

Setup Sovrin's Indy-SDK for MacOS

Sovrin is a non-profit organization building an internet identity system that is freely available to everyone. Rather than hosting sensitive data in one location, Sovrin uses blockchain technology to keep all identity at the control of its owners, not by corporations or governments.

Many computer programs are being developed that make use of Sovrin's Identity Network. To use these programs, you must first have Indy-SDK installed locally on your computer. Indy-SDK is the library of code that other applications use to connect to the blockchain.

Windows and Linux users can simply download an installer program from the internet. However, a self-installing program is not yet available for Mac users. As such, this guide details how to manually install the Indy SDK for macOS.

Time Required: 10 minutes

Instructions

1. Open the Terminal

The terminal is a simple but powerful program preinstalled on all macOS computers. It allows the user to do anything that can be done with with an Internet browser, text editor, or other programs, but everything is instead presented with a textual interface.

The terminal may look daunting at first, but everything you need to type into it is explained below.

Open the terminal by searching for `Terminal` in the Applications/Utilities folder. A simple prompt should open up like this:



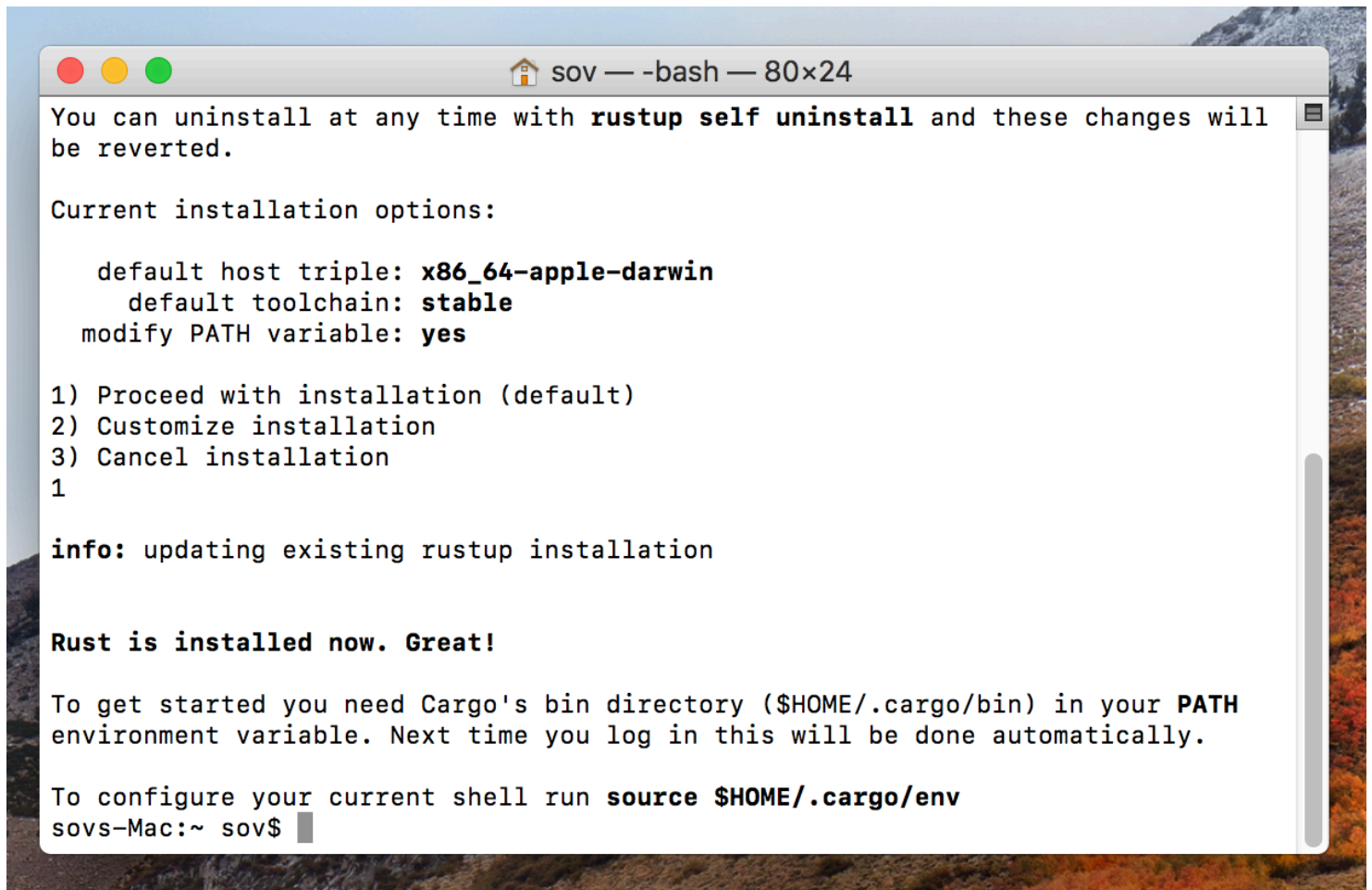
2. Install Rust

Rust is the computer programming language that Indy is written in. In order to install Indy, Rust first needs to be on the computer to interpret the program.

Copy the line of code below, paste it in the terminal, and hit `Enter` :

```
curl https://sh.rustup.rs -sSf | sh
```

When prompted, type in a `1` (Proceed with installation) and hit `Enter` .

