конкретне походження. Ми хочемо дотримуватися ITF2, і ghini-1.0 лише частково дотримується цього стандарту. Параметри, що відповідають ITF2, такі:

- Дикоростучі: приєднання дикоростучого джерела.
- Культивовані: Росток (-и) з рослини дикоростучого джерела.
- Не дикоростучі: приєднання неможливо відстежити до дикоростучого джерела.
- Недостатньо даних

У випадку донорської рослини краще поставити детальну інформацію як записку на приєднання рослини; У випадку з рослиною з невідомим джерелом, ми обираємо параметр недостатньо даних.

Використання вкладки вихідного коду в редакторі приєднання

У цьому розділі ми можемо створити або використати контакт, наше джерело рослинного матеріалу. Це може бути від експедиції до місця збирання, і в цьому випадку ми будемо вказувати назву регіону та назви експедиції або можемо назвати особу чи підприємство, що пожертвувало певну партію рослин.



Вибравши або створивши контактну інформацію, в цьому розділі використовуються інші параметри, тут ви можете вказати регіон, де ви можете вибрати країну походження та конкретне місцезнаходження в регіоні, інформацію про географічну привабливість (включаючи дані GPS), опис місця проживання, назва колекціонера. Для останнього я рекомендую також написати конкретну дату поруч із назвою колекціонера (наприклад, Луїсом Бакеро, 10.11.2016 р.).

Пожертвувані, куплені чи конфісковані рослини

Однак це корисно для експедицій або для донорів, де основна інформація є географічною, така вкладка джерела не дуже практична у наших інших випадках: ми керуємося ще трьома категоріями: конфіскувати, придбати та пожертвувати, для цих категорій параметри, доступні на вкладці вихідного коду, не застосовуються: занадто багато інформації і не до речі.

У цих випадках ми додаємо набір нотаток, відповідно до справи.

— Пожертвувані рослини

Якщо рослина була подарована індивідуумом, ми додаємо індивідуум серед наших контактів і вказуємо його як джерело, потім додаємо примітки:

категорія	текст
source-type	подарунок
source-detail	Науковий внесок у JBQ

— **Купили рослини**

Якщо рослина була куплена, ми додаємо попереднього власника серед наших контактів і вказуємо його як джерело, потім додаємо примітки:

категорія	текст
source-type	придбання
source-detail	необов'язковий, вільний текст
factura	номер рахунку-фактури

— **Конфісковані рослини**

Якщо рослина була конфіскована, ми додаємо попереднього власника серед наших контактів і вказуємо його як джерело, потім додаємо примітки:

категорія	текст
source-type	конфіскований
source-detail	можливо, юридичні деталі, номер закону ...

- Виробництво або відтворення етикеток

Оновлення етикеток рослин

Іноді ми оновлюємо етикетки, наприклад, все, що є в теплиці, або, можливо, просто набір рослин, тому що їх етикетки ризикують стати нечитабельними.

У першому випадку легко вибрати всі рослини в місцезнаходженні, ми просто введемо назву місцезнаходження або надамо пошук `location like <location name>`.

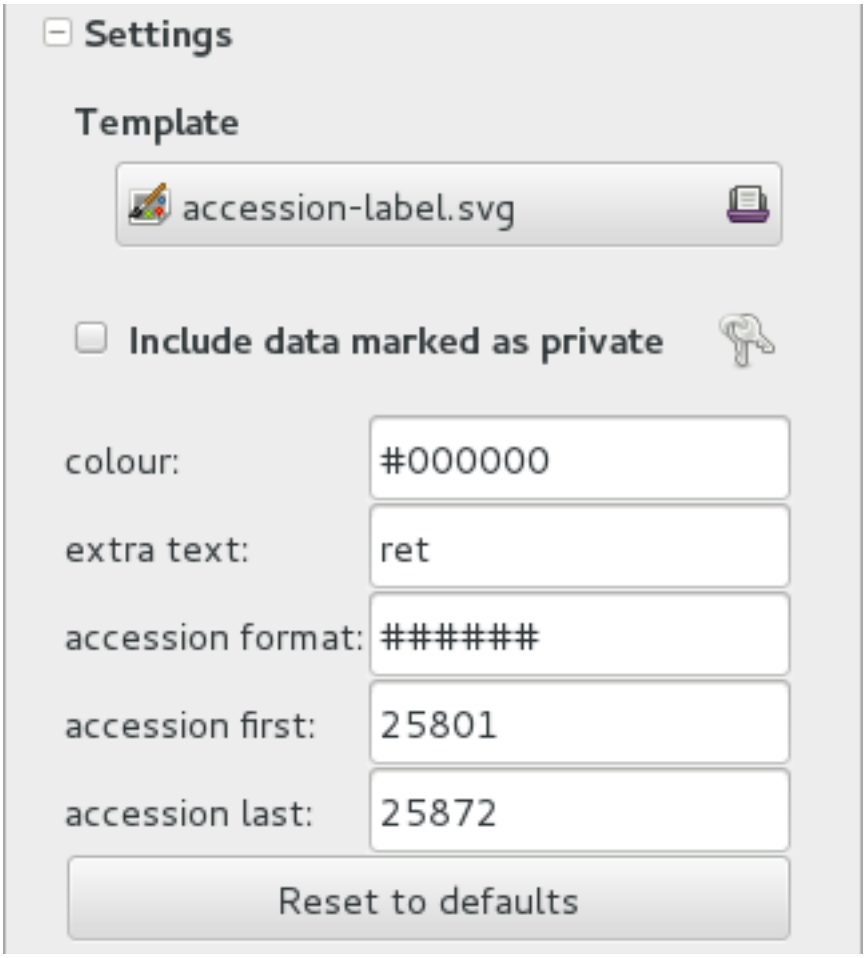
Другий випадок - трохи складніше. Що ми робимо, це створюємо тимчасовий **Тег** і використовуємо його для позначення всіх рослин, які були знайдені в необхідності для нової мітки.

З огляду на вибір, ми запускаємо інструмент звітування, використовуючи шаблон mako `accession-label.svg`. Ми скидаємо свої параметри до значень за замовчуванням, і оскільки ми використовуємо

простий принтер, ми встановлюємо колір на `чорний` замість `синій`, який призначений для гравюри.

Підготовка етикеток для не базових рослин

Для підготовки партії з 72 міток ми використовуємо шаблон звіту mako з ім'ям `accession-label.svg`. Цей шаблон приймає параметри, це приклад, який буде містити мітки від 025801 до 025872.



Етикетки приходять до нас у двох випадках: (1) або нові рослини, які просто прибувають у сад; (2) або рослини в саду, знайдені без етикетки. Ми розрізняємо два випадки, додавши „ret“ додатковий текст для перемаркованих рослин.

Ми зберігаємо два бокси з мітками двох типів, готових до використання.

- Наш сад має дві експозиційні теплиці та кілька теплих і холодних теплиць, в яких ми зберігаємо найбільшу частину нашої колекції. Рослини переміщуються до експозиції під час цвітіння і повертаються в «сховище», коли менш цікаві для експозиції. Для кожної рослини в нашій колекції нам потрібно знати його поточні розташування та історію рухів.

Плановані дії

Дія починається з переміщення рослин навколо та збирання коду рослин на папері або у нашому мобільному додатку, якщо така була.

Потім ми переходимо на настільний термінал і переглядаємо всі рослини по черзі, змінюючи їх розташування в базі даних. Важливо, щоб дата зміни місцезнаходження була правильно збережена, оскільки це говорить нам, як довго рослина залишається в експозиції.

Якщо б ми мали мобільний додаток, ми просто завантажили би інформацію на сервер, і ми зробимо це.

Фактична корекція

Під час перегляду саду ми знаходимо рослину в місці, де не базується база даних. Ми оновлюємо інформацію бази даних.

Наприклад, рослини, що належить до приєднання "012142", вид "*Acineta* sp" був знайдений в "Invernadero 1", тоді як база даних говорить, що вона знаходиться в "ICAlm3".

Все, що ми робимо, це знаходимо рослину в базі даних та оновлюємо інформацію. Ми не змінюємо нічого в початковій інформації про приєднання, а саме про поточну інформацію про рослину.

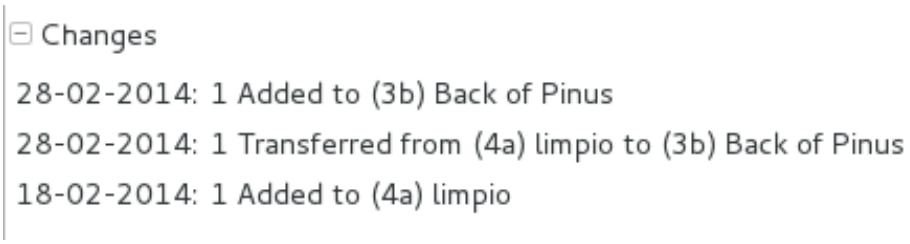
Ми вводимо код приєднання у поле вводу пошуку, з лапками, натискаємо Enter. Результати пошуку тепер показують приєднання, і це говорить нам, скільки рослин належить до нього. Натискаємо на квадраті + у рядку результатів, так що тепер ми також бачимо рядок для рослини, що належить до приєднання.

Натисніть правою кнопкою миші на рядку рослина, з'являться три варіанти: Редагувати, Розділити, Видалити, виберіть Редагувати, ви потрапляєте в редактор рослин.

Просто скопіюйте поле розташування та натисніть OK.

InfoBox містить інформацію про останню зміну об'єкта:

Для рослин, ще цікавіше, він створює історію змін, список яких включає в себе зміни місцезнаходження або кількість змін.



- Коли рослини виходять на етап цвітіння, ми можемо переглянути їх ідентифікацію безпосередньо, або ми робимо знімки деталей квітки, сподіваючись, що спеціаліст-відвідувач може допомогти завершити ідентифікацію.

Додавання зображень

Ми практикуємо з ODK Collect, невеликою програмою, що працює на ручних андроїдних пристроях. Використання Ghini у ODK Collect ще не заморожено до кращої практики. Погляньте на відповідний випуск на github.

- Регулярно нам потрібні звіти про нашу колекцію, які вимагає Міністерство екології Республіки Еквадор (MAE), і це виправдовує саме існування саду.

Створення звітів

Щороку ботанічний сад повинен подати доповідь (щорічний звіт про управління та підтримці колекції орхідей), що відповідає вимогам Міністерства навколишнього середовища Еквадору.

З цією метою ми починаємо вибирати рослини, які ми повинні включити до звіту. Це може бути все придбання в минулому році:

```
accession where _created between |datetime|2017,1,1| and␣
↪|datetime|2018,1,1|
```

або всі рослини в межах місця, або всі рослини, що належать до певного виду, або просто все (але це займе час):

```
plant where location = 'abc'
plant where accession.species.epithet='muricata' and accession.
↪species.genus.epithet='Annona'
plant like %
```

Вибравши у звіті об'єкти бази даних, ми запускаємо інструмент звітування, який діє на вибір.

