icemac.addressbook Documentation

Release 9.4

Michael Howitz

Oct 07, 2020

Contents

1 Features **3**
- 1.1 General . . . 3
- 1.2 Data . . . 3
- 1.3 Search & Export . . . 3
- 1.4 Technical . . . 4

2 Screenshots **5**
- 2.1 persons found by keyword search . . . 5
- 2.2 list of persons stored in address book . . . 6
- 2.3 customization abilities for list of persons . . . 7
- 2.4 customization abilities for a postal address . . . 8
- 2.5 edit form for an address book user . . . 9
- 2.6 calendar of icemac.ab.calendar . . . 10

3 Download **11**

4 Installation **13**
- 4.1 Prerequisites . . . 13
- 4.2 Installation variants . . . 13
- 4.3 Update . . . 15

5 Run the tests **19**
- 5.1 Unit tests . . . 19
- 5.2 All tests . . . 19

6 Run the application **21**

7 Log-in into the application **23**

8 User management **25**
- 8.1 Roles . . . 25
- 8.2 Create new users . . . 25

9 Operations tips for the application **27**
- 9.1 Example Apache configuration . . . 27
- 9.2 The archive . . . 28

10 Uninstall **29**

11 Development **31**
11.1 Change log 31
11.2 To do 32
11.3 Hacking 32

`icemac.addressbook` is a multi user address book application accessible through the web: Store, edit, search and export addresses, phone numbers, ... using a web browser.

CHAPTER 1

Features

1.1 General

- multi-client capability
- user and role management
- Completely translated into German, and easily translatable into other languages.
- Optimized for the following browsers: Firefox, Safari.
- Calendar via icemac.ab.calendar.

1.2 Data

- store data of persons including postal address, phone number, e-mail address, home page address and files
- add custom data fields to persons and addresses in the master data area of the web user interface
- change the order of the fields of persons and addresses
- assign keywords to persons
- import data from XLS (Excel) or CSV files (via icemac.ab.importer resp. icemac.ab.importxls)
- archive persons

1.3 Search & Export

- search for persons by keywords and names
- export persons found using a search as XLS file
- export iCal calender file of the birth dates of persons

- update a single field of multiple persons as returned by a search (multi-update)