A screenshot of a macOS terminal window titled 'sov - bash - 80x24'. The window shows the output of the 'rustup' command. It starts with a message about uninstalling, followed by 'Current installation options:' and a list of defaults: 'default host triple: x86_64-apple-darwin', 'default toolchain: stable', and 'modify PATH variable: yes'. A numbered list shows three options: '1) Proceed with installation (default)', '2) Customize installation', and '3) Cancel installation'. The user has entered '1'. The terminal then shows 'info: updating existing rustup installation' and 'Rust is installed now. Great!'. It provides instructions on setting the PATH environment variable and configuring the shell. The prompt at the bottom is 'sovs-Mac:~ sov\$'.

```
sov — -bash — 80x24
You can uninstall at any time with rustup self uninstall and these changes will
be reverted.

Current installation options:

    default host triple: x86_64-apple-darwin
    default toolchain: stable
    modify PATH variable: yes

1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
1

info: updating existing rustup installation

Rust is installed now. Great!

To get started you need Cargo's bin directory ($HOME/.cargo/bin) in your PATH
environment variable. Next time you log in this will be done automatically.

To configure your current shell run source $HOME/.cargo/env
sovs-Mac:~ sov$
```

Before continuing, some changes require the terminal to restart. Do so by exiting and then reopening the terminal.

3. Install Homebrew

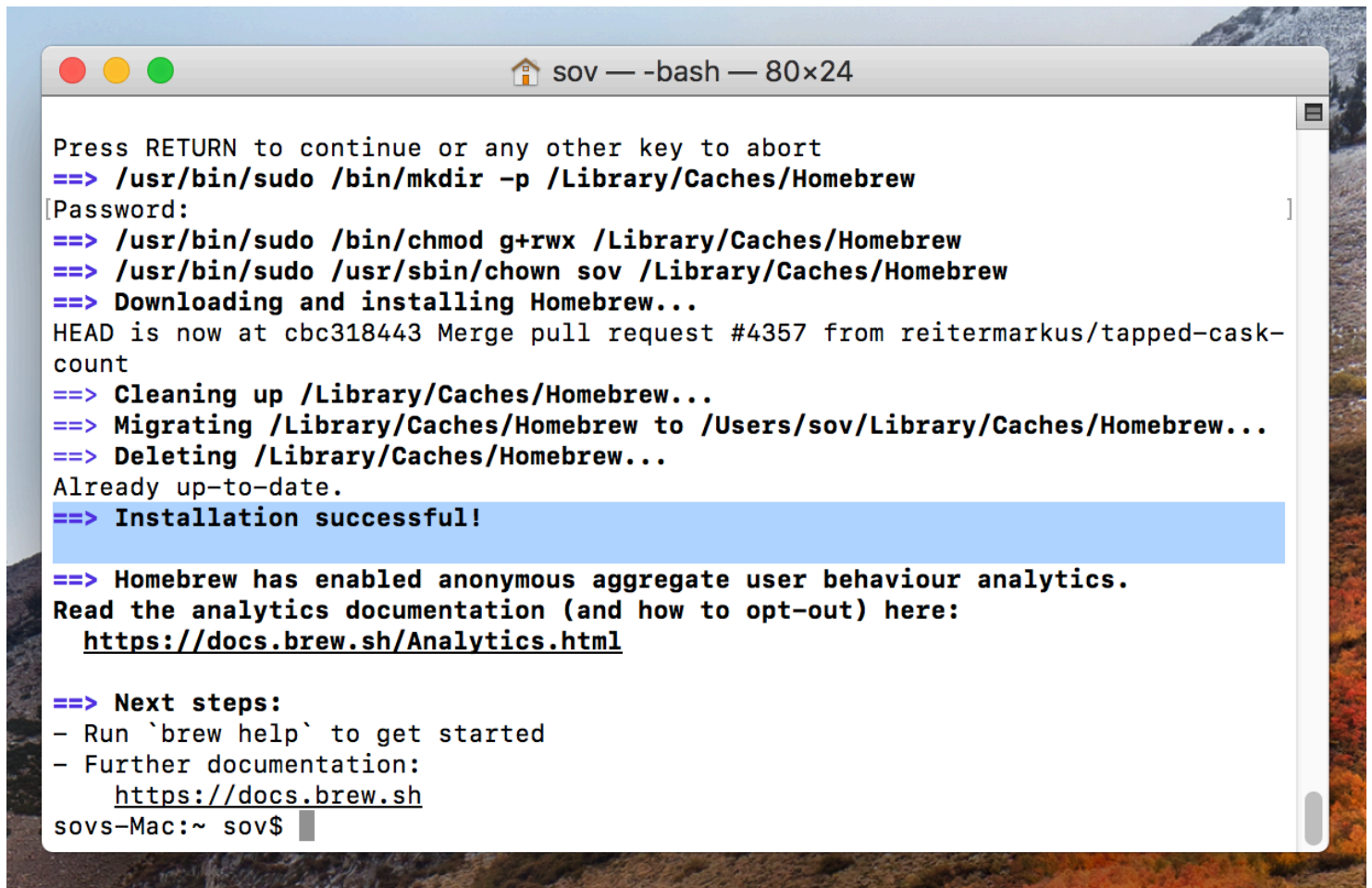
Homebrew is a terminal program that helps you install and manage other programs, similar to Apple's App Store. We will use this program to download several dependencies required for Indy.