Пошук в базі даних

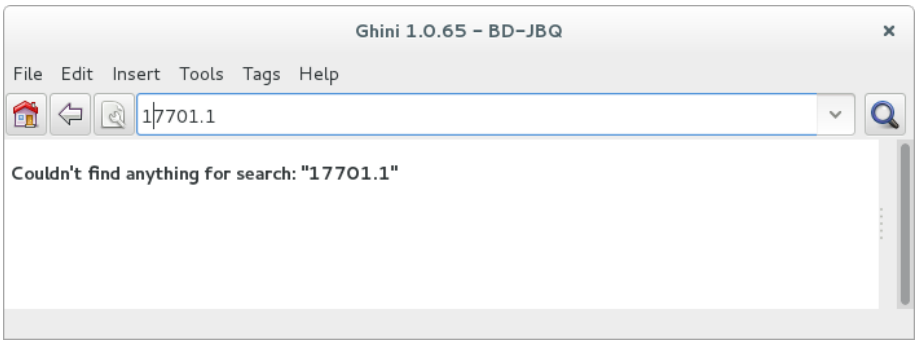
Ви шукаєте в базі даних, щоб змінити дані далі, або тому, що хочете створити звіт. Так чи інакше, ви починаєте щось набирати в полі пошуку

і ви сподіваєтеся побачити свій результат у представленні результатів пошуку.

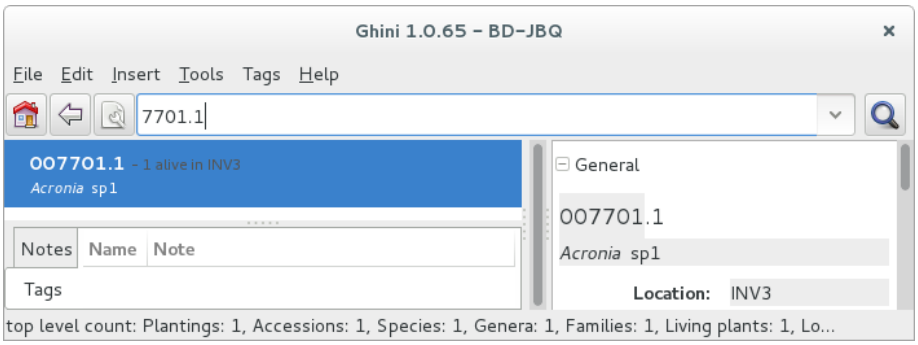
Пошук для редагування (рослина або приєднання)

Під час пошуку для редагування ви хочете бути дуже конкретними та вибрати якомога менше об'єктів. Найточніший пошук на основі номера рослини: ви знаєте код, ви отримуєте один об'єкт.

Якщо вашої рослини там немає, екран виглядатиме так:

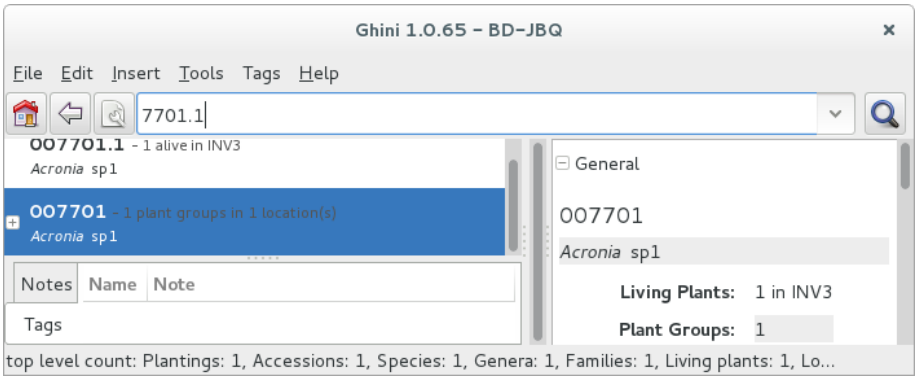


Інший приклад, в базі даних знаходиться рослина `007701.1`:



Всі поля з темнішим фоном у інформаційній коробці з правого боку є гіперпосиланнями на інші об'єкти в базі даних. Натискання на них буде або замінити текст у полі пошуку і виконувати запит, або просто додавати об'єкт до результатів.

Натискання на приєднання робить останній.



Тепер у представленні результатів пошуку є як Рослина, так і Приєднання, і тепер ми можемо редагувати одне або обидва.

Пошук з звітом

Під час пошуку для створення звіту ви хочете бути одночасно певними (ви не хочете повідомляти про невідповідні об'єкти) та широкі (ви не хочете повідомляти про один об'єкт).

Іноді сама доповідь пропонує запит, як, наприклад: всі рослини в теплиці 3; або: всі рослини, що належать до зникаючих видів (ми зберігаємо цю інформацію у примітці, пов'язаної з цим видом); або: всі рослини додані до колекції цього року;

```
plant where location.code = INV3
plant where accession.species.notes.note="endangered"
plant where accession._created > |datetime|2017,1,1|
```

Інший гнучкий спосіб досягти цього - працювати з **Тегами**.

Використовуйте Теги як розширений пошук

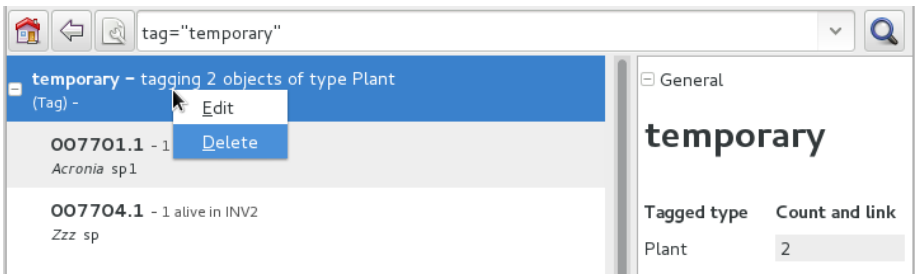
Іноді ми повинні робити однакові дії для об'єктів одного типу, але нам не вдається швидко подумати про пошуковий запит, який буде групувати все, що нам потрібно, і виключити все, що нам не потрібне.

Це одне можливе використання **Тегів**. Почнемо з вибору, ми позначаємо всі об'єкти у виборі під новим тимчасовим тегом. Скажімо, ми називаємо це «тимчасовий».

Ми продовжуємо пошук і додавання об'єктів до тимчасового тегу, доки тег не визначить всього, що нам потрібно.

Нарешті, в меню Теги вибираємо той, який ми щойно створили (у нашому прикладі це відповідає пошуку `tag = 'тимчасовий'`), і ми можемо викликати звіт.

Коли ми закінчимо тимчасовий тег, немає сенсу залишити його надалі, тому ми просто видалимо його.



Будьте обізнані про наявні стратегії пошуку

Це добре документовано, «più non dimandare» і read the docs.

4.1.2 using ghini for a seed database

We keep getting involved in groups focusing on endagered plant seeds. They want to note down when seeds come in, but also when they go out to people that order the seed.

In ghini, we keep speaking of »Plants«, »Locations«, while such user groups focus on »Seeds« and »Jars« and »Drawers« and »Boxes« and »Envelopes«. So people wonder whether ghini could be adapted to their use case, or for directions on how to develop their own database.

Does ghini need being adapted for such a seed database?

no it doesn't need any adaptation, it's just that you need to read some of its terms differently.

the taxonomy part is just taxonomy, plant species information, no need to explain that, no way to interpret it otherwise.

»Accessions« and »Plants«, you know what an »Accession« is, but since you're consistently handling »Plants« still only in seed form, the Wikipedia explanation of an accession sounds like this: it is a seed or group of seeds that are of the same taxon, are of the same propagule type (or treatment), were received from the same source, were received at the same time.

If you hold seeds in jars, or in other sort of containers that is able to hold hundreds of seeds, please make sure that a jar contains seeds of just one accession, as above described: same taxon, same treatment, same source, same time.

Each one of your »Jars« of seeds is in ghini speak a »Plant«, and the amount of seeds in the »Jar« is the »Plant« »quantity«. An »Envelope« is just the same as a »Jar«: a container of seeds from the same »Accession«, just presumably smaller.

A >Box< (where you keep several >Envelopes<) or a >Drawer< (where you keep several >Jars<) are in ghini speak a >Location<.

Since a >Jar< or an >Envelope< contains seeds from an >Accession<, you will clearly label it with its >Accession< code (and trailing >Plant< number). You might write the amount of seeds, too, but this would be repeating information from the database, and repeating information introduces an inconsistency risk factor.

How do I handle receiving a batch of seeds?

Примітка: When we receive seeds, we either collect them ourselves, or we receive it from an other seed collector. We handle receiving them possibly on the spot, or with a small delay. Even when handled together with several other batches of seeds we received, each batch keeps its individuality.

We want to be later able to find back, for example, how many seeds we still have from a specific batch, or when we last received seeds from a specific source.

As long as you put this information in the database, as long as you follow the same convention when doing so, you will be able to write and execute such queries using ghini.

One possibility, the one described here, is based on >Notes<. (Ghini does not, as yet, implement the concept «Acquisition». There is an issue related to the Acquisition and Donation objects, but we haven't quite formalized things yet.)

You surely already use codes to identify a batch of seeds entering the seed bank. Just copy this code in a >Note<, category „received", to each >Accession< in the received batch. This will let you select the >Accessions< by the query:

```
accession where notes[category='received'].note='<your code>'
```

Use the „Source" tab if you think so, it offers space for indicating an external source, or an expedition. When receiving from an external source, you can specify the code internal to their organization. This will be useful when requesting an extra batch.

How do I handle sending seeds?

what you physically do is to grab the desired amount of seeds of the indicated species from a jar, put it in an envelope and send it. what you do from a point of view of the database is exactly the same, but precisely described in a protocol:

- Use the database to identify the »Jar« containing the desired amount of the right seeds.
- remove that amount of seeds from the »Jar« (decrement the quantity),
- put the seeds in an »Envelope« (yes, that's a database object).
- send the envelope (but keep it in the database).

this in short.

When I send seeds, it's not just one bag, how does ghini help me keeping things together?

There's two levels of keeping things together: one is while you're preparing the sending, and then for later reference.

While preparing the sending, we advise you use a temporary »Tag« on the objects being edited.

For later reference, you will have common »Note« texts, to identify received and sent batches.

Can you give a complete example?

Right. Quite fair. Let's see...

Say you were requested to deliver 50 seeds of Parnassia palustris, 30 of Gentiana pneumonanthe, 80 of Fritillaria meleagris, and 30 of Hypericum pulchrum.

step 1

The first step is to check the quantities you have in house, and if you do have enough, where you have them. You do this per requested species:

```
accession where species.genus.epithet=Parnassia and species.
↪epithet=palustris and sum(plants.quantity)>0
```

Expand in the results pane the »Accession« from which you want to grab the seeds, so you see the corresponding »Jars«, highlight one, and tag it with a new »Tag«. To do this the first time, go through the steps, just once, of creating a new »Tag«. The new tag becomes the active tag, and subsequent tagging will be speedier. I would call the tag »sending«, but that's only for ease of exposition and further completely irrelevant.

Repeat the task for Gentiana pneumonanthe, Fritillaria meleagris, Hypericum pulchrum:

```
accession where species.genus.epithet=Gentiana and species.
↪epithet=pneumonanthe and sum(plants.quantity)>0
accession where species.genus.epithet=Fritillaria and species.
↪epithet=meleagris and sum(plants.quantity)>0
```

(continues on next page)

(continued from previous page)

```
accession where species.genus.epithet=Hypericum and species.
↪epithet=pulchrum and sum(plants.quantity)>0
```

Again highilight the accession from which you can grab seeds, and hit Ctrl-Y (this tags the highighted row with the active tag). Don't worry if nothing seems to happen when you hit Ctrl-Y, this is a silent operation.

step 2

Now we prepare to go to the seeds bank, with the envelopes we want to fill.

Select the >sending< >Tag< from the tags menu, this will bring back in the results pane all the tagged >Plants< (>Jars< or >Envelopes<), and will tell you in which >Location< (>Drawer< or >Box<) they are to be found. Write this information on each of your physical envelopes. Write also the >Species< name, and the quantity you can provide.