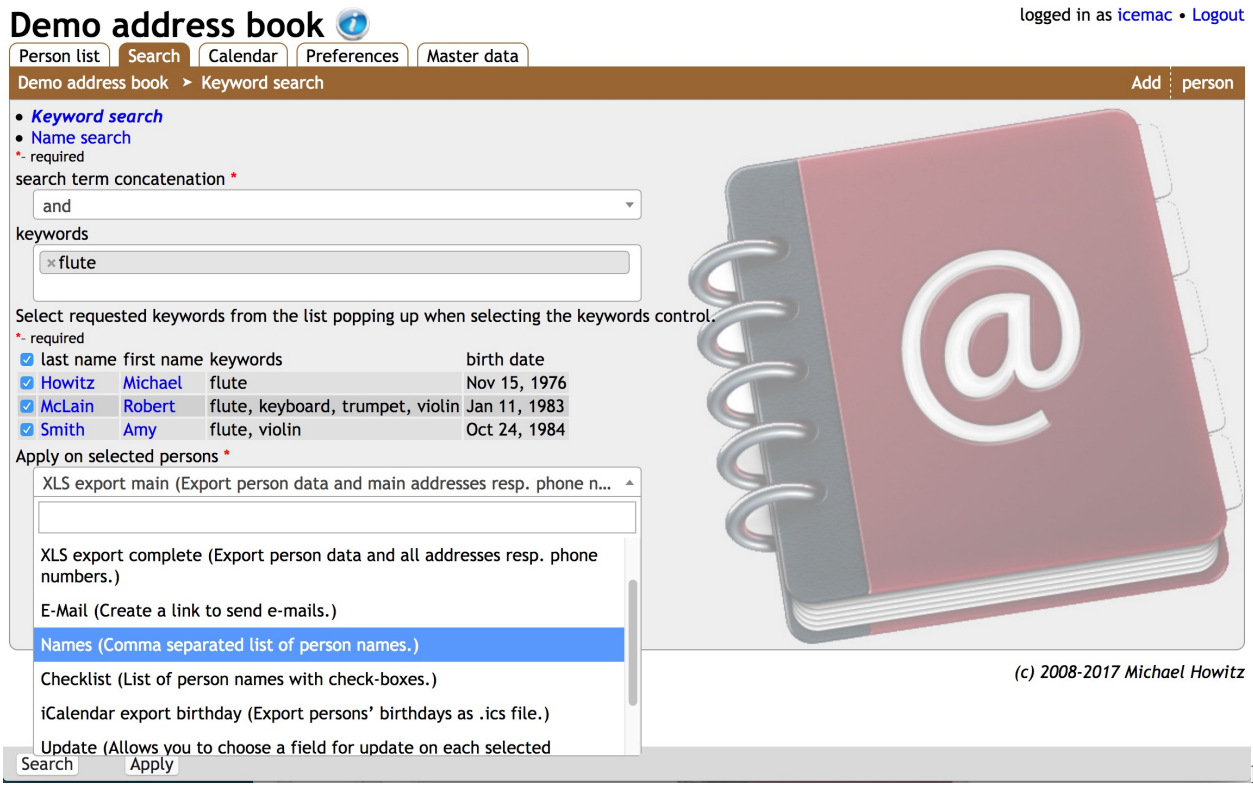
1.4 Technical

- 100 % test coverage of the program code including all possible program execution branches
- The data storage is an object database (ZODB); so no additional database is required.