Paste this in the terminal and hit **Enter** :

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

When prompted, press **Enter** once more. You will then need to type your user account password into the terminal. This is allow the terminal to make system changes and installations.

Note: You will not see your password as you type it in. This is to keep your password secure.

A screenshot of a macOS terminal window titled 'sov — -bash — 80x24'. The terminal shows the installation of Homebrew. It starts with a prompt to press RETURN. Then, the user runs '/usr/bin/sudo /bin/mkdir -p /Library/Caches/Homebrew'. A password prompt is shown. The user then runs '/usr/bin/sudo /bin/chmod g+rxw /Library/Caches/Homebrew' and '/usr/bin/sudo /usr/sbin/chown sov /Library/Caches/Homebrew'. The terminal shows the progress of downloading and installing Homebrew, including a message about a merge pull request. It then shows the cleanup of the old installation, migration to the new location, and deletion of the old files. The installation is successful, and a message about anonymous aggregate user behaviour analytics is shown. The user is then prompted to read the analytics documentation at 'https://docs.brew.sh/Analytics.html'. Finally, the user is prompted to run 'brew help' to get started and to visit 'https://docs.brew.sh' for further documentation. The terminal ends with the prompt 'sovs-Mac:~ sov\$'.

```
sov — -bash — 80x24

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /Library/Caches/Homebrew
[Password:
==> /usr/bin/sudo /bin/chmod g+rxw /Library/Caches/Homebrew
==> /usr/bin/sudo /usr/sbin/chown sov /Library/Caches/Homebrew
==> Downloading and installing Homebrew...
HEAD is now at cbc318443 Merge pull request #4357 from reitermarkus/tapped-cask-count
==> Cleaning up /Library/Caches/Homebrew...
==> Migrating /Library/Caches/Homebrew to /Users/sov/Library/Caches/Homebrew...
==> Deleting /Library/Caches/Homebrew...
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate user behaviour analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics.html

==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh
sovs-Mac:~ sov$
```

