

How To Liederbuch

10.02.2025

Table of contents

Moin.	3
Was können wir dir bieten?	3
Wer „wir“ sind	4
1 Vorbereitung	5
I Einrichtung von LilyPond	6
LilyPond installieren	7
II Einrichtung von Frescobaldi	8
Frescobaldi installieren	9
Beim ersten Ausführen	9
III Einrichtung von Git	10
Account bei tuxgit erstellen lassen	11
Eine Git-GUI aussuchen und herunterladen	11
SmartGit	11
SourceTree	11
TortoiseGit	12
Einrichten eines Liederbuch-Repos	12
SSH-Key erstellen	12
Ein neues Liederbuch-Repo mit dem Vorlagen-Repo erstellen	13
Ein bestehendes Liederbuch-Repo klonen	13
2 Nützliche Lifehacks	15
2.1 Frescobaldi-Lifehacks	15

Moin.



Schön, dass du hergefunden hast! In diesem Leitfaden zeigen wir dir, wie du mit [LilyPond](#) und [Git](#) Liederbücher anhand unserer Vorlagen und Skripte erstellen kannst. Wir versuchen dich dabei bestmöglich an die Hand zu nehmen, um die Hürden, die es dabei gibt, möglichst niedrig zu halten. Zu diesem Zwecke gibt es diesen Leitfaden, aber auch eine Gruppe auf Signal, in der sich alle Personen befinden, die mit unserem System Liederbücher machen.

Was können wir dir bieten?

Eine kleine Übersicht über Vorteile, die unser System mit sich bringt. Wenn dir davon nicht alles was sagt, ist das gar nicht schlimm.

- Du kannst Liederbücher mit Noten und ohne Noten erstellen. Du kannst außerdem entscheiden, ob du Akkorde über den Strophen haben möchtest oder nicht.
- Du benötigst dafür keine kostenpflichtigen Programme. Alle Programme, die wir nutzen, sind Free and Open Source Software.
- Du kannst davon profitieren, dass wir in unserem Liederpool schon etliche Lieder gesetzt haben.
- Durch unsere Struktur profitierst du von Fehlerkorrekturen anderer Liederbuchprojekte.
- LilyPond ist ein codebasiertes Notensatzprogramm, das bedeutet, man kann sehr gut ein Versionierungs- und Nachverfolgungssystem mit Git nutzen.

Wer „wir“ sind

Wir sind ein Zusammenschluss an verschiedenen Liederbuchprojekten, die alle LilyPond nutzen. Irgendwann kamen wir an den Punkt, dass es cool wäre, wenn man mehr Synergien untereinander hätte. Daraufhin haben wir das Konzept unseres Liederpools entwickelt, in dem sich nur der musikalische Inhalt eines Liedes befindet, den man dann in seinem Liederbuchprojekt spezifisch layouten kann. Das hat zwei riesige Vorteile: Einerseits kann man von dem profitieren, was andere schon gesetzt haben, und andererseits kann man von den Fehlerkorrekturen anderer profitieren.

So, jetzt genug der langen Vorrede. Los geht's!

1 Vorbereitung

Zuerst richten wir alle Programme ein, die du brauchst. Du benötigst als erstes LilyPond und Frescobaldi. Dann Git und dann einen Account bei dem Git-Server von tux.

Wir erklären dir jetzt Schritt für Schritt, was du dafür tun musst.

Part I

Einrichtung von LilyPond

LilyPond ist die Software, die wir zum Noten setzen benutzen. Als erstes müssen wir LilyPond bei dir installieren.

LilyPond installieren

1. Gehe auf <https://lilypond.org/> und wähle unter **Downloads** die neueste LilyPond-Version. Grundsätzlich ist immer die stabile Version zu bevorzugen (alle geraden Zahlen, z.B. 2.24). Unser System funktioniert aber (unter Windows) erst ab der Version 2.25.24, das ist eine Entwicklerversion (zu erkennen an der ungeraden 25).
2. Nun sollte der Download starten. Gehe in den Ordner und entpacke den Download. Herauskommen sollte ein Ordner namens `lilypond-*`, wobei `*` die Versionsnummer darstellt.
3. Fertig. Allerdings solltest du den Ordner an einen Ort legen, an dem er nicht verschoben wird. Zu empfehlen ist
 - unter Windows: Ins Verzeichnis `C` und dann in den Ordner `Programme (x86)`. Dort sind auch andere Programmordner, deswegen ergibt das dort Sinn.
 - unter Linux: Zum Beispiel ins Verzeichnis `home`.