CHAPTER 2

Screenshots

2.1 persons found by keyword search

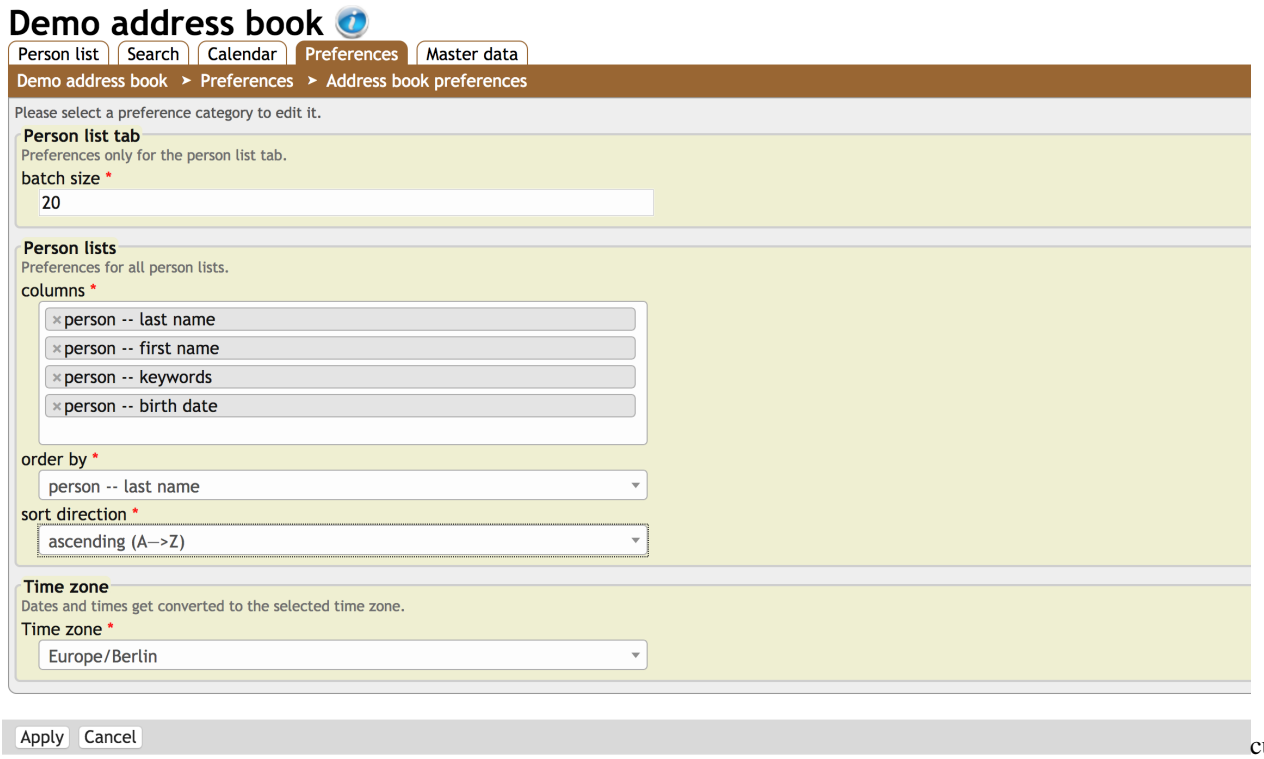


keyword search

2.2 list of persons stored in address book

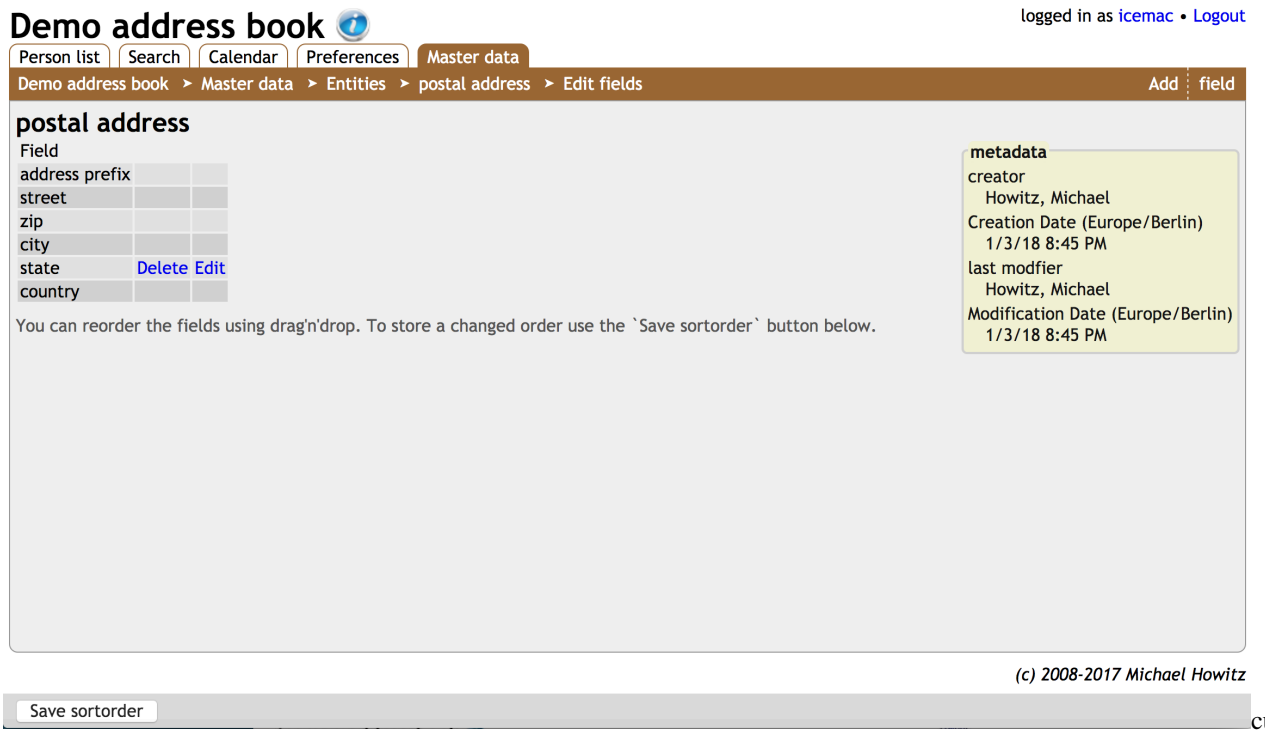
person list

2.3 customization abilities for list of persons



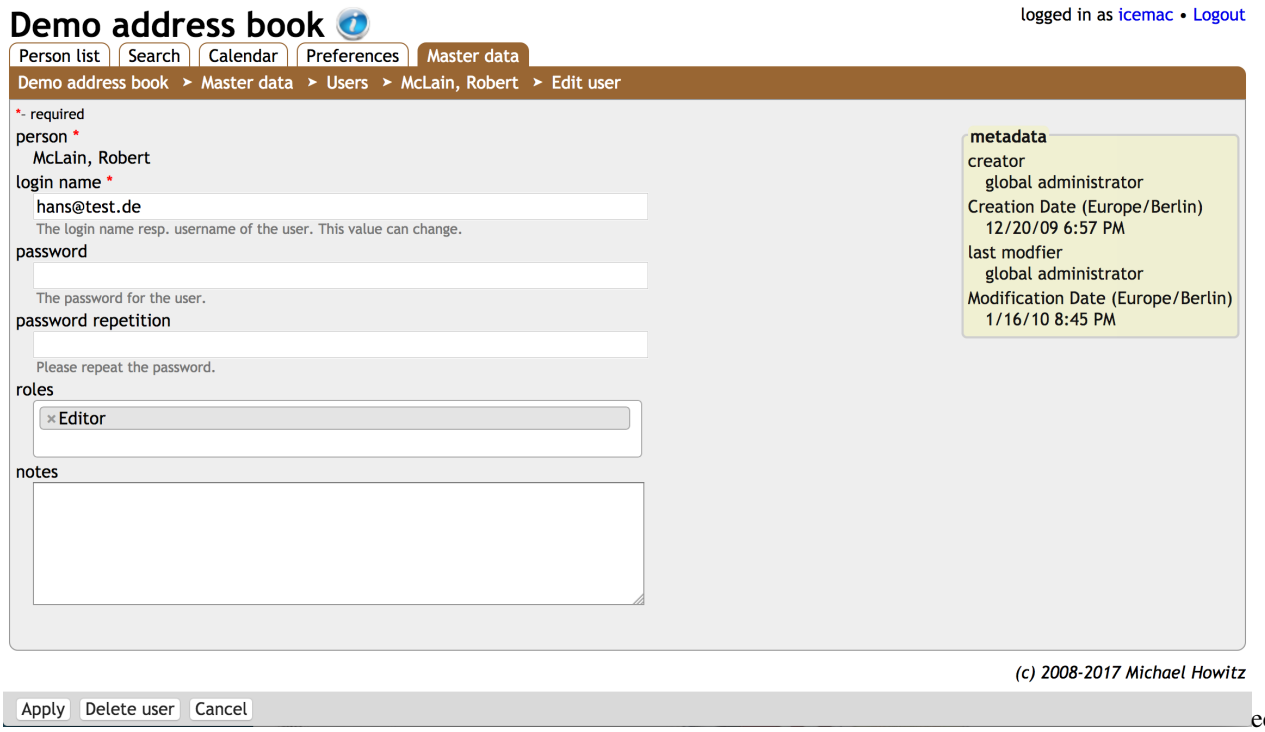
customization of person list

2.4 customization abilities for a postal address



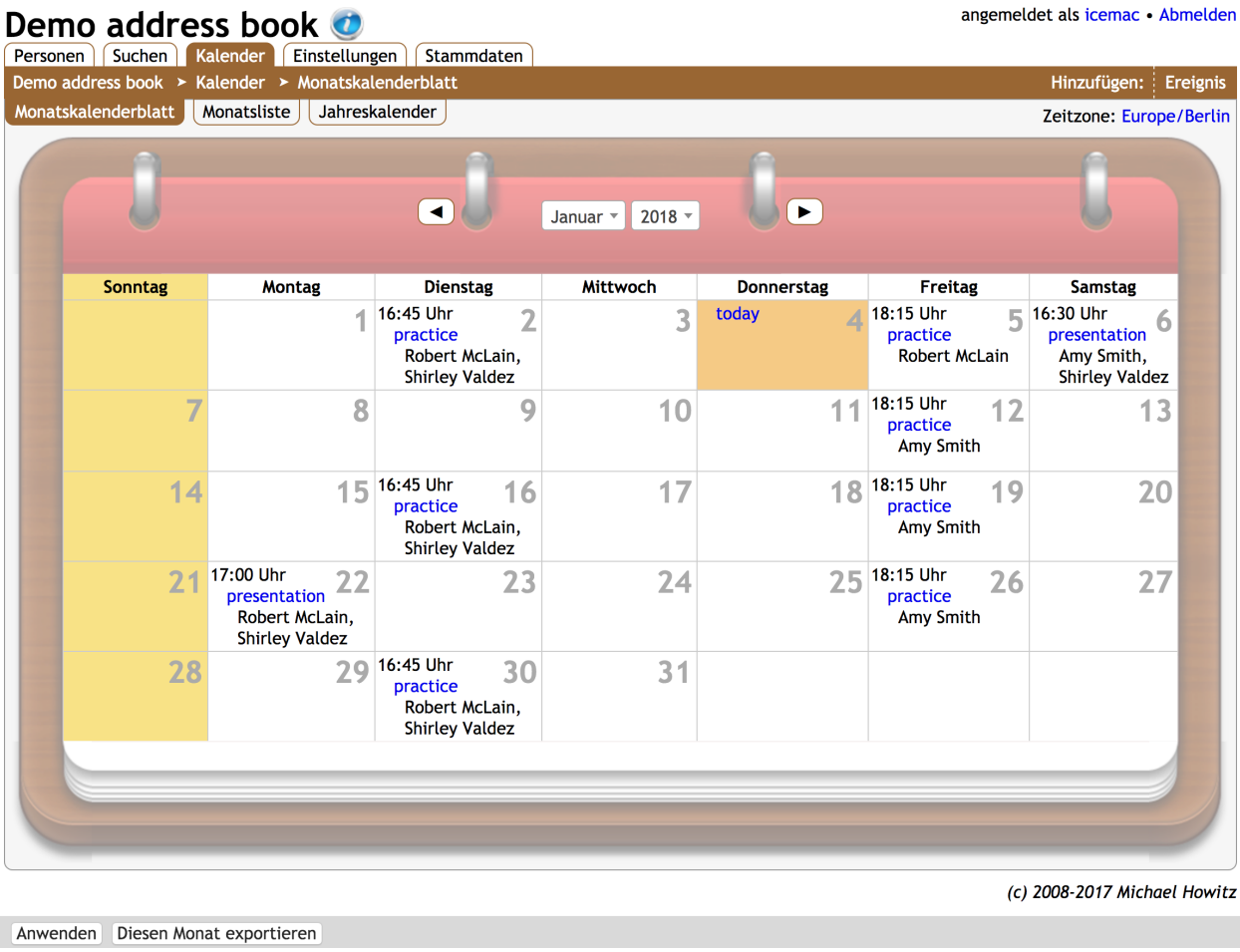
customization of addresses

2.5 edit form for an address book user



edit form

2.6 calendar of icemac.ab.calendar



calendar

CHAPTER 3

Download

Releases are uploaded to PyPI.

CHAPTER 4

Installation

4.1 Prerequisites

- You need:
  - CPython 2.7.x.
  - virtualenv
- Other Python versions resp. implementations are currently not supported.

4.2 Installation variants

There are two variants for the installation plus a deprecated one. (The deprecated installation variant will be removed in the next major release.)

4.2.1 Guided package installation

Follow these steps if you want to install the pre-packaged address book from PyPI. (This is the preferred way of the installation.)

Note: `icemac.addressbook` can't be installed directly using `easy_install` or `pip`, you have to follow these simple steps.

Neither you need any root privileges nor it installs anything outside its directory.