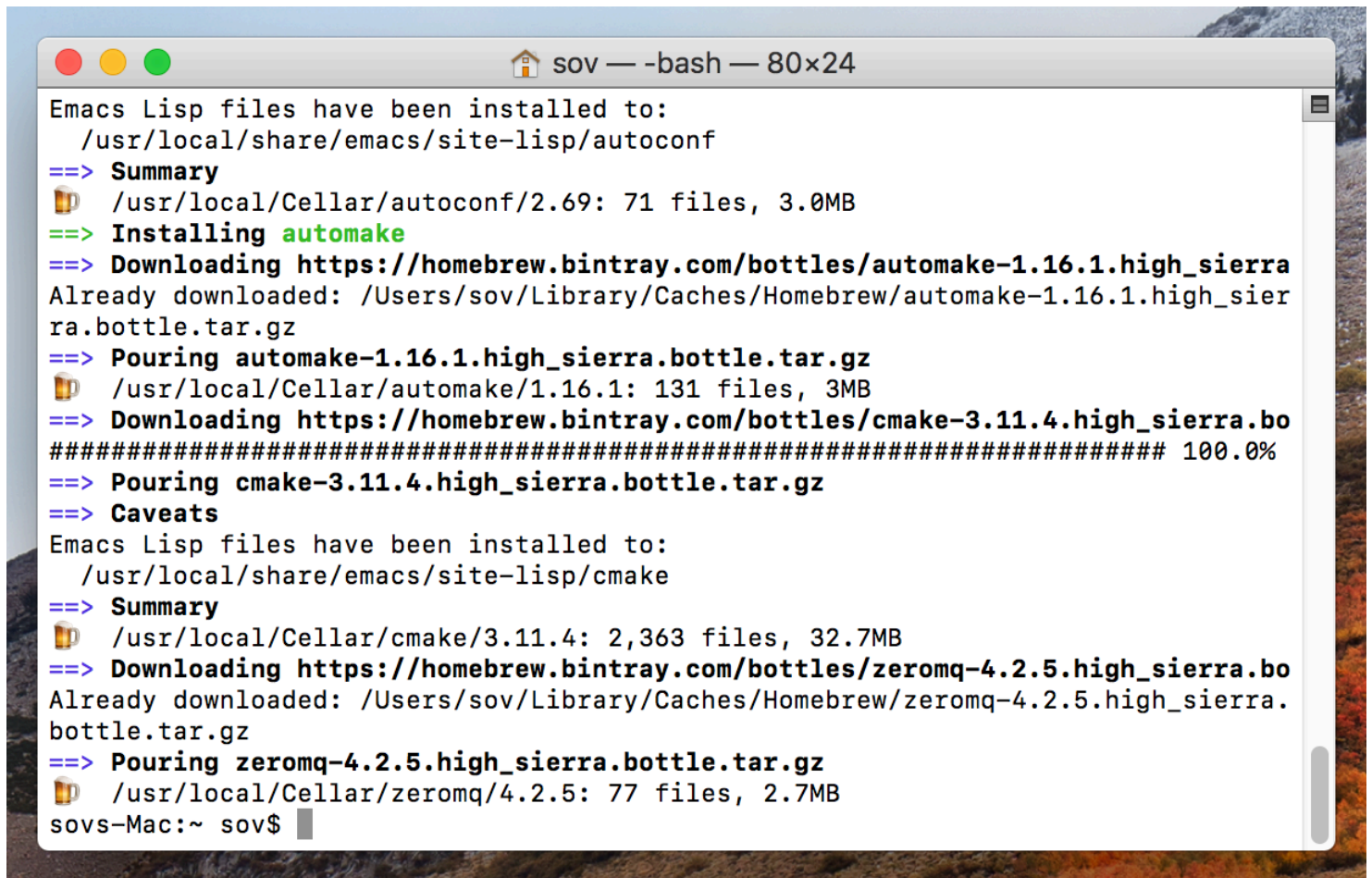
4. Install required dependencies

Several programs and libraries of code are required by `indy`. All dependencies will be installed using Homebrew.

The following command will install 7 dependencies. Paste this in the terminal and hit `Enter`:

```
brew install pkg-config libsodium automake autoconf cmake openssl zeromq
```

Note: If the terminal outputs any sort of error with `libzmq`, it may help to also run the command `brew install zmq`.