Walk now to your seeds bank and, for each of the envelopes you just prepared, open the >Location<, grab the >Plant<, extract the correct amount of seeds, put them in your physical envelope.

And back to the database!

step 3

If nobody used your workstation, you still have the Tag in the results pane, and it's expanded so you see all the individual plants you tagged.

One by one, you have to >split< the plant. This is a standard operation that you activate by right-clicking on the plant.

A plant editor window comes in view, in „split mode".

Splitting a plant lets you create a database image of the plant group you just physically created, eg: it lets you subtract 30 items from the Gentiana pneumonanthe plant (group number one, that is the one in the jar), and create a new plant group for the same accession. A good practice would be to specify as >Location< for this new plant the „out box", that is, the envelope is on its way to leave the garden.

Не забудьте видалити тимчасове „sending" >Tag<.

крок 4

Final step, it represents the physical step of sending the envelope, possibly together with several other envelopes, in a single sending, which should have a code.

Just as you did when you received a batch of plants, you work with notes, this time the category is „sent", and the note text is whatever you normally do to identify a sending. So suppose you're doing a second sending to Pino in 2018, you add the note to each of the newly created envelopes: category „sent", text: „2018-pino-002".

When you finally do send the envelopes, these stop being part of your collection. You still want to know that they have existed, but you do not want to count them among the seeds that are available to you.

Поверніть всі рослини у відправленні „2018-pino-002“ :

```
plant where notes[category='sent'].note = '2018-pino-002'
```

You now need to edit them one by one, mark the >quantity< to zero, and optionally specify the reason of the change, which would be >given away<, and the recipient is already specified in the „sent“ >Note<.

Ця остання операція може бути автоматизованою, ми маємо на увазі, що це стане скриптом, який діє на вибір. Залишайтеся на зв'язку.

Розділ 5

Адміністрація

5.1 Адміністрування бази даних

Якщо ви використовуєте реальну СУБД для зберігання ваших ботанічних даних, то вам потрібно робити адміністрування бази даних. Хоча адміністрування баз даних далеко виходить за рамки цього документа, ми виходимо з того, що наші користувачі знають про це.

5.1.1 SQLite

SQLite - це не те, що можна розглянути як справжню СУБД: кожна база даних SQLite знаходиться тільки в одному файлі. Зробіть копії для безпеки, і ви будете в порядку. Якщо ви не знаєте, де шукати ваші файли бази даних, то вважається, що за замовчуванням, bauble поміщає свої дані в `~/.bauble/` каталог.

У Windows це десь у каталозі ‘‘ AppData‘‘, швидше за все, в `AppData\Roaming\Bauble`. Майте на увазі, що Windows робить все можливе, щоб приховати `AppData` структуру каталогів для звичайних користувачів.

Найшвидший спосіб його відкрити за допомогою файлового провідника: тип `%APPDATA%` і натиснути Enter.

5.1.2 MySQL

Будь ласка зверніться до офіційної документації.

Резервне копіювання та відновлення баз даних широко і глибоко описано починаючи з цієї сторінки.

5.1.3 PostgreSQL

Будь ласка, зверніться до офіційної документації. Дуже ретельне обговорення варіантів резервного копіювання починається з розділу 24.

5.2 Конфігурація Ghini

Ghini використовує файл конфігурації для зберігання значень через виклики. Цей файл пов'язаний з обліковим записом користувача, і кожен користувач матиме власний файл конфігурації.

Щоб переглянути вміст конфігураційного файлу Ghini, введіть `:prefs` в області введення тексту, де ви зазвичай вводите пошукові запити, а потім натисніть enter.

Вам зазвичай не потрібно налаштувати файл конфігурації, якщо потрібно це можна зробити за допомогою звичайного текстового редактора. Конфігураційний файл Ghini знаходиться за замовчуванням папці бази даних SQLite.

5.3 Повідомлення про помилки

Якщо ви помітили що-небудь несподіване в поведінці Ghini, будь ласка, подайте проблеми на сайт розробки Ghini.

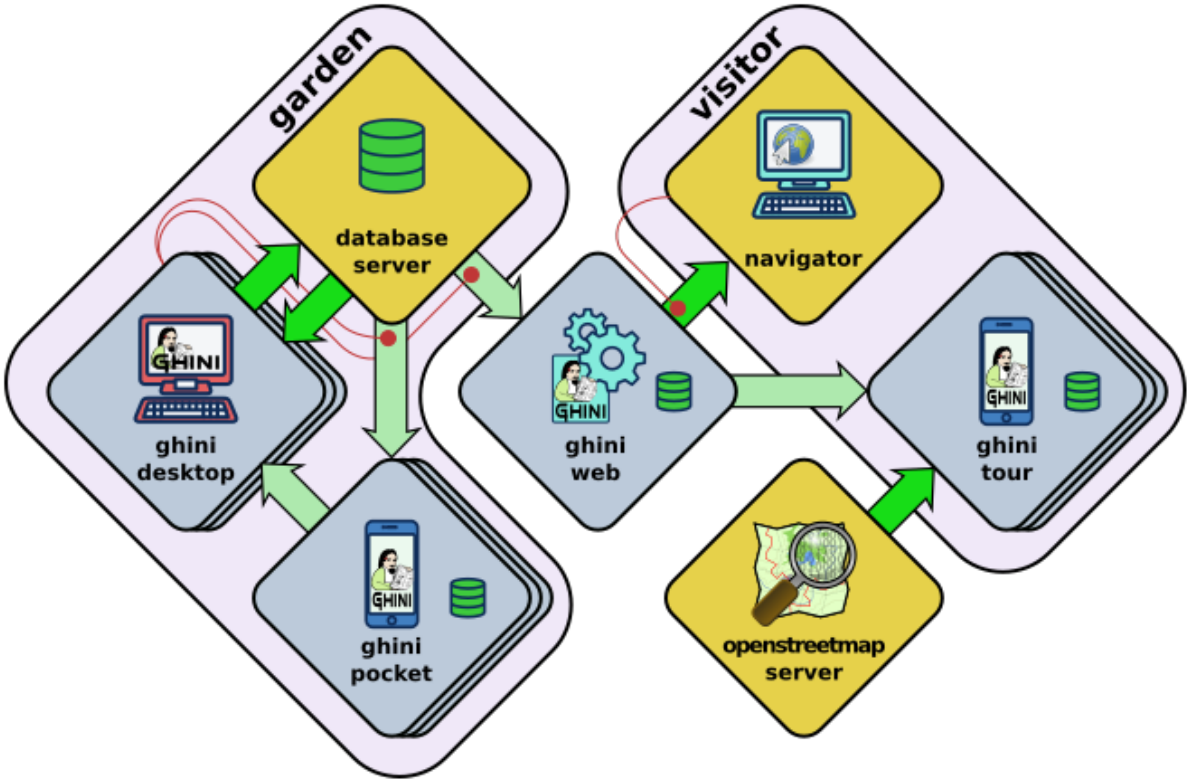
Сайт розробки Ghini можна отримати за допомогою меню Довідка.

Розділ 6

Сім'я Ghini

6.1 Сім'я Ghini

Почнемо з нагадування про склад сім'ї Ghini, як показано на схемі:



Ви навчилися використовувати ghini.desktop, тут ми вводимо інших членів сім'ї та їх взаємодію.

6.1.1 ghini.pocket

ghini.pocket - це програма для Android, яку ви можете встановити з play store. ghini.pocket, безумовно, інструмент, який ви будете використовувати найчастіше, поряд з ghini.desktop.

З ghini.pocket у вас завжди є останній знімок вашої бази даних з вами.

Введіть номер приєднання або скануйте його штрих-код або мітку QR, і ви взнаєте:

- ідентифікацію рослини,
- чи має зображення,
- коли увійшов до саду і
- з якого джерела.

Крім швидкого перегляду даних, ви можете використовувати ghini.pocket для...

корекції даних

If by your judgement, some of the information is incorrect, or if the plant is flowering and you want to immediately take a picture and store it in the database, you do not need take notes on paper, nor follow convolute procedures: ghini.pocket lets you write your corrections in a log file, take pictures associated to the plant, and you will import this information straight into the database, with further minimal user intervention.

перегляду інвентаризації

Початкова ідея, на якій ми заснували ghini.pocket, як і раніше є однією з його функціональних можливостей: перевірка інвентаризації.

Використовуючи ghini.pocket, перегляд інвентаризації теплиці, зокрема, якщо у вас є QR-коди на етикетках рослин, проходить так само швидко, як ви можете йти: просто введіть код місця положення вашої теплиці, скиньте журнал, потім один за одним скануйте рослинні коди рослин у теплиці. Немає необхідності в подальшому зборі даних.

Коли ви закінчите, імпортуйте журнал у ghini.desktop. Процедура, доступна в ghini.desktop, включає в себе додавання невідомих, але позначених рослин в базу даних, позначення як загублених / загиблих всіх рослин, що базуються в звітах як

живі та присутні в інвентарі, але не були знайдені під час інвентаризації.

таксономічної підтримки

Як бонус, ghini.pocket містить пошук фонетичного роду та цілком повну базу даних ботанічних таксонів з рангом між порядком і родом, у тому числі поколінням, та синонімами.

подальшої перевірки :будь-якої: *взаємодії між компонентами.*

6.1.2 ghini.web

ghini.web - це веб-сервер, написаний на nodejs.

Його найбільш помітна частина працює на http://gardens.ghini.me та показує як карту світу, де ви переглядаєте сади та шукаєте їх опубліковану колекцію.

Він також надає дані конфігурації для екземплярів ghini.tour.

подальшої перевірки :будь-якої: *взаємодії між компонентами.*

6.1.3 ghini.tour

ghini.tour - це програма для Android, яку ви можете встановити з play store.

Люди, які відвідують ваш сад, встановлюють ghini.tour на свій телефон або планшет, насолоджуються картою саду, знаючи, де вони знаходяться, і зможуть слухати аудіофайли, які ви розмістили у вигляді віртуальних інформаційних панелей у стратегічних місцях у вашому саді.

світогляд

при запуску ви бачите світ і сади. виберіть сад і введіть.

огляд саду

під час перегляду на рівні саду ви бачите панелі. виберіть панель і слухайте.