Jetzt hast du LilyPond installiert. Damit es jetzt auch Spaß macht, mit LilyPond zu arbeiten, benötigst du noch Frescobaldi.

Part II

Einrichtung von Frescobaldi

Frescobaldi ist eine sogenannte „Integrierte Entwicklungsumgebung“ (IDE). Das bedeutet, sie schafft ein angenehmes Arbeitsumfeld, wenn du mit LilyPond arbeitest. LilyPond selbst ist nämlich quasi nur die „Sprache“ in der du Noten setzt und das Programm, womit du sie auswertest („setzt“). Frescobaldi vereint dann unter anderem einen PDF-Viewer, einen MIDI-Player und noch ein paar andere nützliche Werkzeuge.

Frescobaldi installieren

1. Gehe auf <https://frescobaldi.org/> und wähle unter **Download** die neueste Version für dein Betriebssystem. In eigentlich allen Linux-Distributionen sollte Frescobaldi zudem in der Paketverwaltung zu finden sein.
2. Du wirst auf eine Github-Seite weitergeleitet, wo du unter **Assets** die richtige Version auswählen kannst. Für Windows wähle ***.exe**, für Linux ***.tar.gz** und für MacOS **.dmg**.
3. Lade die Datei herunter und führe sie aus.
4. Folge dem Installationsprozess.

Beim ersten Ausführen

1. Gehe in die Einstellungen.
2. Unter Lilypond-Einstellungen musst du die korrekte Lilypond-Version einstellen. Gehe dazu auf **Hinzufügen** und wähle dann im Explorersfenster die ***.exe**-Datei aus. Dazu musst du in den Ordner gehen, in dem du den LilyPond-Ordner eben abgelegt hast (siehe oben), und die Datei **bin/lilypond.exe** auswählen.
3. Außerdem ist die Einstellung **Dokument speichern, wenn möglich unter Lilypond-Einstellungen** und dort **Lilypond starten** sinnvoll, damit man gleich kompilieren kann, ohne dauernd speichern zu müssen.
4. Eine weitere sinnvolle Einstellung ist unter **Allgemeine Einstellungen** im Reiter **Saving** die Einstellung **Leerzeichen am Zeilenende entfernen**. Es macht Sinn, dort einen Haken zu setzen.
5. Falls du Linux nutzt, ist es wichtig, dass du jeweils vorm Starten von Frescobaldi einen MIDI-Player startest. Was du dafür tun musst, ist in diesem [GitHub-Issue](#) erklärt.
6. Das war's schon. Fertig!

Part III

Einrichtung von Git

Hier geht es jetzt nur um eine kurze und möglichst unkomplizierte Einrichtung von Git. Du musst gar nicht alles verstehen, was im Hintergrund passiert. Das kann an späterer Stelle irgendwann auch noch kommen.

Account bei tuxgit erstellen lassen

Du brauchst einen Account für unseren Git-Server. Aufgrund von Spam ist die Registrierung nicht öffentlich möglich, deswegen musst du dafür eine E-Mail an tux schreiben. Die Web-Adresse unseres Git-Servers ist: <https://git.zahlenlabyrinth.de/>.

Eine Git-GUI aussuchen und herunterladen

Git ist per se erstmal ein Kommandozeilen-Programm. Das ist nicht für jede*n was – erst recht nicht, wenn man noch nicht so viel mit Coden zu tun hatte. Das ist aber gar nicht schlimm, denn es gibt zahlreiche verschiedene grafische Oberflächen für Git, die das Arbeiten sehr erleichtern können. Verbreitet bei uns sind gerade diese hier:

SmartGit

- [Link](#)
- für Windows, MacOS, Linux
- kostet was, es gibt aber eine non-commercial Lizenz für gemeinnützige Organisationen und Bildungseinrichtungen: [Apply for Non-Commercial License](#)
Dafür brauchst du eine gemeinnützige Organisation, mit deren E-Mail-Adresse du dich dort registrieren kannst. Alternativ geht auch eine Uni-Mail-Adresse.