A terminal window titled 'sov — -bash — 80x24' showing the output of Homebrew commands. It displays the installation of automake and cmake, including file counts and sizes. The terminal text is as follows:

```
Emacs Lisp files have been installed to:
  /usr/local/share/emacs/site-lisp/autoconf
==> Summary
🍺 /usr/local/Cellar/autoconf/2.69: 71 files, 3.0MB
==> Installing automake
==> Downloading https://homebrew.bintray.com/bottles/automake-1.16.1.high_sierra
Already downloaded: /Users/sov/Library/Caches/Homebrew/automake-1.16.1.high_sier
ra.bottle.tar.gz
==> Pouring automake-1.16.1.high_sierra.bottle.tar.gz
🍺 /usr/local/Cellar/automake/1.16.1: 131 files, 3MB
==> Downloading https://homebrew.bintray.com/bottles/cmake-3.11.4.high_sierra.bo
##### 100.0%
==> Pouring cmake-3.11.4.high_sierra.bottle.tar.gz
==> Caveats
Emacs Lisp files have been installed to:
  /usr/local/share/emacs/site-lisp/cmake
==> Summary
🍺 /usr/local/Cellar/cmake/3.11.4: 2,363 files, 32.7MB
==> Downloading https://homebrew.bintray.com/bottles/zeromq-4.2.5.high_sierra.bo
Already downloaded: /Users/sov/Library/Caches/Homebrew/zeromq-4.2.5.high_sierra.
bottle.tar.gz
==> Pouring zeromq-4.2.5.high_sierra.bottle.tar.gz
🍺 /usr/local/Cellar/zeromq/4.2.5: 77 files, 2.7MB
sovs-Mac:~ sov$
```

5. Set OpenSSL Path

OpenSSL is a code library used by many other programs for enhanced internet security and encryption. We installed `openssl` in the last step, but one more step is needed.

The terminal uses environmental variables, which are pieces of data that are needed by multiple programs, stored in text form. For example, the `USER` environmental variable contains your username.

We need to set an environmental variable called `OPENSSL_DIR` which represents the directory or folder on your computer in which the `openssl` program was installed. To find that folder, type the following command:

```
ls /usr/local/Cellar/openssl
```

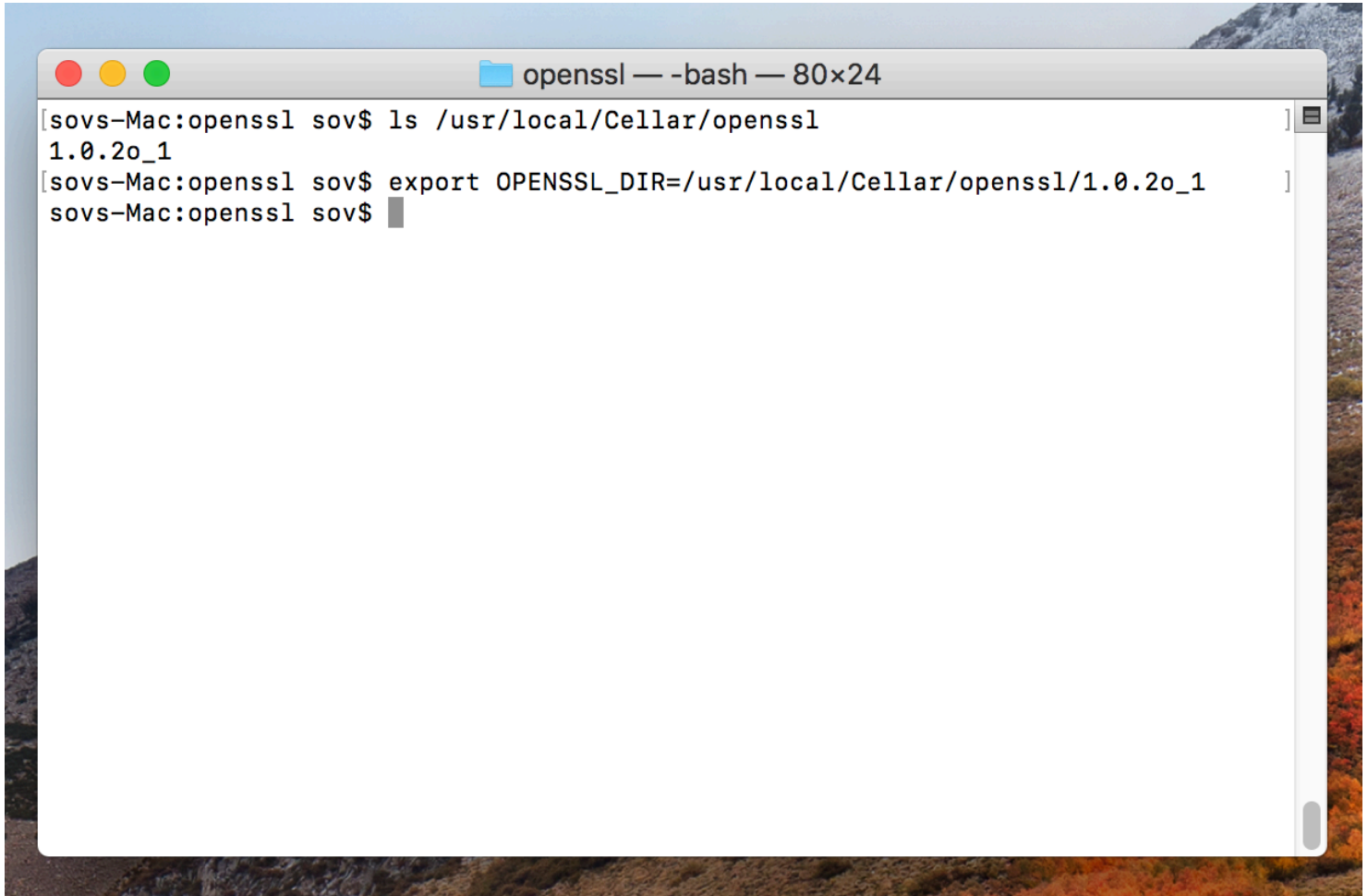
This will list the names of every file and folder within that directory. Since you only just installed `openssl`, there should only be one item, which is the version of `openssl` you installed. **Note that this may vary from the screenshot.**