подальшої перевірки :будь-якої: *взаємодії між компонентами.*

6.1.4 потоки даних між програмними компонентами

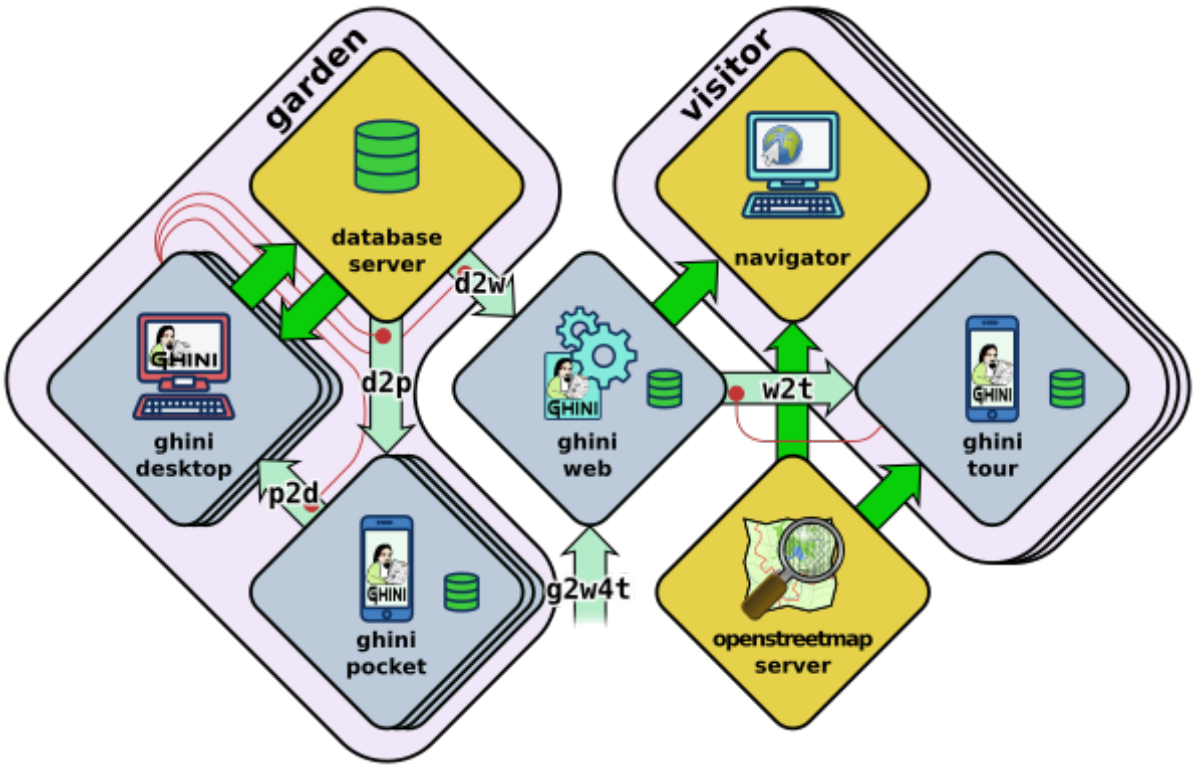
Примітка: Цей розділ містить технічну інформацію для менеджерів баз даних та розробників програмного забезпечення.

На діаграмі, що показує склад сім'ї Ghini, читач помітив попередження, як різні стрілки представляють різні потоки даних, мають різні кольори: деякі - глибоко зелені, інші - світліші відтінки.

Більш глибокі зелені потоки - це постійні потоки даних, що представляють основну активність компонента, наприклад: взаємодія між ghini.desktop і його сервером баз даних або вашим інтернет-браузером та ghini.web.

Більш світлі зелені потоки - це дії імпорту експорту, ініційовані користувачем на панелі команд ghini.desktop або на сторінці налаштувань ghini.tour.

Це той самий графік, в якому всі потоки імпорту даних отримано на ідентифікатор.



d2p: скопіювати знімок настільної бази до ghini.pocke

- експортувати настільну базу даних до pocket знімка
- скопіювати знімок на портативний пристрій

ghini.pocket тісно інтегрується з ghini.desktop, і це не інструмент для випадкового, ані зовнішнього користувача. Одним з завдань вашого менеджера баз даних для садів є регулярне копіювання оновленого знімка бази даних на ваш пристрій Android.

Ми радимо ввімкнути налагодження USB на пристрої. У перспективі це дозволить писати ghini.desktop безпосередньо в пристрій ghini.pocket.

Експортуйте файл з ghini.desktop, викликавши файл pocket.db, скопіюйте його на телефон:

```
adb -d push /tmp/pocket.db /sdcard/Android/data/me.ghini.pocket/
↪files/
```

Вище вказане місце дійсне, навіть якщо на вашому телефоні немає картки пам'яті.

Інші параметри включають Bluetooth або будь-який інший спосіб, який ви зазвичай використовуєте для копіювання звичайних файлів на своєму пристрої Android.

p2d: імпорт з файлу журналу ghini.pocket та зображень у центральну базу даних

навіть якщо ми як і раніше називаємо це журналом інвентаризації, журнал ghini.pocket містить більше, ніж просто виправлення інвентаризації.

- зробіть журнал на портативному пристрої
- імпортувати журнал у настільну базу даних

перш за все скопіюйте зібрану інформацію з ghini.pocket у свій комп'ютер:

```
export DIR=/some/directory/on/your/computer
adb -d pull /sdcard/Android/data/me.ghini.pocket/files/searches.txt
↪$DIR
adb -d pull -a /sdcard/Android/data/me.ghini.pocket/files/Pictures
↪$DIR
```

потім використовуйте ghini.desktop, щоб імпортувати цю інформацію у свою базу даних.

d2w: відправте вибрані дані вашого саду на ghini.web

Пропонуйте вибрані дані вашого саду на центральний сайт ghini.web, щоб онлайн-відвідувачі могли переглядати його. Це включає в себе ідентифікацію рослин та їх географічне розташування.

вміст цього потоку:

- сад: координати, назва, рівень масштабування (для початкового перегляду)
- рослини: координати, ідентифікація, рівень масштабування (за видимості)
- види: біноміальне, фонетичне наближення

---

g2w: додати географічні не-ботанічні дані до ghini.web

- Напишіть географічну інформацію про не-ботанічні дані (тобто: визначну пам'ятку в саду, потрібну ghini.tour) на веб-сайті ghini.web.

  вміст цього потоку:

  – віртуальні панелі: координати, заголовок, аудіофайл

  – світлини: координати, назва, зображення

  віртуальні панелі не обов'язково мають пов'язану світлину, світлини не обов'язково мають пов'язаний звуковий файл.

---

w2t: імпортування місць розташування та POI з ghini.web до екскурсії

вміст цього потоку:

- Сад (координати, назва, рівень масштабування)
- Цікаві місця (координати, назва, аудіофайл, світлина)

Розділ 7

Розвиток Ghini

7.1 Посібник розробника

If you ran the `devinstall` installation instructions, you have downloaded the sources, connected to the github repository. You are in the ideal situation to start looking into the software, understand how it works, contribute to ghini.desktop's development.

7.1.1 Допомога в розвитку Ghini

Якщо ви хочете внести свій внесок у Ghini, ви можете зробити це кількома різними способами:

- Використовуйте програмне забезпечення, відзначте те, що вам не подобається, *відкрийте проблему <http://github.com/Ghini/ghini.desktop/issues/new>* _ для кожної з них. Розробник буде реагувати швидше, ніж ви можете собі уявити.
- Якщо у вас є ідея про те, що пропущено в програмному забезпеченні, але не можете повністю формалізувати її в окремі питання, ви можете розглянути питання про наймання професіонала. Це найкращий спосіб переконатися, що щось відбувається швидко на Ghini. Переконайтеся, що розробник відкриває проблеми та публікує їхній внесок у github.
- Перекласти! Будь-яка допомога з перекладом буде прийнята, тому, будь ласка, будь ласка! Ви можете це зробити, не встановлюючи нічого на своєму комп'ютері, просто скористайтеся службою перекладів онлайн, яку пропонує http://hosted.weblate.org/
- виділіть репозиторій, виберіть проблему, вирішіть її, відкрийте запит зняти. Дивіться *bug solving workflow* нижче.

Якщо ви ще не встановили Ghini і хочете поглянути його історію кодів, ви можете відкрити нашу ‘github сторінку проєкту <http://github.com/Ghini/ghini.desktop>‘_і побачити все, що відбувалося навколо Ghini з моменту його створення, як в Bauble, ще в 2004 році.

If you install the software according to the `devinstall` instructions, you have the whole history in your local git clone.

7.1.2 Джерело програмного забезпечення, версії, розширення

Якщо ви хочете отримати певну версію Ghini, ми випускаємо та підтримуємо версії як розширення. Ви повинні ‘‘ git checkout‘‘ розширення, яку відповідає вибраній вами версії.

виробнича лінія

Назва розширень для стабільних (виробничих) версій Ghini складається з форми `ghini-x.y` (наприклад: ghini-1.0); імена розширень, у яких публікуються тестові версії Ghini, мають форму `ghini-x.y-dev` (наприклад, ghini-1.0-dev).

7.1.3 Процес розробки

Наш робочий процес полягає в тому, щоб постійно займатися тестуванням розширень, часто поширювати їх на github, щоб дозволити travis-ci та coveralls.io перевірити якість тестовних розширень, нарешті, час від часу, об’єднати тестовані розширення у відповідну версію.

Під час роботи над складними проблемами, які тривають більше двох днів, я можу відкрити філію, пов’язану з цією проблемою. Я не роблю це дуже часто.

більше питань

Коли постали перед однією складною проблемою, створіть тег відгалуження на кінці основної лінії розробки (наприклад, `ghini-1.0-dev`) і дотримуйтесь робочого процесу, описаного в

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

коротко:

```
git up
git checkout -b issue-xxxx
git push origin issue-xxxx
```

Працюйте в новій тимчасовій гілці. По готовності, перейдіть на github, з’єднайте гілку з основною лінією розвитку, з якої ви розгалужені, вирішіть конфлікти у разі потреби, видаліть тимчасову гілку.

Коли ви готові до публікації, об'єднайте лінію розвитку в відповідну виробничу лінію.

7.1.4 Оновлення установки перекладних рядків

Час від часу під час оновлення програмного забезпечення ви будете додавати або змінювати рядки в джерелах python, в документації, в glade джерелах. Більшість наших рядків є перекладами, і пропонуються weblate для людей, щоб зробити внесок у формі декількох `.po` файлів.

В основному `po` складається з пар текстових частин, оригінал і переклад і є специфічним для цільової мови. Коли перекладач додає переклад на weblate, це потрапляє до нашого сховища на github. Коли програміст додає рядок до програмного забезпечення, це досягає weblate «для перекладу».

Weblate хост Ghini проєкт. У рамках цього проєкту ми маємо компоненти, кожен з яких відповідає гілці сховища на github. Кожен компонент приймає переклад на декількох мовах.

компонент	сховище	гілка
Desktop 1.0	ghini.desktop	ghini-1.0-dev
Desktop 3.1	ghini.desktop	ghini-3.1-dev
Documentation 1.0	ghini.desktop-docs.i18n	ghini-1.0-dev
Documentation 3.1	ghini.desktop-docs.i18n	ghini-3.1-dev
Web 1.2	ghini.web	master
Pocket	ghini.pocket	master
Tour	ghini.tour	master

Щоб оновити файли `po` відносно *програмного забезпечення*, встановіть робочий каталог у корінь вашої копії *ghini.desktop* і запустіть скрипт:

```
./scripts/i18n.sh
```

Щоб оновити `po` файли *документації*, встановіть робочий каталог у корінь вашої копії *ghini.desktop-docs.i18n*, і запустіть скрипт:

```
./doc/runme.sh
```

Коли ви запускаєте будь-який з вищезазначених сценаріїв, швидше за все, вам потрібно фіксувати *all* "po"файли в проєкті. Ви можете переглянути зміни, перш ніж застосувати їх до репозиторію. Це найголовніше коли ви виконуєте критичну корекцію до рядка, наприклад видалення помилки.

Те, що відбувається: вступ у конфлікт. Вирішити конфлікти не складно, коли ви знаєте, як це зробити. Перш за все, додати weblate як віддалений:

```
git remote add weblate-doc10 https://hosted.weblate.org/git/ghini/documentation-
↪10/
```

Потім переконайтеся, що ми знаходимося в правильному сховищі, на правильній гілці, оновіть дистанційне керування, злийте з ним:

```
git checkout ghini-1.0-dev
git remote update
git merge weblate-doc10/ghini-1.0-dev
```

Our documentation на readtodocs є оригінальна англійська версія, і кілька перекладів. Ми просто дотримуємося description of localisation, немає нічого, що ми винайшли собі тут.

