Hello!

You may or may not have heard about paperless [1]. It is a web application that manages y
our documents and is intented to be used in conjunction with a document scanner to get ri
d of your physical documents. The biggest advantage by far is that this database is searc
hable.

I’ve been using the project since 3 years now and its working fine. Iâ\200\231ve got abou
t 2000 scanned documents and the application has helped me countless times to find what I
need. It has its flaws but overall the system does what I need it to do. I’ve implemente
d my own share of extensions to the project in the past, some of which have made it into
the main repository. Since the project is kind of stalled on github, I decided to start e
xtending and maintaining the code myself.

In the past few months I’ve made some radical changes to the project, the detailed list i
s provided below. Have a look at some screenshots if youâ\200\231re interested:

Also, the dependencies of the project have been updated and now use the most recent versi
ons of django and associated libraries.

Interested in trying out the project? The recommended way of deployment is now docker-com
pose, the Docker files have been updated and work on Windows, Linux and Raspberry Pi (alt
hough the build on Pi is terribly slow).

Want to migrate to the new version? Since there are so many different ways to setup paper
less, youâ\200\231re a bit on your own. However, the migrations provided by paperless sho
uld take care of everything related to your data. The docker-compose files have changed,
however, they still expect the same volumes and the location of the data inside the volum
es has not changed.

Where do we go from now on? Im using the project and will ensure that it keeps running an
d receives fixes. Thatâ\200\231s one of the reasons I forked the project and worked on it
. I thought about pushing the changes back to the main project as well, however, the code
base introduces so many changes that I feel this is not possible anymore. I also want to
have full control over the source code.

Added

â\200¢ A new single page UI built with bootstrap and Angular. Its much more responsiv
e than the django admin pages.

â\200¢ Document uploading on the web page. This is very crude right now, but gets the
job done. It simply uploads the documents and stores them in the configured consumer dir
ectory. The API for that has always been in the project, there simply was no form on the
UI to support it.

â\200¢ Full text search with a proper document indexer: The search feature sorts docu
ments by relevance to the search query, highlights query terms in the found documents and
provides autocomplete while typing the query. This is still very basic but will see exte
nsions in the future.

â\200¢ Document types. Similar to correspondents, each document may have a type (i.e.
, invoice, letter, receipt, bank statement, ...). I’ve initially intented to use this for
some individual processing of differently typed documents, however, no such features exi
sts yet.

â\200¢ Inbox tags. These tags are automatically assigned to every newly scanned docum
ent. They are intented to be removed once you have manually edited the meta data of a doc
ument.

â\200¢ Automatic matching for document types, correspondents, and tags. A new matchin
g algorithm has been implemented (Auto), which is based on a classification model (simple
feed forward neural nets are used). This classifier is trained on your document collecti
on and learns to assign metadata to new documents based on their similiarity to existing
documents.

â\227¦ If, for example, all your bank statements for a specific account are tagge
d with "bank_account_1234" and the matching algorithm of that tag is set to Auto, the cla
ssifier learns relevant phrases and words in the documents and assigns this tag automatic
ally to newly scanned and matching documents.

â\227¦ This works reasonably well, if there is a correlation between the tag and
the content of the document. Tags such as ’TODO’ or ’Contact Correspondent’ cannot be ass
igned automatically.

â\200¢ Archive serial numbers. These are there to support the recommended workflow fo
r storing physical copies of very important documents. The idea is that if a document has

to be kept in physical form, you write a running number on the document before scanning (the archive serial number) and keep these documents sorted by number in a binder. If you need to access a specific physical document at some point in time, search for the document in paperless, identify the ASN and grab the document.

Modified

â\200¢ [breaking] API changes. In order to support the new UI, changes had to be made to the API. Some filters are not available anymore, why other filters were added. Furthermore, foreign key relationships are not expressed with URLs anymore, but with their respective ids. Also, the old urls for fetching documents and thumbnails are not valid anymore.

â\200¢

Removed (Mostly because I donâ\200\231t want to maintain the removed code)

â\200¢ [breaking] Reminders. I have no idea what they were used for and thus removed them from the project.

â\200¢ [breaking] Filename handling. The master branch of the paperless project has seen some changes regarding the filename handling of stored documents. These changes allow you to change the filename of stored documents from their default form â\200\230{id}.pdf â\200\231. These changes have not made it into this project. If you are using version 2.7.0, this does not affect you. If you are on the most recent push on the master branch, you need to make sure the filenames of your documents are not modified.

â\200¢ Every customization made to the admin interface. Since this is not the primary interface for the application anymore, there is no need to keep and maintain these. Besides, some changes were incompatible with the most recent versions of django. The interface is completely usable, though.

Planned (Will be added in the near future)

â\200¢ Saveable filters. Save filter and sorting presets and optionally display a couple documents of saved filters (i.e., your inbox sorted descending by added date, or tagged TODO, oldest to newest) on the dash board.

â\200¢ More search. The search backend is incredibly versatile and customizable. Searching is the most important feature of this project and thus, I want to implement things like:

- â\227¦ Group and limit search results by correspondent, show â\200\234more from thisâ\200\235 links in the results.
- â\227¦ Ability to search for â\200\234Similar documentsâ\200\235 in the search results
- â\227¦ Provide corrections for mispelled queries

â\200¢ More robust consumer that shows its progress on the web page.

On the chopping block (code that might get removed in the future).

â\200¢ GnuPG encrypion. Since its disabled by default and the website allows transparent access to encrypted documents anyway, this doesnâ\200\231t really provide any benefit over having the application stored on an encrypted file system.

â\200¢ Email scanning. I donâ\200\231t use it and donâ\200\231t know the state of the implementation. Iâ\200\231ll have to look into that.