Copy the command below to the terminal, replace `<version_folder>` with the output of the previous

command, and press `Enter` .

```
export OPENSSL_DIR=/usr/local/Cellar/openssl/<version_folder>
```

Unlike the other commands, this will not return any output. This is expected.

A screenshot of a macOS terminal window titled "openssl — -bash — 80x24". The terminal shows the following commands and output:

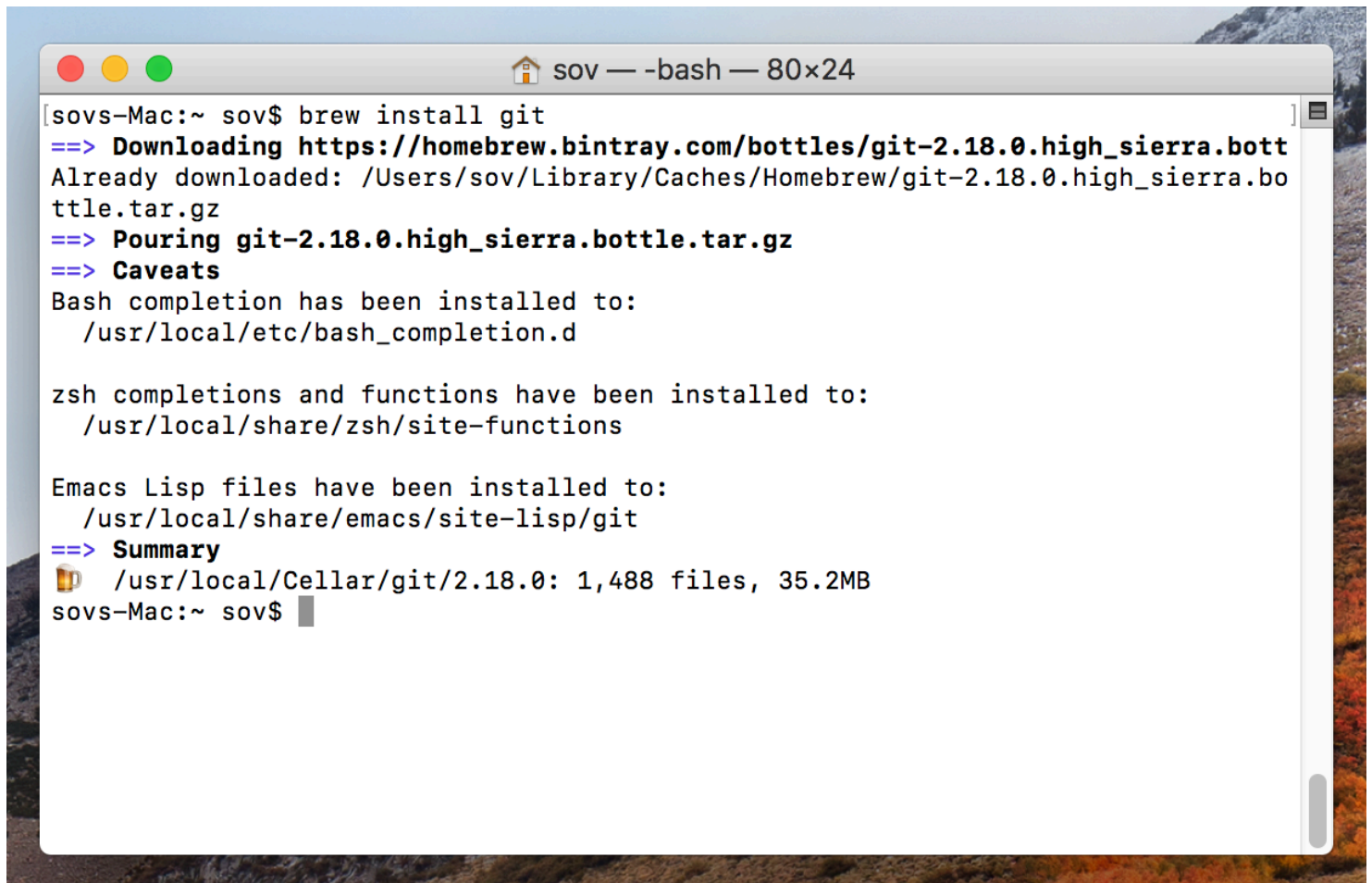
```
[sovs-Mac:openssl sov$ ls /usr/local/Cellar/openssl
1.0.2o_1
[sovs-Mac:openssl sov$ export OPENSSL_DIR=/usr/local/Cellar/openssl/1.0.2o_1
sovs-Mac:openssl sov$
```