Readthedocs перевіряє налаштування проєкту *Language*, і викликає `sphinx-intl``для отримання відформатованої документації на цільовій мові. З конфігурацією за замовчуванням -що ми не змінювали- ``sphinx-intl` очікує один `po` файл на один вихідний документ, названий як вихідний документ і що всі вони перебувають у каталозі `local/$(LANG)/LC_MESSAGES/`.

З іншого боку, Weblate (і ми самі) віддає перевагу одному `po` файлу на кожну мову, і зберігаємо їх всіх у однакових директоріях `/ po`, подібно тому, як ми робимо для програмного проєкту: `/po/$(LANG).po`.

Щоб не повторювати інформацію, і дозволити обом системам працювати природним способом, у нас є два набори символічних посилань (git визначає їх).

Підбиваючи підсумок: коли файл у документації оновлюється, `runme.sh` скрипт буде:

1. скопіюйте `rst` файли з програмного забезпечення в документацію;
2. Створіть новий `pot` файл для кожного з файлів документації;
3. об'єднайте всі `pot` файли в один `doc.pot`;
4. використовуйте оновлений `doc.pot` для оновлення всіх файлів `doc.po` (по одному на кожну мову);
5. створити всі символічні посилання:
   a. ті, що очікуються `sphinx-intl` в `/local/$(LANG)/LC_MESSAGES/`
   б. ті, що використовуються weblate в `/po/$(LANG).po`

Ми могли б напевно записати вищевказане в Makefile, або навіть краще включити його в `/doc/Makefile`. Хто знає, може, ми це зробимо.

7.1.5 Producing the docs locally

The above description is about how we help external sites produce our documentation so that it is online for all to see. But what if you want to have the documentation locally, for example if you want to edit and review before pushing your commits to the cloud?

In order to run sphinx locally, you need to install it **within** the same virtual environment as ghini, and to install it there, you need to have a sphinx version whose dependencies don not conflict with ghini.desktop's dependecies.

What we do to keep this in order?

We state this extra dependency in the `setup.py` file, as an `extras_require` entry. Create and activate the virtual environment, then run `easy_install ghini.desktop[docs]`. This gets you the sphinx version as declared in the `setup.py` file.

If all you want is the html documentation built locally, run `./setup.py install docs`. For more options, enter the `doc` directory and run `make`.

7.1.6 Яким чином перекладені рядки досягають наших користувачів?

Новий перекладач поставив запитання, додавши: «Чи це автоматичний процес від Weblate –> GIT –> Ghini Desktop встановлений на комп'ютерах користувачів, чи це вимагає ручних кроків?

Вважається, що вся взаємодія досить складна, і це залежить від компонента.

When you install `ghini.desktop` or one of the Android apps, the installation doesn't assume a specific run-time language: a user can change their language configuration any time. So what we do is to install the software in English together with a translation table from English to whatever else.

At run-time the GUI libraries (Android or GTK) know where to look for the translation strings. These translation tables are generated during the installation or upgrade process, based on the strings you see on Weblate.

The path followed by translations is: You edit strings on Weblate, Weblate keeps accumulating them until you are done, or you don't interact with Weblate for a longer while; Weblate pushes the strings to github, directly into the development line `ghini-1.0-dev`; I see them and I might blindly trust or prefer to review them, maybe I look them up in wikipedia or get them translated back to Italian, Spanish or English by some automatic translation service; sometimes I need to solve conflicts arising because of changed context, not too often fortunately. As said, this lands in the development line `ghini-1.0-dev`, which I regularly publish to the production line `ghini-1.0`, and this is the moment when the new translations finally make it to the distributed software.

Users will notice a *new version available* warning and can decide to ignore it, or to update.

For `ghini.pocket`, it is similar, but the notification is handled by the Android system. We publish on the Play Store, and depending on your settings, your phone will update the software automatically, or only notify you, or do nothing. It depends on how you configured automatic updates.

Для `ghini.web` ми ще не визначили, як його розповсюджувати.

Для документації ghini вона повністю автоматична, і все це обробляється readthedocs.org.

7.1.7 Додавання відсутніх одиничних тестів

Якщо ви зацікавлені в сприянні розвитку Ghini, хороший спосіб зробити це буде, допомагаючи нам знаходити та писати пропущені тестові блоки.

Добре перевіреною функцією є та, чия поведінка не може змінюватися, не порушуючи принаймні одне тестування.

Ми всі погоджуємося, що в теорії теорія та практика ідеально узгоджуються, і що спочатку пишуть тести, потім реалізують функцію. На практиці, однак, практика не збігається з теорією, і ми писали тести після написання та навіть публікації функцій.

Цей розділ описує процес додавання одиночних тестів для `bauble.plugins.plants.family.remove_callback`.

Що для тесту

Перш за все, відкрийте індекс звіту про охоплення та виберіть файл з низьким рівнем охоплення.

Для цього прикладу, починаючи з жовтня 2015 року, ми приземлилися на `bauble.plugins.plants.family`, на 33%.

https://coveralls.io/builds/3741152/source?filename=bauble%2Fplugins%2Fplants%2Ffamily.py

Перші дві функції, які потребують тестів, `edit_callback` і `add_genera_callback`, включають створення та активацію об'єкта, що спирається на спеціальне діалогове вікно. Ми повинні спочатку написати одиничні тести для цього класу, а потім повернутися сюди.

Наступна функція `remove_callback` також активує пару діалогових вікон і повідомлень, але у формі виклику функції, що запитує вхід користувача через поля yes-no-ok. Ці функції ми можемо легко замінити функцією, що висміює поведінку.

як протестувати

Отже, вирішивши, що описувати в одиничному тесті, ми розглянемо код, і бачимо, що це потребує дискримінації пари випадків:

**** коректність параметра ****

- у списку сімей немає елементів.
- список сімей містить більше одного елемента.
- список сімей має рівно один елемент.

каскад

- у сім'ї немає родів
- у сім'ї один чи більше родів

підтвердити

- користувач підтверджує видалення
- користувач не підтверджує видалення

видалення

- при видаленні сім'ї все пройшло добре
- під час видалення сім'ї виникає помилка

Я вирішу зосередитись лише на аспектах **каскад** і **підтвердити**. Два бінарних питання: 4 випадки.

де поставити тести

Знайдіть тестовий скрипт і виберіть клас, де слід поставити додаткові одиничні тести.

https://coveralls.io/builds/3741152/source?filename=bauble%2Fplugins%2Fplants%2Ftest.py#L273

що про пропущені тести

Клас `FamilyTests` містить пропущене тестування, і його реалізація займе дещо час, тому що нам потрібно переписати FamilyEditorPresenter, відокремити його від FamilyEditorView і переглянути, що робити з класом FamilyEditor, який, на мою думку, повинен бути вилучений і замінений однією функцією.

написання тестів

Після останнього тесту в класі FamilyTests я додаю чотири випадки, які я хочу описати, і я переконаний, що вони завершаться невдачею, і оскільки я ледачий, я пишу найбільш компактний код, який я знаю, для генерації помилки:

```
def test_remove_callback_no_genera_no_confirm(self):
    1/0

def test_remove_callback_no_genera_confirm(self):
    1/0

def test_remove_callback_with_genera_no_confirm(self):
    1/0

def test_remove_callback_with_genera_confirm(self):
    1/0
```

Один тест, крок за кроком

Почнемо з першого тесту.

Під час написання тестів я зазвичай дотримуюсь зразка:

- T_0 (початковий стан),
- дія,
- T_1 (тестування результату дії з урахуванням початкових умов)

що в імені - одиничні тести

Є причина, чому одиничні тести називаються одиничними тестами. Будь-ласка, ніколи не перевіряйте дві дії в одному тесті.

Тож давайте опишемо T_0 для першого тесту, бази даних, що містить родину без родів:

```
def test_remove_callback_no_genera_no_confirm(self):
    f5 = Family(family=u'Arecaceae')
    self.session.add(f5)
    self.session.flush()
```

Ми не хочемо, щоб тестована функція викликала інтерактивну функцію `utils.yes_no_dialog`, а хочемо викликати неінтерактивну функцію заміни `remove_callback`. Ми досягаємо цього просто, зробивши вираз `utils.yes_no_dialog` виразом `lambda`, який, як і початкова інтерактивна функція, приймає один параметр і повертає логічний. У цьому випадку: `False`:

```
def test_remove_callback_no_genera_no_confirm(self):
    # T_0
    f5 = Family(family=u'Arecaceae')
    self.session.add(f5)
    self.session.flush()

    # action
    utils.yes_no_dialog = lambda x: False
    from bauble.plugins.plants.family import remove_callback
    remove_callback(f5)
```

Далі ми перевіряємо результат.

Що ж, ми не хочемо просто перевірити, чи був об'єкт Arecaceae видалений, чи ні, ми також повинні перевірити значення, яке повертає `remove_callback`, і чи `yes_no_dialog` і “ message_details_dialog“ були викликані чи ні .

Для цього не вистачає виразу `lambda`. Ми робимо щось набагато складніше, що зробить життя набагато простіше.

Давайте спочатку визначимо досить загальну функцію:

```
def mockfunc(msg=None, name=None, caller=None, result=None):
    caller.invoked.append((name, msg))
    return result
```

і ми захоплюємо `partial` з стандартного модуля `functools`, частково застосувати вищенаведене `` mockfunc``, залишивши тільки `msg` невизначеним, і використовуйте цю часткову програму, яка є функцією, що приймає один параметр і повертає значення, замінити дві функції в `utils`. Тестова функція тепер виглядає так:

```
def test_remove_callback_no_genera_no_confirm(self):
    # T_0
    f5 = Family(family=u'Arecaceae')
    self.session.add(f5)
    self.session.flush()
    self.invoked = []

    # action
    utils.yes_no_dialog = partial(
        mockfunc, name='yes_no_dialog', caller=self, result=False)
    utils.message_details_dialog = partial(
        mockfunc, name='message_details_dialog', caller=self)
    from bauble.plugins.plants.family import remove_callback
    result = remove_callback([f5])
    self.session.flush()
```

Тестовий розділ перевіряє, що *message_ details_ dialog‘* не було викликано, що `yes_no_dialog` було викликано, з правильним параметром повідомлення, що Arecaceae все ще існує:

```
# effect
self.assertFalse('message_details_dialog' in
                 [f for (f, m) in self.invoked])
self.assertTrue(('yes_no_dialog', u'Are you sure you want to '
                 'remove the family <i>Arecaceae</i>?')
                in self.invoked)
self.assertEquals(result, None)
q = self.session.query(Family).filter_by(family=u"Arecaceae")
matching = q.all()
self.assertEquals(matching, [f5])
```

І так далі

„Є два види людей, ті, хто завершує те, що вони починають, і так далі“

Наступне тестування майже однакове, з тією різницею, що `utils.yes_no_dialog` повинен повертати `True` (це ми досягаємо, вказавши `result = True` в частковому застосуванні загальної форми `mockfunc` ‘).