SourceTree

- [Link](#)
- für Windows, MacOS

TortoiseGit

- [Link](#)
- für Windows

Oder man nutzt es in der Konsole. Dann braucht man die Git Shell: [Link](#).

Einrichten eines Liederbuch-Repos

Da du Git vermutlich zum ersten Mal nutzt, brauchst du zuerst einen SSH-Key. Der SSH-Key ist dafür da, dass das Remote-Repo weiß, dass dein lokales Repo auch zu dir gehört. Sonst könnte ja jeder, der den SSH-Link hat (siehe dazu Figure 1.1), Zugriff aufs Repo haben. Er ist also ein Identifikationsmittel für deinen Account. Dafür hinterlegst du den *public key* in deinem Git-Account und sagst deinem Git-Programm (SmartGit, SourceTree, TortoiseGit), wo deine Schlüssel-Datei liegt. Diese Datei ist auch nochmal extra passwortgeschützt. Sie sollte auf keinen Fall gelöscht oder verschoben werden. Die lange Erklärung kannst du [hier](#) nachlesen.

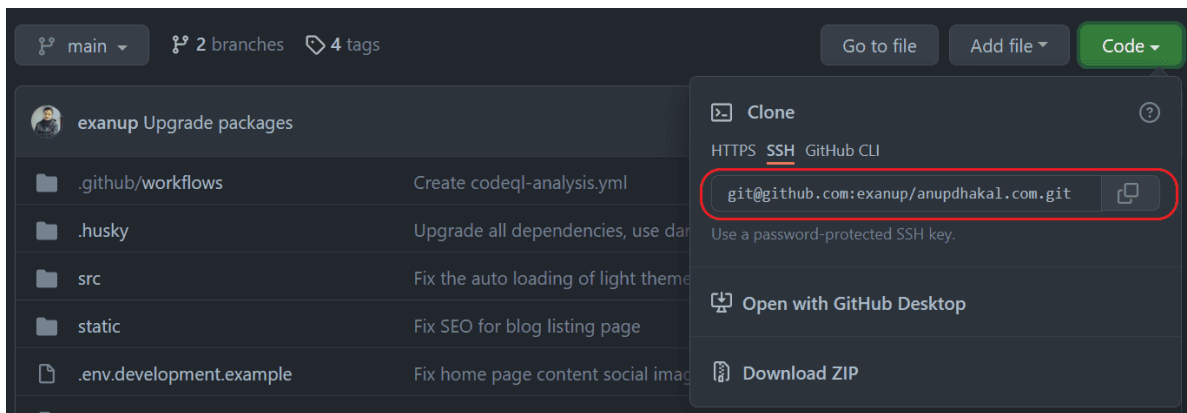


Figure 1.1: Das rot markierte ist ein SSH-Link.

SSH-Key erstellen

Du musst dir einen SSH-Key erstellen und das **Passwort**, *public key* und *private key* und **Schlüsseldatei** speichern/aufbewahren. Eine Anleitung dafür findest du [hier](#).

Anschließend hinterlegst du den *public key* in deinem Git-Account (unter Einstellungen -> SSH- / GPG-Schlüssel -> Schlüssel hinzufügen).

Später wirst du beim „Klonen“ eines Liederbuchs irgendwann nach deiner Schlüsseldatei gefragt (für die du dann auch das Passwort brauchst). Es ist also wichtig, dass du beides gut speicherst!

Klingt alles sehr kompliziert und das ist es auch, **aber**: Du musst das **nur einmal** machen und dann funktioniert es einfach. Wenigstens etwas.