Installation steps:

1. Create a *virtualenv* using:

```
$ virtualenv-2.7 addressbook
```

2. Change into the directory of the *virtualenv* using:

```
$ cd addressbook
```

3. Install the installation script using:

```
$ bin/pip install icemac.install.addressbook
```

4. Start the installation
   - call:

     ```
     $ bin/install-addressbook VERSION_NUMBER
     ```

   - `VERSION_NUMBER` can be omitted to install the latest version or should be replaced by the desired version.
5. Answer the questions about admin user name, password and so on.
6. Change into the installation directory using:

```
$ cd current
```

7. *Run the tests*
8. *Run the application*
9. *Log-in into the application*
10. *Operations tips for the application*

Note: This installation variant installs the following a cron jobs:

- at each reboot start the address book
- each week pack the database
- each day create a backup of the database

You may want to add `MAILTO=you@your.domain` (replaching `you@your.domain` with your e-mail address) to recive reports from the cron jobs.

4.2.2 Upgrade installation from manual to guided installation variant

The following steps are necessary to upgrade an installation made with the deprecated manual installation method to the recent *Guided package installation* method.

1. Install *virtualenv*, see *Prerequisites*.
2. Make a new installation as described in *Guided package installation*. Including the creation of a new *virtualenv*. The answers to the questions about user name etc. do not matter. But using the same eggs directory speeds up the process. Do not try to migrate the old address book content to the new instance, yet.
3. Change into the installation directory using:

```
$ cd current
```

4. Remove the *buildout.cfg* created by the installation process:

```
$ rm buildout.cfg
```

5. To run the migration

   - call:

```
$ ../bin/python2.7 install.py PATH_TO_PREVIOUS_ADDRESSBOOK_INSTALLATION
```

   - `PATH_TO_PREVIOUS_ADDRESSBOOK_INSTALLATION` has to be replaced with the path to the installation directory of the address book (the one created when extracting the source distribution).
   - The questions during the installation should be pre-filled with the values from your previous installation.
   - Make sure to allow to migrate the old address book content to the new instance when asked during the installation process.

6. *Run the tests*
7. *Run the application*
8. *Log-in into the application*
9. If the installation was successful you can remove the old installation.

4.2.3 Source installation

This installation should only be used to work on the address book code itself.

1. Get the source code:

```
$ git clone https://github.com/icemac/icemac.addressbook
```

2. Install the sources:

```
$ cd icemac.addressbook
$ virtualenv-2.7 .
$ bin/pip install zc.buildout
$ printf "[buildout]\nextends = profiles/%s\n" dev.cfg > buildout.cfg
$ bin/buildout
```

3. *Run the tests*
4. To run the application you have to start the database (in demon mode) and the actual application in the foreground:

```
$ bin/zeoserver start
$ bin/addressbook
```

5. *Log-in into the application*

4.3 Update

4.3.1 Update guided package installation

This steps can be applied if you installed via *Guided package installation*.

1. Switch to the directory you created for the address book, in *Guided package installation* it was suggested to call it `addressbook`:

```
$ cd addressbook
```

2. Update the installer package using:

```
$ bin/pip install -U icemac.install.addressbook
```

3. Start the update

   - using:

   ```
   $ bin/install-addressbook VERSION_NUMBER
   ```

   - `VERSION_NUMBER` can be omitted to install the latest version.

4. Answer the questions about admin user name, password and so on, they should be pre-filled with the values of the previous installation.

   If the upgrade procedure breaks with an error, first try to upgrade to the next major version (first digit of the version number) which comes after your current version.

5. Start the new instance of the application as described in *Run the application*.

6. You might archive the previous version

   - by calling:

   ```
   $ bin/archive-addressbook VERSION_NUMBER
   ```

   - Where `VERSION_NUMBER` is the version number of the previous version.
   - It creates an archive of the installation folder and removes the installation folder afterwards.

4.3.2 Change configuration of the installation

This description can only be applied if you chose the *Guided package installation* method.

The values you entered during installation resp. update are stored in a file named `install.user.ini` in the `current` installation directory of the address book.

You can change the configuration values like this:

1. Switch to the directory you created for the address book, in *Guided package installation* it was suggested to call it `addressbook`:

```
$ cd addressbook
```

2. If you have made changes to the file `buildout.cfg` move it aside to be able to port the changes over afterwards. Otherwise just remove the file.

3. Call the script to change the configuration:

```
$ bin/change-addressbook-config
```

4. The configuration questions get re-presented to you with your previously entered values as defaults.

5. The address book instance has to be restarted which is done by the script if you want this. Otherwise see *Run the application*.