За допомогою цієї дії значення, яке повертає `remove_callback`, має бути `True`, і не повинно більше бути сім’ї Arecaceae в базі даних:

```
def test_remove_callback_no_genera_confirm(self):
    # T_0
```

(continues on next page)

(continued from previous page)

```
    f5 = Family(family=u'Arecaceae')
    self.session.add(f5)
    self.session.flush()
    self.invoked = []

    # action
    utils.yes_no_dialog = partial(
        mockfunc, name='yes_no_dialog', caller=self, result=True)
    utils.message_details_dialog = partial(
        mockfunc, name='message_details_dialog', caller=self)
    from bauble.plugins.plants.family import remove_callback
    result = remove_callback([f5])
    self.session.flush()

    # effect
    self.assertFalse('message_details_dialog' in
                     [f for (f, m) in self.invoked])
    self.assertTrue(('yes_no_dialog', u'Are you sure you want to '
                     'remove the family <i>Arecaceae</i>?')
                    in self.invoked)
    self.assertEquals(result, True)
    q = self.session.query(Family).filter_by(family=u"Arecaceae")
    matching = q.all()
    self.assertEquals(matching, [])
```

подивіться на commit 734f5bb9fffc2f4bd22578fcee1802c8682ca83 для двох інших тестових функцій.

Реєстрація тестування

Наші об'єкти `bauble.test.BaubleTestCase` використовують обробники класу `bauble.test.MockLoggingHandler`. Кожен раз, коли починається індивідуальне тестування, метод `setUp` створить новий `handler` I пов'язати його з кореневим реєстратором. Метод `tearDown` дбає про його видалення.

Ви можете перевірити присутність конкретних повідомлень журналу в “ self.handler.messages“. “ Messages“ - словник, спочатку порожній, з двома рівнями індексації. По-перше, ім'я реєстратора, який видає журнал реєстрації, потім ім'я рівня запису журналу. Ключі створюються, коли це необхідно. Значення містить списки повідомлень, відформатовані за будь-яким форматом, який ви зв'язуєте з обробником, за умовчанням для `logging.Formatter ("%(message)s")`.

Ви можете явним чином порожнім зібраним повідомленням викликати `self.handler.clear()`.

Введення всіх разом

Час від часу ви хочете активізувати тестовий клас, з яким ви працюєте:

```
nosetests bauble/plugins/plants/test.py:FamilyTests
```

І в кінці процесу ви хочете оновити статистику:

```
./scripts/update-coverage.sh
```

7.1.8 Структура користувацького інтерфейсу

Інтерфейс користувача побудований відповідно до архітектурного шаблону **Model** — **View** — **Presenter**. Для більшості інтерфейсів **Model** є об'єктом бази даних SQLAlchemy, але у нас також є елементи інтерфейсу, де немає відповідної моделі бази даних. В загальному:

- **View** описується як частина файлу **glade**. Це повинно включати асоціації зворотного виклику сигналу та ListStore-TreeView. Просто повторно використовуйте базовий клас `GenericEditorView`, визначений у `bauble.editor`. Коли ви створюєте екземпляр цього загальнодоступного класу, передайте його назву файлу **glade** та ім'я root-віджет, а потім передайте цей екземпляр класу конструктору **presenter**.

  У файлі поля, у розділі `action-widgets`, що закриває опис об'єкта GtkDialog, переконайтеся, що кожен елемент `action-widget` має дійсне `response` значення. Використовуйте дійсні значення GtkResponseType, наприклад:

  – GTK_RESPONSE_OK, -5
  – GTK_RESPONSE_CANCEL, -6
  – GTK_RESPONSE_YES, -8
  – GTK_RESPONSE_NO, -9

  Існує непростий спосіб об'єднати тестування представленням підкласу, тому, будь ласка, не переглядайте підклас, там насправді немає потреби.

  У файлі поля, кожен вхідний віджет повинен визначити, який обробник активується на який сигнал. Загальний клас Presenter пропонує загальні зворотні виклики, які охоплюють найбільш поширені випадки.

  – GtkEntry (однорядкове введення) буде обробляти сигнал `changed`, з будь-яким `on_text_entry_changed` або `on_unique_text_entry_changed`.
  – GtkTextView: пов'язати його з GtkTextBuffer. Для обробки сигналу `changed` на GtkTextBuffer, ми повинні визначити обробник, який викликає загальний `on_textbuffer_changed`, єдина роль для цієї функції полягає в тому, щоб передати нашому загальному обробнику ім'я атрибуту моделі, який отримує зміну. Це робочий апарат для невирішеної помилки в GTK.
  – GtkComboBox з перекладеними текстами не може бути легко оброблено з glade файлу, тому ми навіть не намагаємося. Використовуйте метод `init_translatable_combo` загального класу `GenericEditorView`, але будь ласка, введіть його з **presenter **.

- **Model** - це лише об'єкт з відомими атрибутами. У цій взаємодії **model** є просто пасивним контейнером даних, це не більше, ніж дозволити **presenter** змінити це.
- Підклас **Presenter** визначає та реалізує:
  - `widget_to_field_map`, словник, що пов'язує імена віджетів з назвою атрибутів моделі,
  - `view_accept_buttons`, список імен віджетів, які, якщо вони активуються користувачем, означають, що вигляд повинен бути закритий,
  - всі необхідні зворотні виклики,
  - за бажанням, це також відтворює **model**.

  presenter постійно оновлює **model** відповідно до змін у **view**. Якщо **model** відповідає об'єкту бази даних, **presenter** зобов'язує всі **model** оновити базу даних, коли **view** успішно закрито або повертається назад, якщо **view** скасовано (ця поведінку залежить від параметра `do_commit`)

  Якщо **model** є чимось іншим, то **presenter** зробить щось інше.

Примітка: Хороша поведінка **presenter** використовує **view** api, щоб запитати значення, вставлені користувачем, або насильно встановити статуси віджетів. Будь ласка, не використовуйте практики наших невірних ведучих, деякі з яких безпосередньо обробляють поля „view.widgets“. Таким чином, ведучі перешкоджають нам створити одиничні тести.

Базовий клас для ведучого, `GenericEditorPresenter`, визначений у `bauble.editor`, реалізує багато корисних загальних зворотних викликів. Існує клас `MockView`, який ви можете використовувати при написанні тестів для своїх ведучих.

Приклади

`Contact` та `ContactPresenter` реалізуються за наведеними вище рядками. Вид визначається у файлі `contact.glade`.

Хорошим приклад шаблону Presenter/View (без model) надає менеджер з'єднань.

Ми використовуємо той самий архітектурний шаблон для взаємодії, що не пов'язаний з базою даних, шляхом встановлення ведучого також як модель. Ми робимо це, наприклад, для діалогового вікна експорту JSON. Наступна команда дасть вам список `GenericEditorView` інстанцій:

```
grep -nHr -e GenericEditorView\( bauble
```

7.1.9 Розширення Ghini з плагінами

Майже все Ghini розширюється за допомогою плагінів. Плагіни можуть створювати таблиці, визначати власні пошукові запити, додавати пункти меню, створювати власні команди та багато іншого.

Щоб створити новий плагін вам необхідно розширити `bauble.pluginmgr.Plugin` клас.

Плагін `Tag` є гарним мінімальним прикладом, навіть якщо `TagItemGUI` виходить за рамки моделі архітектури Model-View-Presenter.

7.1.10 Структура плагінів

Ghini є основою для обробки колекцій, і поширюється разом із набором плагінів, що робить Ghini ботанічним менеджером збору. Але Ghini залишається рамкою, і ви могли б теоретично видалити всі плагіни, які ми поширюємо та пишемо власноруч, або напишіть свої власні плагіни, які розширять чи завершать поточну поведінку Ghini.

Після того як ви вибрали та відкрили з'єднання з базою даних, ви виходите в вікно пошуку. Вікно пошуку - це взаємодія між двома об'єктами: SearchPresenter (SP) та SearchView (SV).

SV - це те, що ви бачите, SP зберігає стан програми та обробляє запити, які ви висловлюєте через SV. Обробка цих запитів впливає на вміст SV та статус програми в SP.

Результати пошуку, відображені у більшій частині SV, є рядками, об'єктами, що є класами, зареєстрованими у плагіні.

Кожне з цих класів повинно реалізовувати певну кількість функцій, щоб правильно поводитися в рамці Ghini. Система Ghini зберігає простір для підключених класів.

SP знає всі зареєстровані (підключені) класи, вони зберігаються в словнику, асоціюючи клас із його реалізацією плагінів. SV має слот (gtk.Box), де ви можете додати елементи. У будь-який час, максимум лише один елемент у слоті видно.

Плагін визначає один або кілька класів плагінів. Клас плагіна відіграє роль часткового ведучого (pP - плагін-ведучий), оскільки він реалізує зворотні виклики, необхідні для відповідного приєднання часткового вигляду в слоті (pV-plugin view), а шаблон MVP завершується батьківським ведучим (SP), знову діючи як модель. Підсумувати і завершити:

- SP виступає як модель,
- частковий вигляд pV визначається у glade файлі.
- зворотні зв'язки, реалізовані за допомогою pP, посилаються на glade файл.
- контекстне меню для рядка SP,
- дочірні власності.

коли ви реєструєте клас плагіна, SP:

- додає pV в слот і робить його невидимим.
- додає екземпляр pP у зареєстровані класи плагінів.
- повідомляє pP, що SP є моделлю.
- з'єднує всі виклики від pV до pP.

коли елемент у тригерах pV викликає дію в pP, pP може передати дію SP і може запитати SP, що він оновлює модель та оновлює представлення даних.

Коли користувач вибирає рядок в SP, SP ховає все в сполученому слоті і показує лише один pV відносно типу вибраного рядка, і запитує pP, щоб оновити pV з будь-яким параметром відносно вибраного рядка.

Окрім встановлення видимості різних pV, нічого не потрібно відключати чи видаляти: невидиме pv не може викликати події!

7.1.11 робочий процес над помилками

нормальний робочий процес розробки

- під час користування програмним забезпеченням, ви помічаєте проблеми, або ви отримаєте уявлення про те, що може бути краще, ви думаєте про це, достатньо для того, щоб мати дуже чітке уявлення про це. Ви відкриваєте випуск і описуєте проблему. Хтось може реагувати на це.
- ви відкриваєте сайт випусків і вибираєте один який ви хочете вирішувати.
- призначити цей випуск для себе, таким чином, ви інформуєте спільноту, що у вас є намір працювати над ним.
- за потреби, розгалуження сховища у вашому обліковому записі бажано створити гілку, безспірно пов'язану до випуску.
- написати одиничні тести і передати їх у свою гілку (не надавайте несправні тести на github, спочатку запустіть `nosetests`).
- написати додатковий блок тестів (в ідеалі, тести утворюють повний опис функції, ви це додаєте або коректуєте).
- переконайтеся, що функція, яку ви додаєте або коректуєте дійсно повністю описує блок тестів, які ви написали.
- переконайтеся, що блок тестів є простим, тобто, що ви тестуєте варіації змін по одній змінній. не додають введення складних для блоку тестів або тестів які не поміщаються на одному екрані (25 рядків в коді).
- написати код, який робить тести успішними.
- оновити файли i18n (запустити `./scripts/i18n.sh`).

- всякий раз, коли це можливо, переведіть нові рядки які ви вставляєте в код або glade файли.
- коли ви змінюєте рядки, переконайтеся, що старі переклади повторно використовуються.
- зафіксувати зміни.
- передати github.
- відкрити пул запиту.

публікація продукту

please use the `publish.sh` script, in the `scritps` directory. This one takes care of every single step, and produces recognizable commit comments, it publishes the release on pypi, and in perspective it will contain all steps for producing a `deb` file, and a windows executable.

Ви також можете це зробити вручну:

- відкрийте сторінку запиту, використовуючи як основу виробничу лінію `ghini-x.y` у порівнянні з `ghini-x.y-dev`.
- переконайтеся що `bump` фіксація включена в розбіжностях.
- повинна бути забезпечена можливість автоматичного злиття гілок.
- створити новий пул запиту, викличте його як “publish to the production line”.
- вам, можливо, буде потрібно почекати travis-сі для виконання перевірки.
- об’єднати зміни.