6. Install Git

Github is a website that software developers use to host their computer programs. To interact with Github from the terminal, we will use `Homebrew` to install a program called `git` . This program will download `indy-sdk` for us without needing to use an internet browser.

Run this in the terminal:

```
brew install git
```


A screenshot of a macOS terminal window titled 'sov — -bash — 80x24'. The terminal shows the command 'brew install git' being executed. The output indicates that git-2.18.0 was downloaded from Homebrew's bintray and is now installed. It also shows that bash completion, zsh completions, and Emacs Lisp files have been installed to their respective system directories. A summary shows the installation path as '/usr/local/Cellar/git/2.18.0' with 1,488 files and a size of 35.2MB. The prompt returns to 'sov\$' at the end of the session.

```
[sov-Mac:~ sov$ brew install git
==> Downloading https://homebrew.bintray.com/bottles/git-2.18.0.high_sierra.bott
Already downloaded: /Users/sov/Library/Caches/Homebrew/git-2.18.0.high_sierra.bo
ttle.tar.gz
==> Pouring git-2.18.0.high_sierra.bottle.tar.gz
==> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d

zsh completions and functions have been installed to:
  /usr/local/share/zsh/site-functions

Emacs Lisp files have been installed to:
  /usr/local/share/emacs/site-lisp/git
==> Summary
📦 /usr/local/Cellar/git/2.18.0: 1,488 files, 35.2MB
sov-Mac:~ sov$
```