Ein neues Liederbuch-Repo mit dem Vorlagen-Repo erstellen

Da du ja ganz neu einsteigst, hast du vermutlich noch kein eigenes Liederbuch-Repo. Deswegen erstellen wir erstmal eins. Das tolle ist: Wir haben zu genau diesem Zweck ein Vorlagen-Repo gebaut, mit dem du sofort die meisten wichtigen Sachen eingerichtet hast. Wichtig: Du brauchst *vorher* einen Account bei tuxgit (siehe [?@sec-tuxgit-account](#)). So geht's:

1. Du gehst zu unserem [LilyPond-Vorlagen-Repo](#) auf unserem Git-Server.
2. Du klickst auf **Dieses Template verwenden**.
3. Du folgst den Schritten (Besitzer*innen, Name, ... festlegen) Wichtig: Bei Template muss `lilypond-vorlagen` ausgewählt sein.
4. Dann kannst du einfach unten weitermachen.

Ein bestehendes Liederbuch-Repo klonen

Wenn du die obigen Schritte befolgt hast oder schon ein Liederbuch-Repo eingerichtet hast und dass einfach nur klonen willst, dann geht das so:

1. Remote-Repo klonen
Je nach GUI funktioniert das unterschiedlich. Irgendwo gibt es die Option **Clone** und da dann einen Dialog, in dem du unter anderem den SSH-Link des Repos eintragen kannst. Wichtig ist bei dem Prozess: Submodule müssen inkludiert werden. Irgendwo kann man dann den Speicherort des lokalen Repos festlegen. Es empfiehlt sich, einen Ordner zu haben, wo alle Git-Repos drin sind. Dies ist aber nicht notwendig für irgendetwas.
 - SmartGit: In Smartgit navigierst du zu Repository -> Clone und folgst dem Dialog. Zuerst gibst du deinen SSH-Link ein. Dann achtest du darauf, dass Submodules inkludiert werden und wählst zuletzt den Speicherort des Repos aus.
2. Als nächstes müssen die Submodule noch richtig eingebunden werden, das geht nämlich leider nicht automatisch. Du brauchst auf jeden Fall unsere [Common-Includes](#) als Submodul. Wenn du mit unserem Liederpool arbeitest, brauchst du noch die [Song-Includes](#) als Submodul. Für jedes Submodul musst du diesen Prozess einzeln durchführen. Auch das funktioniert in jeder GUI anders. So geht es bei SmartGit:

1. Im Menü Remote anwählen -> Submodule -> Add
 2. Dann brauchst du die SSH-Links der Submodule. Die findest du auf die bekannte Art und Weise (siehe Figure [1.1](#)).
 3. Im nächsten Schritt wird gefragt, an welcher Stelle das Submodul eingebunden werden soll, da kannst du einfach weiterklicken.
3. Fertig!

Und jetzt kann's richtig losgehen!

2 Nützliche Lifehacks

2.1 Frescobaldi-Lifehacks

- Du kannst die Silben eines Textes automatisiert trennen: Text auswählen und unter **Werkzeuge/Musical Transformations/Liedtext/Trennungsstriche in Text einführen** oder einfach **Strg + L**. Achtung! Die Funktion funktioniert leider nicht absolut fehlerfrei, deswegen definitiv nochmal Korrektur drüber lesen.
- Du kannst Noten ganz einfach transponieren, was in diesem Falle meint, dass sich der tatsächliche Code verändert: entsprechenden Code auswählen unter **Werkzeuge/Musical Transformations/Tonhöhen/Transponieren**. Dazu wichtig: Abschnitt auswählen und im Dokument vorher *temporär* `\language "deutsch"` aufrufen.
- Auch Notenlängen Halbieren oder Verdoppeln geht sehr einfach: entsprechenden Code auswählen **Werkzeuge/Musical Transformations/Rythmus/** und dann auf **Notenlänge verdoppeln** bzw. **Notenlänge halbieren**.
- Ebenfalls kann eine bestimmte rhythmische Form übernommen werden: **Werkzeuge/Musical Transformations/Rhythmus/**.
- Lange Zeilen können im Editor standardmäßig umgebrochen werden: Unter **Bearbeiten/Einstellungen/Editor-Einstellungen/Einstellungen zur Ansicht** -> Haken setzen.
- Zahlreiche LilyPond-Codeschnipsel gibt es unter **Schnipsel**. Das wollen wir künftig noch mehr ausbauen.