не забудьте розповісти світові про новий випуск: на facebook <https://www.facebook.com/bauble.thesoftware/>‘_, the google group, у будь-якій відповідній пов’язаній групі на нашій веб-сторінці <http://ghini.github.io/>‘_.

ваша власна гілка

Якщо ви хочете зберегти власну гілку проєкту, майте на увазі, що це робота в повну силу над результатом, і щоб бути в курсі, потрібні певні зусилля з вашої сторони.

Найкращий спосіб зберегти власну гілку полягає в тому, щоб сфокусуватися на певній конкретній проблемі, часто відкривають тягові запити для вашої роботи, переконайтеся, що ви це приймаєте. Просто дотримуйтесь стилю кодування Ghini, пишіть одиничні тести, лаконічні та ясні, і не повинно бути ніяких проблем в тому, щоб ваша робота була включена до Ghini.

Якщо гілка вийшла з синхронізації з Ghini вгору: читати, зрозуміти, слідкувати за github довідниками configuring a remote for a fork і syncing a fork.

крок закриття

- розглянути цей робочий процес. розглядати це в якості орієнтира, до себе і до своїх колег. будь ласка, допоможіть зробити його краще і відповідно практичним.

7.1.12 Distributing ghini.desktop

Python Package Index - PyPI

This is not much mentioned, but we keep ghini.desktop on the Python Package Index, so you could install it by no more than:

```
pip install ghini.desktop
```

There are a couple packages that can't be installed with `pip`, but otherwise that's really all you need to type, and it's platform independent.

Publishing on PyPI is a standard `setup` command:

```
python setup.py sdist --formats zip upload -r pypi
```

Windows

For building a Windows installer or executable you need a running Windows system. The methods described here has been used successfully on Windows 7, 8 and 10. Windows Vista should also work but has not been tested.

If you are on GNU/Linux, or on OSX, you are not interested in the remainder of this section. None of Ghini's contributors knows how to produce a Windows installer without having a Windows system.

The goal of the present instructions is to help you produce a Windows installer, that is a single executable that you can run on any Windows workstation and that will install a specific version of ghini.desktop. This is achieved with the NSIS script-driven installer authoring tool.

As a side product of the installer production, you will have a massive but relocatable directory, which you can copy to a USB drive and which will let you use the software without needing an installation.

The files and directories relevant to this section:

- `scripts/build-win.bat` — the single batch script to run.
- `setup.py` — implements the NSIS and py2exe commands.
- `scripts/build-multiuser.nsi` — the nsis script, used by the above.
- `nsis/` — contains redistributable NSIS files, put here for conveniency.

- `ghini-runtime/` — built by `py2exe`, used by `nsis`.
- `dist/` — receives the executable installation file.

Most steps are automated in the `build-win.bat` script. Installation of a few tools needs to be done manually:

1. Download and install Git, Python 2.7 and PyGTK.

   This is outlined in the `devinstall`-based installation instructions.

2. Download and install NSIS v3.

3. Рекомендується **перезавантажити**.

4. Clone the ghini.desktop repository.

   Use your own fork if you plan contributing patches, or the organization's repository `https://github.com/Ghini/ghini.desktop.git` if you only wish to follow development.

   Clone the repository from GitHub to wherever you want to keep it, and checkout a branch. Replace `<path-to-keep-ghini>` with the path of your choice, e.g. `Local\github\Ghini\`. Production branch `ghini-1.0` is recommended as used in the example.

   To do this, open a command prompt and type these commands:

```
cd <path-to-keep-ghini>
git clone <ghini.desktop repository URL>
cd ghini.desktop
git checkout ghini-1.0
```

The result of the above is a complete development environment, on Windows, with NSIS. Use it to follow development, or to propose your pull requests, and to build Windows installers.

All subsequent steps are automated in the `scripts\build_win.bat` script. Run it, and after a couple of minutes you should have a new `dist\ghini.desktop-<version>-setup.exe` file, and a working, complete relocatable directory named `ghini-runtime`.

Read the rest if you need details about the way the script works.

The `build_win.bat` script

Доступний пакетний файл, який може завершити останні кілька кроків. Для його використання скористайтеся цією командою:

```
scripts\build_win.bat
```

`build_win.bat` приймає 2 аргументи:

1. `/e` — executable only.

Produce an executable only, skipping the extra step of building an installer, and will copy `win_gtk.bat` into place.

2. `venv_path` — A path to the location for the virtual environment to use.

   Defaults to `"%HOMEDRIVE%%HOMEPATH%"\.virtualenvs\%CHECKOUT%-exe`, where `CHECKOUT` corresponds to the name of the branch you checked out.

If you want to produce an executable only and use a virtual environment in a folder beside where you have ghini.desktop, you could execute `scripts\build_win.bat /e ..\ghi2exe`

---

py2exe не буде працювати з eggs

Створення виконуваного файлу Windows із py2exe вимагає встановлення пакетів ** не** як eggs. Для цього є кілька методів, зокрема:

- Install using `pip`. The easiest method is to install into a virtual environment that doesn't currently have any modules installed as eggs using `pip install .` as described below. If you do wish to install over the top of an install with eggs (e.g. the environment created by `devinstall.bat`) you can try `pip install -I .` but your mileage may vary.
- Додавши:

  ```
  [easy_install]
  zip_ok = False
  ```

  до setup.cfg (або аналогічним чином `zip_safe = False` до `setuptools.setup()` в `setup.py`) можна використовувати `python setup.py install`, але вам необхідно завантажити та встановити 'Microsoft Visual C++ Compiler для Python 2.7 <http://aka.ms/vcpython27>'_щоб отримати будь-яке розширення C, і йому буде потрібно свіже віртуальне середовище без залежних пакетів, встановлених як eggs.

The included `build-win` script uses the `pip` method.

---

installing virtualenv and working with environments

Install virtualenv, create a virtual environment and activate it.

With only Python 2.7 on your system (where `<path-to-venv>` is the path to where you wish to keep the virtual environment) use:

```
pip install virtualenv
virtualenv --system-site-packages <path-to-venv>
call <path-to-venv>\Scripts\activate.bat
```

On systems where Python 3 is also installed you may need to either call pip and virtualenv with absolute paths, e.g. `C:\Python27\Scripts\pip` or use the Python launcher e.g. `py -2.7 -m pip` (run `python --version` first to check. If you get anything other than version 2.7 you'll need to use one of these methods.)

Populate the virtual environment

Встановити залежності та ghini.desktop у віртуальне середовище:

```
pip install psycopg2 Pygments py2exe_py2
pip install .
```

Compile for Windows

Створіть виконуваний файл:

```
python setup.py py2exe
```

The `ghini-runtime` folder will now contain a full working copy of the software in a frozen, self contained state.

This folder is what is packaged by NSIS.

This same folder can also be transferred however you like and will work in place. (e.g. placed on a USB flash drive for demonstration purposes or copied manually to `C:\Program Files` with a shortcut created on the desktop). To start ghini.desktop double click `ghini.exe` in explorer (or create a shortcut to it).

Fixing paths to GTK components.

If you run the relocatable compiled program, unpackaged, you might occasionally have trouble with the GUI not displaying correctly.

Should this happen, you need to set up paths to the GTK components correctly. You can do this by running the `win_gtk.bat`, from the `ghini-runtime` folder.

You will only need to run this once each time the location of the folder changes. Thereafter `ghini.exe` will run as expected.

Finally, invoke NSIS

Створіть встановлювач:

```
python setup.py nsis
```

This should leave a file named `ghini.desktop-<version>-setup.exe` in the `dist` folder. This is your Windows installer.

про програму установки

- Можливість однокористувацької або глобальної інсталяції.
- На даний момент ghini.desktop, встановлений таким чином, що не буде перевіряти або повідомляти вам про оновлену версію. Вам потрібно буде перевірити це самому.
- Можливість завантаження та встановлення необов'язкових додаткових компонентів:
  - Apache FOP - Якщо ви хочете використовувати шаблони звітів xslt, встановіть FOP. FOP вимагає Java Runtime. Якщо у вас цього не встановлено, програма установки повідомить вас і запропонує відкрити веб-сайт Oracle для завантаження та встановлення з нього.
  - MS Visual C runtime - Ви, швидше за все, цього не потребуєте, але якщо у вас виникли проблеми з запуском ghini.desktop, спробуйте встановити час виконання MS Visual C (наприклад, перезапустити програму встановлення та вибрати лише цей компонент).
- Може бути запущено без використання командного рядка (наприклад, для віддаленого розгортання) з наступними аргументами:
  - `/S` для тихого;
  - `/ AllUser` (при запуску від імені адміністратора) або `/CurrentUser`
  - `/C = [gFC]`, щоб вказати компоненти, де:

    `g` = Скасувати вибір основного компонента ghini.desktop (корисний для додавання додаткового компонента після первинної установки)

    `F` = вибрати Apache FOP

    `C` = вибрати MS Visual C runtime

Debian

Between 2009 and 2010 someone packaged the then already obsolete Bauble 0.9.7 for Debian, and the package was included in Ubuntu. That version is still being distributed, regardless being it impossible to install.

Only recently has Mario Frasca produced a new bauble debian package, for the latest bauble.classic version 1.0.56, and proposed for inclusion in Debian. View it on mentors. This version depends on `fibra`, a package that was never added to Debian and which

Mario also has packaged and proposed for inclusion in Debian. Mario has been trying to activate some Debian Developer, to take action. There's not much more we can do, other than wait for a sponsor, and hoping the package will eventually get all the way to Ubuntu.

Once we get in contact with a Debian Sponsor who will review what we publish on mentors, then we will be definitely expected to keep updating the debian package for `ghini.desktop` and `fibra`.

I am not going to explain in a few words the content of several books on Debian packaging. Please choose your sources. For a very compact idea of what you're expected to do, have a look at `scripts/pubish.sh`.

7.2 Template Letters

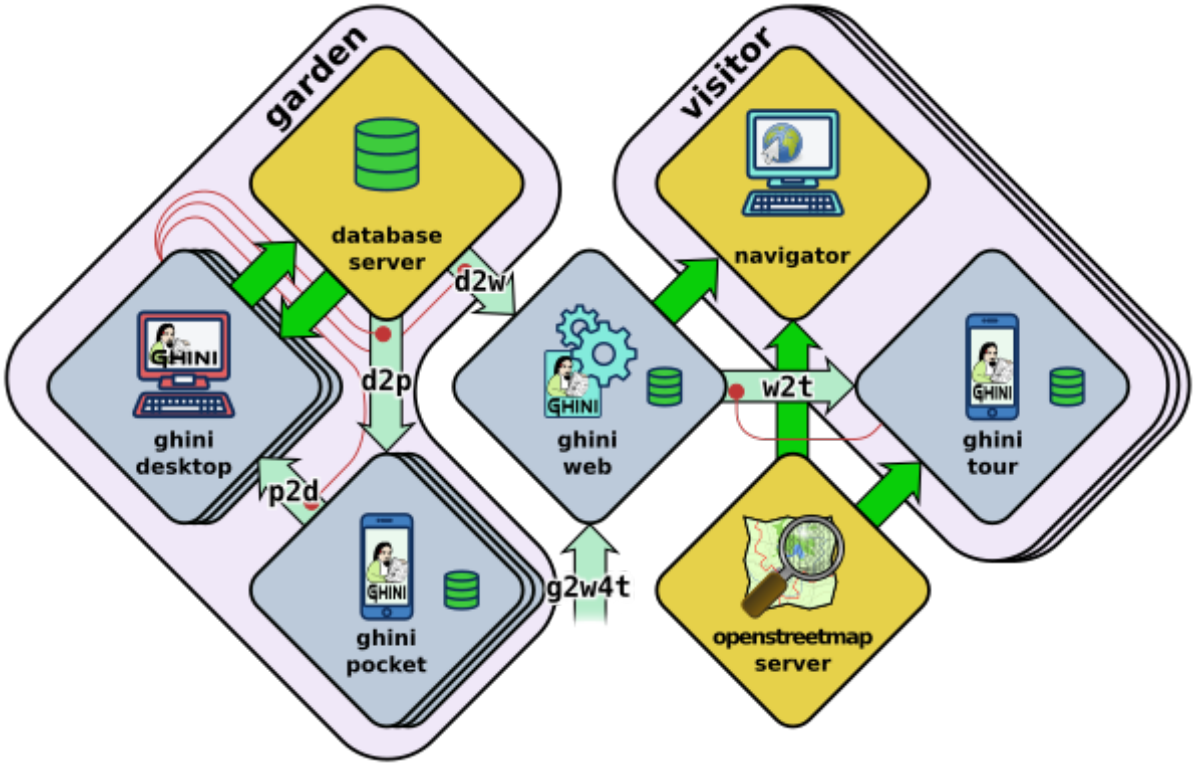
The reader getting to this point in the documentation probably understood that this Ghini project is above all a very open and collaborative project.