Caution: To remove additional packages you have to edit the `[package]` section of `install.user.ini`. There is currently no other way for removal. But be careful this might break your address books in the browser if the removed package(s) already stored data in the database. Afterwards call `bin/change-addressbook-config` as described above.

Caution: To remove the user name which should own the process you have to edit the `[server]` section of `install.user.ini`. Remove the value from the `user` line. Afterwards call `bin/change-addressbook-config` as described above.

4.3.3 Update source installation

This steps can be applied if you installed via the *Source installation* method.

1. Update via mercurial in the checkout directory:

```
$ hg pull -u
```

2. Run *buildout*:

```
$ bin/buildout
```

3. *Run the application*

CHAPTER 5

Run the tests

Running the tests is independent from your chosen installation kind but the prerequisites differ.

5.1 Unit tests

5.1.1 Prerequisites

If you have used the *Guided package installation* method you have to install the test scripts first:

```
$ cd current
$ ../bin/buildout install node pytest
```

5.1.2 Run the unit tests

Run the unit tests using:

```
$ bin/py.test -k "not webdriver"
```

5.2 All tests

These tests include some tests which require an actual browser to run.

5.2.1 Prerequisites for the browser tests

These prerequisites are only needed to run some tests in an actual browser.

Default prerequisites

- Fairly recent chromedriver version. It has to be installed in the search path.
- Google Chrome (a version matching your chromedriver version)

Alternative prerequisites

- geckodriver matching the `selenium` version used in the address book, see profiles/version.cfg resp. the file you are actually using. It has to be installed in the search path.
- Firefox (version as described by geckodriver)

5.2.2 Run all tests

Run all tests including the browser tests using:

```
$ bin/py.test
```

Caution: By default the browser does not show up, it runs in headless mode. To additionally show the actual browser during the tests, call them using:

```
$ NOT_HEADLESS=1 bin/py.test
```

To run the tests in Firefox, call them using:

```
$ SELENIUM_FIREFOX=1 bin/py.test
```

Firefox also supports the non-headless mode:

```
$ SELENIUM_FIREFOX=1 NOT_HEADLESS=1 bin/py.test
```

CHAPTER 6

Run the application

Running the application is quite independent from your chosen installation method.

The application is started and controlled by Supervisor. During the installation a *crontab* entry is generated which starts Supervisor at reboot of the machine the address book was installed on.

Note: All commands described here have to be called inside the directory called `current`.

To start Supervisor and all application processes by hand call:

```
$ bin/svd
```

To see which application processes are running call:

```
$ bin/svctl status
```

To start/restart/stop parts of the application use the the appropriate sub commands of `bin/svctl`.

To stop the application and Supervisor call:

```
$ bin/svctl shutdown
```

The next step is to *Log-in into the application*.

CHAPTER 7

Log-in into the application

The default URL to access the application is:

http://127.0.0.1:8080

(You may have changed the port number during the *Installation*.)

Caution: This URL is only accessible for a browser running on the same machine as the application. If you are installing it on an (external) server you might need to create a port forward to your local machine.

To log-in at this URL you need a user name and a password.

- If you installed via *Guided package installation* you had to chose both during the installation. They are stored in `admin.zcml`.
- If you installed via *Source installation* you find user name and password in `dev_admin_user.zcml`.

Create a new address book using the *add address book* link on the right. How to create new users inside this address book is described in *User management*.

CHAPTER 8

User management

8.1 Roles

Access to the address book is only granted after authentication. There are five roles to authorize a user:

- `visitor`: visit all person's data, search and export, change own password
- `editor`: permissions of `visitor` and additionally edit all person's data, delete resp. archive persons, change own log-in name
- `archive visitor`: visit persons and their data in the archive, change own password
- `archivist`: permissions of `archive visitor` plus ability to un-archive persons.
- `administrator`: permissions of editor and additionally create and change of address books, master data and users

8.2 Create new users

Users are persons from the address book augmented by log-in information.