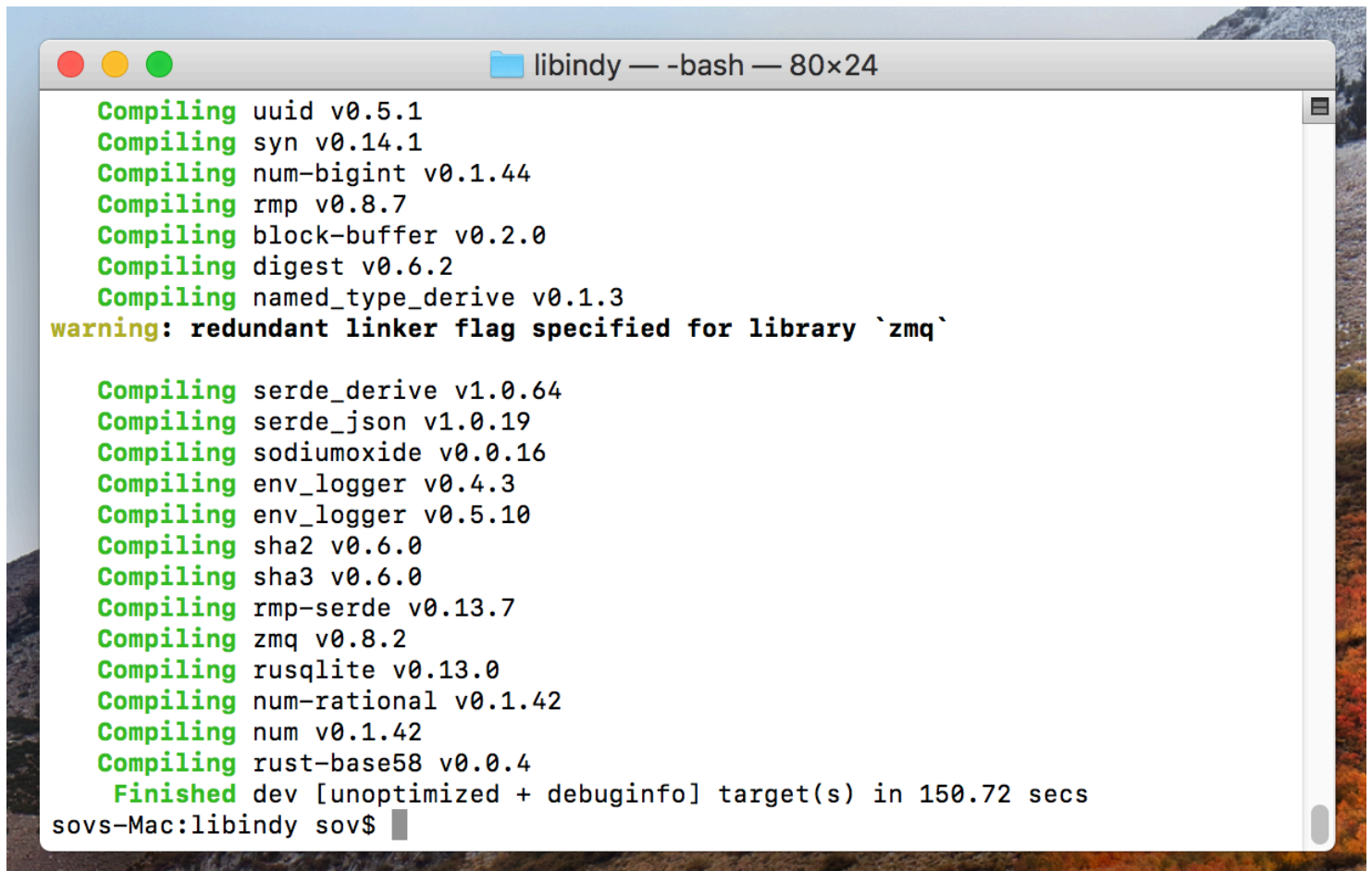
7. Download and Build the `indy-sdk` library

With all prerequisites satisfied, we can now finish the process. We use the program `git` to download the indy programs and code onto our computer (also known as `clone`). Then we change directories using `cd` to go to the folder where the code is located. Last, we use `cargo build` to install `indy` on the computer.

Copy the following three commands into the terminal and hit `Enter`. Note that you may need to hit enter one more time. The process may take a few minutes to download and install.

```
git clone https://github.com/hyperledger/indy-sdk
cd ./indy-sdk/libindy
cargo build
```

The warning in the screenshot may be disregarded.

A screenshot of a macOS terminal window titled "libindy — -bash — 80x24". The window shows the output of a compilation process. It lists various libraries being compiled, such as uuid, syn, num-bigint, rmp, block-buffer, digest, and named_type_derive. A warning message indicates a redundant linker flag for the library 'zmq'. The compilation continues with more libraries like serde_derive, serde_json, sodiumoxide, env_logger, sha2, sha3, rmp-serde, zmq, rusqlite, num-rational, num, and rust-base58. The process concludes with a "Finished" message indicating the target was built in 150.72 seconds. The prompt "sovs-Mac:libindy sov\$" is visible at the bottom.

```
libindy — -bash — 80x24

Compiling uuid v0.5.1
Compiling syn v0.14.1
Compiling num-bigint v0.1.44
Compiling rmp v0.8.7
Compiling block-buffer v0.2.0
Compiling digest v0.6.2
Compiling named_type_derive v0.1.3
warning: redundant linker flag specified for library `zmq`

Compiling serde_derive v1.0.64
Compiling serde_json v1.0.19
Compiling sodiumoxide v0.0.16
Compiling env_logger v0.4.3
Compiling env_logger v0.5.10
Compiling sha2 v0.6.0
Compiling sha3 v0.6.0
Compiling rmp-serde v0.13.7
Compiling zmq v0.8.2
Compiling rusqlite v0.13.0
Compiling num-rational v0.1.42
Compiling num v0.1.42
Compiling rust-base58 v0.0.4
Finished dev [unoptimized + debuginfo] target(s) in 150.72 secs
sovs-Mac:libindy sov$
```

The Indy libraries are now installed!