Here in this page you find some template letters, used to welcome new users, or that you can correct, print, and go with it to a garden, and propose them to adopt Ghini, or share with a group of your local friends, so you can make Ghini become a (voluntary, or paid) part-time job for you.

7.2.1 Dear conservator or scientist,

You are reading Ghini's presentation letter. Ghini is a libre software project on GitHub, focusing on botany. Brought to you by a small community of coders, botanists, translators, and supported by a few institutions around the world, among which, gardens that have adopted it for all their collection management needs.

The Ghini family is a software suite composed of standalone programs, data servers and handheld clients, for data management, and publication:



- Ghini's core, `ghini.desktop`, lets you
  - enter and correct your data
  - navigate its links,
  - produce reports
  - import and or export using several standard or ad-hoc formats
  - review your taxonomy using online sources

  all according best practices suggested by top gardens, formalized in standard formats like ABCD, ITF2, but also as elaborated by our developers, based on the feedback of Ghini users.

  `ghini.desktop` is developed and continously tested on GNU/Linux, but runs equally well on Windows, or OSX. [1]

- `ghini.pocket` is your full time garden companion, an Android app installed from the Play Store,
  - assisting you in collecting or correcting data while in the field,
  - associate pictures to your plants, and verify taxonomic information.
  - Import your collected data into the desktop client when back in the office,

  `ghini.pocket` reduces the time spent in front of your desktop PC to a true minimum.

- `ghini.web` is a web server and a courtesy data hub service, offering you world wide visibility: Export a selection of your data from your desktop database, and handle it for publication to the Ghini project, and we will include it at http://gardens.ghini.

me/, at no cost while we're able to do that, or for a guaranteed minimal amount of time if you are able to support our hosting costs. `ghini.web` serves a world map to help locate participating gardens, and within each garden, its contributed georeferenced plants.

- `ghini.tour`, a geographic tour Android app aimed at visitors, using OpenStreetMap as a base map, retrieving its data, gardens and virtual panels, from the web data aggregator `ghini.web`.

All software within the Ghini family is either licensed GNU Public License v2+ or v3+. It is a strong copyleft license. In short, the GPL translates the ethical scientific need to share knowledge, into legal terms. If you want to read more about it, please refer to https://www.gnu.org/licenses/copyleft.html

Ghini's idea about knowledge and software ownership is that software is procedural knowledge and as such, should be made a «commons»: With software as a commons, «libre software» and more specifically «Copylefted software», you not only get the source code, you receive the right to adapt it, and the invitation to study and learn from it, and to share it, both share forward to colleagues, and share back to the source. With proprietary software, you are buying your own ignorance, and with that, your dependency.

This fancy term «copyleft» instead of just «libre software», means the software you received is libre software with one extra freedom, guaranteeing every right you were granted upon receiving the software is never lost.

With copylefted software you are free —actually welcome— to employ local software developers in your neighbourhood to alter the software according to your needs, please do this on GitHub, fork the code, develop just as openly as the common practice within Ghini, and whenever you want, open a pull request so your edits can be considered for inclusion in the main branch. Ghini is mostly continuously unit tested, so before your code is added to the main branch, it should follow our quality guidelines for contributions. With libre software you acquire freedom and contribute to it, something that earns you visibility: Your additions stays yours, you share them back to the community, and will see them completed and made better by others. Having your code added to the main branch simplifies your upgrade procedure.

You can also contribute to the software by helping translate it into your native language. [5]

Some videos are published on YouTube, highlighting some of the software capabilities. [6]

Share back with the community. Several developers have spent cumulatively many thousand hours developing this software, and we're sharing with the community. We hope by this to stimulate a community sentiment in whoever starts using what we have produced.

Thanks for your consideration; please let me know if you have any questions,

In case you're interested in publishing your tree collection on the net, I would be happy to include your plants, species, coordinates to http://gardens.ghini.me. Georeferenced textual information panels are also very welcome, all offered as a courtesy: We're still

defining the offer. The idea behind this is allowing visitors to explore aggregated garden collections, and the current focus is on trees.

A small example: http://gardens.ghini.me/#garden=Jardín%20el%20Cuchubo

Mario Frasca MSc

[1] http://ghini.readthedocs.io/ - http://ghini.github.io/
[2] https://play.google.com/store/apps/details?id=me.ghini.pocket
[3] http://gardens.ghini.me/
[4] https://play.google.com/store/apps/details?id=me.ghini.tour
[5] https://hosted.weblate.org/projects/ghini/#languages
[6]
https://www.youtube.com/playlist?list=PLtYRCnAxpinU_8WEDuRlgsYnNVe4J_4kv

7.2.2 free botanic data management systems

Many institutions still consider software an investment, an asset that is not to be shared with others, as if it was some economic good that can't be duplicated, like gold.

As of right now, very few copylefted programs exist for botanic data management:

- `ghini.desktop`, born as `bauble.classic` and made a commons by the Belize Botanical Garden. `ghini.desktop` has three more components, a pocket data collecting Android app, a Node.js web server, aggregating data from different gardens and presenting it geographically, again a geographic tour app aimed at visitors using the web data aggregator as its data source. You can find every Ghini component on GitHub: http://github.com/Ghini
- Specify 6 and 7, made a Commons by the Kansas University. A bit complex to set up, very difficult to configure and tricky to update. The institutions I've met who tried it, only the bigger ones, with in-house software management capabilities manage to successfully use it. They use it for very large collections. Specify is extremely generic, it adapts to herbaria, seed collections, but also to collections of eggs, organic material, fossils, preserved dead animals, possibly even viruses, I'm not sure. It is this extreme flexibility that makes its configuration such a complex task. Specify is also on GitHub: https://github.com/specify and is licensed as GPLv2+.
- Botalista, a French/Swiss cooperation, is GPL as far as rumours go. Its development has yet to go public.
- `bauble.web` is an experimental web server by the author of `bauble.classic`. `bauble.classic` has been included into Ghini, to become `ghini.desktop`. Bauble uses a very permissive license, making it libre, but not copylefted. As much as 50% of bauble.web and possibly 30% of ghini.desktop is shared between the two projects. Bauble seems to be stagnating, and has not yet reached a production-ready stage.
- `Taxasoft-BG`, by Eric Gouda, a Dutch botanist, specialist in Bromeliaceae, collection manager at the Utrecht botanical garden. It was Mario Frasca who convinced

Eric to publish what he was doing, licensing it under the GPL, but the repository was not updated after 2016, April 13th and Eric forgot to explicitly specify the license. You find it on github: https://github.com/Ejgouda/Taxasoft-BG

- `BG-Recorder`, by the BGCI, runs on Windows, and requires Access. Developed mostly between 1997 and 2003, it has not been maintained ever since and isn't actively distributed by the BGCI. I've not managed to find a download link nor its license statement. It is still mentioned as *the free option* for botanic database management.

Of the above, only `ghini.desktop` satisfies these conditions: Copylefted, available, documented, maintained, easy to install and configure. Moreover: Cross platform and internationalized.

7.2.3 Welcome to Ghini/Bauble

Dear new user,

Welcome to Ghini/Bauble.

As the maintainer, I have received your registration for bauble.classic/ghini.desktop, many thanks for taking your time to fill in the form.

I see you are using bauble.classic-1.0.55, whereas 1.0.55 is the last released version of bauble.classic, however, bauble.classic is now unmaintained and superseded by the fully compatible, but slightly aesthetically different ghini.desktop. Install it following the instructions found at http://ghini.rtfd.io

The registration service says you're not yet using the newest Python2 version available. As of 2018-05-01, that is 2.7.15. Using any older version does not necessitate problems, but in case anything strange happens, please update your Python (and PyGTK) before reporting any errors.

Also thank you for enabling the «sentry» errors and warnings handler. With that enabled, Ghini/Bauble will send any error or warning you might encounter to a central server, where a developer will be able to examine it. If the warning was caused by an error in the software, its solution will be present in a subsequent release of the software

If you haven't already, to enable the sentry and warnings handler, open the «:config» page in Ghini and double click on the row «bauble.use_sentry_client».

I hope Ghini already matches your expectations, if this is not the case, the whole Ghini community would be very thankful if you took the time to report your experience with it.

The above is one way to contribute to Ghini's development. Others are: - contribute ideas, writing on the bauble google forum (https://groups.google.com/forum/#!forum/bauble), - contribute documentation, or translations (https://hosted.weblate.org/projects/ghini/), - give private feedback, writing to ghini@anche.no, - rate and discuss Ghini openly, and promote its adoption by other institutions, - open an issue on GitHub (https://github.com/Ghini/ghini.desktop/issues/), - contribute code on GitHub (fork the project on (https://github.com/Ghini/ghini.desktop/), - hire a developer and have a set of GitHub

issues solved, per-haps your own - let me include your garden on the still experimental worldmap (http://gardens.ghini.me)

I sincerely hope you will enjoy using this copylefted, libre software

Best regards, Mario Frasca

https://ghini.github.io https://github.com/Ghini/ghini.desktop/issues/

7.2.4 Do you want to join Ghini?

Примітка: I generally send a note similar to the following, to GitHub members who «star» the project, or to WebLate contributors doing more than one line, and at different occasions. If it's from GitHub, and if they stated their geographic location in their profile, I alter the letter by first looking on institutos botánicos if there's any relevant garden in their neighbourhood.

Dear GitHub member, student, colleague, translator, botanist,

Thank you warmly for your interest in the Ghini project!

From your on-line profile on github, I see you're located in Xxxx, is that correct?

If you are indeed in Xxxx, you live very close to gardens Yyyy and Zzzz. Maybe you would consider the following proposition? All would start by contacting the botanical garden there, and get to know what software they use (what it offers, and at which price) and if they're interested in switching to ghini.desktop+pocket+tour+web.

The business model within Ghini is that the software is free and you get it for free, but time is precious and if a garden needs help, they should be ready to contribute. Maybe you already have a full-time job and don't need more things to do, but in case you're interested, or you have friends who would be, I'm sure we can work something out.

Let me know where you stand.

best regards, and again thanks for all your contributed translations.

Mario Frasca

Розділ 8

Підтримка Ghini

If you're using Ghini, or if you feel like helping its development anyway, please consider donating.