Plep Documentation

Release stable

Feb 06, 2019

Contents

Plepping is the new planning.

Plep works and looks like a real agenda, but with the advantages of having it on your computer.

CHAPTER 1

Downloads

Latest stable version: Latest pre-release version:

We try to adhere to Semantic Versioning. An update with a new first number means it won't work with your old database.

CHAPTER 2

Features

- Drag and drop
- Custom labels
- Subtasks
- Custom colours
- Repeat for x weeks
- Customize number of days and columns
- Turn pages by a custom amount of days
- Deletion and undoing deletion
- Plep will remember all deleted tasks until you close it.

Note: The first time plep is ran, it will create a database file in the folder where the jar is stored. Check for this file, otherwise your data won't be saved. You can backup this if you don't want to lose your data.

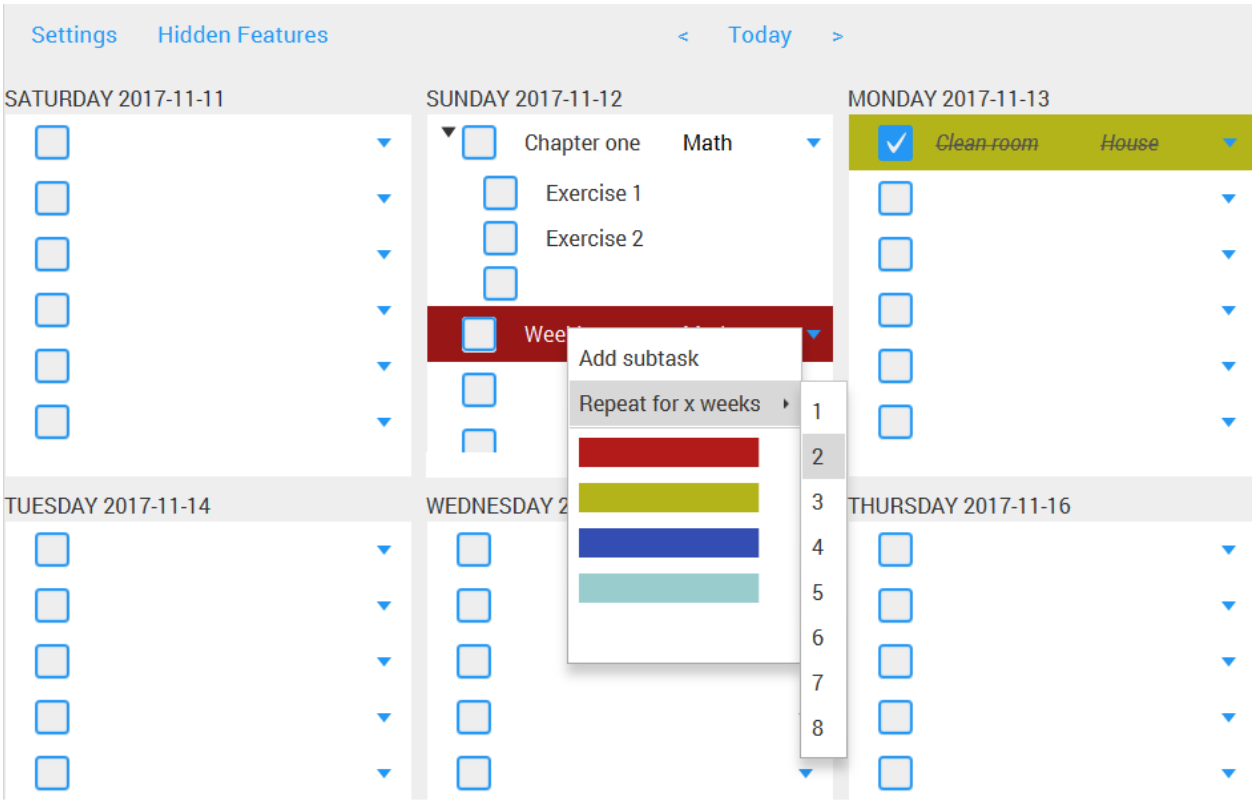
CHAPTER 3

Updating Plep on Linux

Because of file permissions, an empty database with the correct permissions is created in `/usr/lib/plep/lib/plep.db` when the package is installed. If you want to reuse an old database, backup that first and then replace it again (e.g. with `cp -a --remove-destination plep_backup.db /usr/lib/plep/lib/plep.db`).