To create the first new user inside an address book the administrator has to log-in and then do the following:

1. Create a new person with an e-mail address using the `Add person` link.
2. Create a new user using `Master data --> Users --> Add user`.

The newly created user has now a log-in (e-mail address) for the address book the person belongs to.

CHAPTER 9

Operations tips for the application

The application is not meant to be operated on a port open to the internet. It should run behind front end web server like Apache or nginx. The front end web server should handle the SSL handshake and rewrite the URL to the application server.

9.1 Example Apache configuration

For Apache you can use the following configuration. It has the following assumptions:

- The address book listens on port 13080 on *localhost*.
- The Apache runs handles HTTPS requests (port 443).
- The application will be bound to a sub domain, so all requests for the sub domain are redirected to the application.

```
ProxyRequests Off

<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

RewriteEngine on
RewriteRule ^(/?.*) http://127.0.0.1:13080/CONTAINERNAME/++vh++https:SUBDOMAINNAME.
↪DOMAIN.TLD:443/++//$1 [P,L]
```

CONTAINERNAME is the ID of the address book. *SUBDOMAINNAME.DOMAIN.TLD* is the FQDN for the target sub domain.

9.2 The archive

The archive contains archived people. They cannot be found any more by the person search, cannot be changed and are only visible in the archive for users with the role `archivist` or `archive visitor`. The `archive visitor` has read only access to the archive while the `archivist` is also able to un-archive a person which moves the person outside of the archive and makes it editable again.

Users with the role Editor are able to move persons to the archive.

9.2.1 Disable the archive function

If you don't need an archive of persons, and you want to disable the ability to archive a person, you only need to add the *Archive* tab to the list of deselected tabs in *Master data –> Address book*.

CHAPTER 10

Uninstall

If you want to uninstall the address book consider the following steps:

1. Switch to the directory you created for the address book, in *Guided package installation* it was suggested to call it `addressbook`:

```
$ cd addressbook
```

2. Stop the address book using:

```
$ bin/svctl stop all
$ bin/svctl shutdown
```

3. Remove the created cronjobs via:

```
$ crontab -e
```

4. Delete the whole installation directory.
5. Maybe remove the configuration made for the address book from your frontend server (Apache, ngnix, . . .).

Documentation in other languages:

- Deutsprachige Dokumentation (https://offb213.de)

CHAPTER 11

Development

11.1 Change log

11.1.1 9.4 (2020-04-21)

- Update most libraries needed for the address book to their newest versions.
- Port code to Python 3.8, supporting now additionally to Python 2.7.

11.1.2 9.3 (2020-03-23)

- Update most libraries needed for the address book to their newest versions.
- Adapt *print.css* to menu changes in version 9.2.
- Prevent entering date values before 1900 as they cannot be rendered, so they even cannot be changed.
- Move source code to: https://github.com/icemac/icemac.addressbook.

11.1.3 9.2 (2019-12-30)

- Update most libraries needed for the address book to their newest versions.
- Fix padding on log-in form.
- On mobile devices replace horizontal menus (tabs) and the add menu with vertical ones hidden behind a burger icon resp. plus sign.
- Make header of application in browser less tall and move link to the about view to the main menu.

11.1.4 Previous Versions

See OLD_CHANGES.rst.

11.2 To do

11.2.1 Next version

- Read only archive for persons.
- Maybe: Full text search.

11.2.2 Longer term

The following features might be implemented eventually:

- person data
  - store picture per person
- company as entity (?)
- import
  - vCard
  - SQLite (Apple's Addressbook.app)
- export
  - custom XLS export
  - LDAP export (?)
- more search abilities
  - full text search
  - combined search
  - wildcard search

11.3 Hacking

Fork me on: https://github.com/icemac/icemac.addressbook

File an issue on: https://github.com/icemac/icemac.addressbook/issues

Create the documentation using `tox`.

See *Source installation*, too.

11.3.1 Debug the application

If you disable *handle_errors* you get a full traceback for each exception type rendered on your console. To do so start the address book using:

```
$ HANDLE_ERRORS=0 bin/addressbook
```

Caution: This should not be used in a production environment!

Image credits: Martin Šnajdr