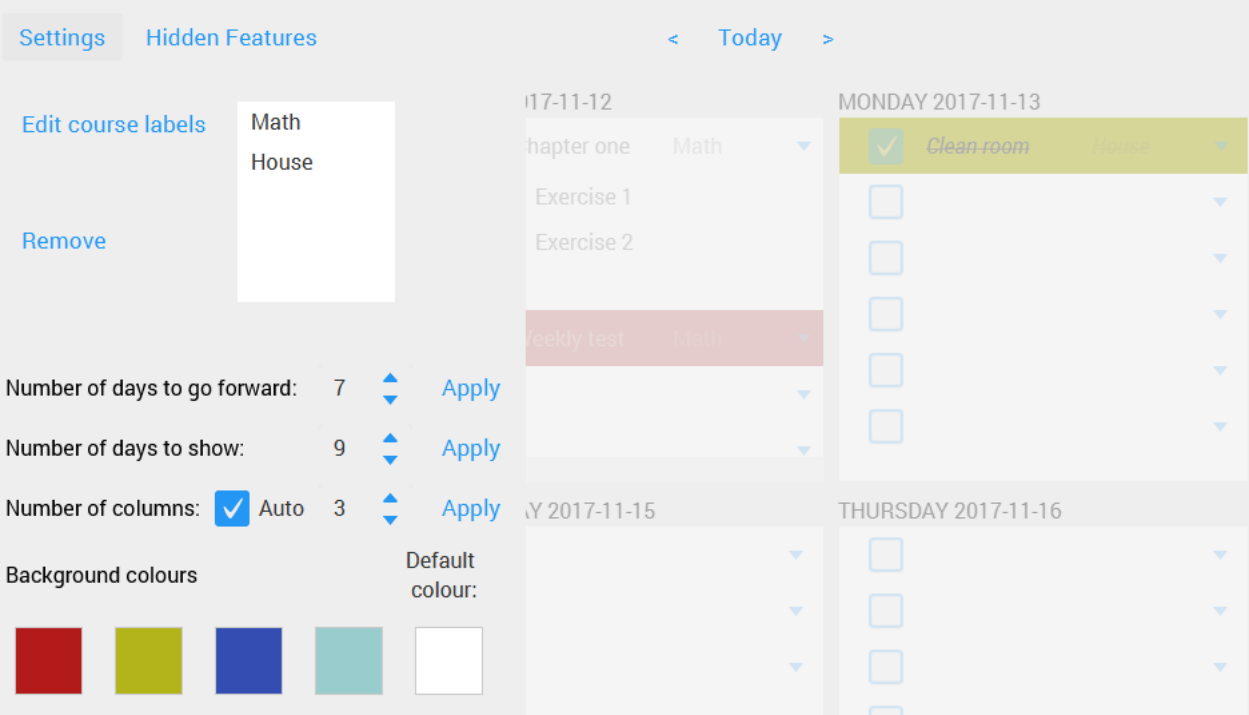
CHAPTER 4

Screenshot of v2.0.0



CHAPTER 5

Settings of v2.0.0



CHAPTER 6

Instructions for building from source in IntelliJ

To install java on Fedora, install the `java-openjdk-devel` package (the regular `openjdk` only contains the jre, and not the jdk).

Make sure you have JavaFX installed, for example on Arch Linux and Fedora you can install the `java-openjfx` package. If *javafx* still cannot be resolve, delete and re-add the SDK.

- To run or debug, use the Gradle task `run`.
- To build an executable jar, use the task `build`, the file is then in `build/libs/`.
- To run tests from IntelliJ, install the Spek (and possibly also Spek Framework) plugin.

CHAPTER 7

Building a Windows release

- Update the version number in `build.gradle.kts`
- Run the gradle task launch4j/createExe
- Open InnoSetup on the setup file in `releasing/Windows/innoSetup/PlepInstaller.iss`
- Update the version number
- Click Build | Compile, the generated installer will be next to the setup file.
- If Plep doesn't start, some exception was thrown. Try the Gradle task 'shadowJar' and execute the jar file generated in build/libs with *java -jar plep-vx.x.x-all.jar*.

CHAPTER 8

Building an Arch Linux release

- Update the version number in `PKGBUILD` and `build.gradle.kts`
- Make sure you have everything pushed (you don't have to have the `PKGBUILD` pushed, but all the other build files)
- Go to `releasing/arch-linux`
- Run `makepkg -f` or to only repackage `makepkg -f --repackage`
- The generated package file will be `plep-vx.x.x-1-any.pkg.tar.xz`
- Check that it contains the right files with `pacman -Qlp plep-vx.x.x-1-any.pkg.tar.xz`
- Install with `sudo pacman -U plep-vx.x.x-1-any.pkg.tar.xz`

CHAPTER 9

Building a Fedora release

- Make sure you have the `rpm-build` and `rpmdevtools` packages installed.
- Compress the sources into a `tar` file, using (in the `plep` directory): `tar czf ~/rpmbuild/SOURCES/plep-x.x.x.tar.gz file1 file2 file3 ... filen`. You should include (at least) the following files/directories: - `./gradle` - `./releasing/fedora` (it contains the `plep.desktop` file) - `./src` - `build.gradle.kts` - `gradlew` - `gradlew.bat` - `LICENSE` - `settings.gradle.kts`
- Update the version number in `plep.spec` and `build.gradle.kts`.
- Go to `releasing/fedora`.
- Run `rpmbuild -bb plep.spec` to create the rpm file. This will create the rpm file in `~rpmbuild/RPMS/noarch/`.
- Install the rpm with `rpm -Uvh plep-x.x.x-y.fc29.noarch`, where `x.x.x` is the version number, and `y` the release number.

CHAPTER 10

Javadoc (obsolete)

Badges with thanks to shields.io.