

## [SelfServe.sovrin.org](https://selfserve.sovrin.org)

# Create a DID and Payment Address using indy-cli

### Introduction

[Selfserve.sovrin.org](https://selfserve.sovrin.org) is a website designed to help you with one part of getting started using the Sovrin Test Networks, BuilderNet and StagingNet. The BuilderNet is for your initial trials with setting up your initial agents, DIDs, Schemas, etc. and is the most volatile of the networks because it gets the newest upgrades first. The StagingNet is more stable and can be safely used for end-user testing and demo's when you are ready to move forward. To be able to enter a DID, a Verkey, and/or a Payment Address into the form provided, you will need to install an agent with a wallet that can store the private counterparts of the public keys for these items that you will provide to the form. In other words, you have to create the items in your own wallet, before the SelfServe tool can add them to the ledger on the requested Sovrin Network. These instructions will give you a step-by-step guide to installing indy-cli, a command line interface that provides access to your very own wallet on your Ubuntu, Mac, or Windows machine. Let's get started!

### I. Install indy-cli

- A. Please see Appendix A (below) and select the right section for instructions on installing the indy-cli on your Ubuntu, Mac, or Windows machine.

### II. Start indy-cli

- . Start your indy-cli using the instructions from Appendix A for your platform.

#### 1. **indy-cli --config cliconfig**

- A. Type 'y' to accept the TAA agreements, if needed.

### III. Create Pools

- . "Creating a pool" is really just adding a named link to an existing network (or pool).

You only need to create the one you need, but here are the instructions for each:

- A. **pool create buildernet gen\_txn\_file=pool\_transactions\_builder\_genesis**
- B. **pool create stagingnet gen\_txn\_file=pool\_transactions\_sandbox\_genesis**
- C. Creating a pool only needs done once for each pool on each machine on which you install indy-cli. Pool connect (the following command) must be done every time you rerun indy-cli.
- D. pool connect <pool-name>

#### 1. **pool connect buildernet**

### IV. Create and open a Wallet

- . The <wallet key> is a secure key that only you should know. Quotes are not required for this key. Save it in a secure place for later use. You will use it every time you need to run commands from the CLI.

- A. **wallet create my\_wallet key=<wallet key>**
- B. **wallet open my\_wallet key=<wallet key>**
- C. Note: Keep your wallet **open** for the remaining commands in this guide.

### V. Create a DID

- . **did new**

A. The return for this command is the DID and Verkey needed in the selfserve web page. To see all of the DID's in your wallet use **did list** . To verify that your DID is on a network use **ledger get-nym did=<your DID>**

B. Please note that you can add a **seed=<32 character secret seed>** to the above command. The seed parameter allows you to recreate your DID in a different wallet if needed. The seed parameter is not required, and might not be needed on the test networks but should be used for DIDs needed for demos or DIDs used on the MainNet.

## VI. Create a Payment Address

. **payment-address create payment\_method=sov**

A. Copy the payment address returned to the appropriate place on the SelfServe web page. To see all of your payment addresses use **payment-address list** . To verify that test-tokens have been added to your payment address use **ledger get-payment-sources payment\_address=<your payment-address>**

B. Like the DID creation command, you can also add a **seed=<32 character secret seed>** to this command. Again, for the test networks it's not a big deal, but for other networks this secret seed will be critical to keep track of.

## VII. Using the Test Tokens

. Now that you have test-tokens shared with you by the SelfServe website you can try sharing those test-tokens with others and using them to pay for transactions by doing one or more of the following:

A. Transfer test-tokens to another payment address

1. Create another payment address by following step VI.A in your indy-cli.

2. **ledger get-fees payment\_method=sov**

3. **ledger payment source\_payment\_address=<payment addr with tokens> target\_payment\_address=<payment address you want to xfer tokens to> amount=<amount to xfer> fee=<amount of token\_xfer fee from get-fees>**

a. For example, the following will transfer 1 test-token from payment addr1 to payment addr2 for a fee of 1 test-token:

b. **ledger payment**

**source\_payment\_address=pay:sov:wZVVnA4mj475vjMxPRuLYtY78dnum6TMyKUx7oEreodEFTxE8**

**target\_payment\_address=pay:sov:6iwcAUxnrU5Ys6cz7s8R7TLKYtYwLekDAYbX9YPf7gV8cf4Bt amount=100000000 fee=100000000**

c. As shown in VI.B the **get-payment-sources** command will show the resulting amount in the payment sources for each payment address. If this is the first time you have tried this, payment addr1 will show 1998 test-tokens (199800000000) and payment addr2 will show 1 test-token.

B. Add a DID to the ledger using test-tokens:

1. Create a new DID using step V.a

2. **ledger nym did=<new DID> verkey=<new verkey> role=source\_payment\_address=<payment addr1> fee=<amount of did fee from get-fees>**

. For example, the following command will add an unprivileged DID to the ledger using 50 test-tokens from payment addr1

a. **ledger nym did=XYzYgWjoH8ej1F94zpkMo verkey=~WiDzXMWXriaxJuJihjafgg role=**

**source\_payment\_address=pay:sov:wZVVnA4mj475vjMxPRuLYtY78dnum6TMyKUx7oEreodEFTxE8 fee=5000000000**

- b. Check for remaining tokens using same method as VII.B.3.c
- C. Add a schema to the ledger using test-tokens
  1. Using the same pattern of adding a `source_payment_address` and a fee, you can perform write commands to the ledger, such as creating a schema as in this example:
  2. **ledger schema name=MyFirstSchema version=1.0  
attr\_names=FirstName,LastName,Address,Birthdate,SSN  
source\_payment\_address=pay:sov:wZVVnA4mj475vjMxPRuLYtY78dnum6TMyKUx7oEreodEFTxE8 fee=25000000000**
- D. If you run out of tokens, just put your payment address into the SelfServe tool again and it will give you more test tokens as long as there are less than 4000 test-tokens at your address.

## APPENDIX A

### Installing the Indy CLI

You will need to perform the following once for each Indy CLI machine you would like to set up. Assistance for each method listed below is available via [support@sovrin.org](mailto:support@sovrin.org).

#### Windows:

To install the CLI on Windows 10 perform the following steps:

1. Download [indy-cli\\_1.14.2.zip](#) and unzip it.
2. Download [libsovtoken\\_1.0.5.zip](#), unzip it, and copy the resulting sovtoken library files to **the right places**.
3. Run `indy-cli.exe` to verify proper installation. You should see a new window appear with an `indy>` prompt. Type "exit" to close the app.
4. If you get an error stating that it is missing `vcruntime140.dll` then do the following:
  - a. Download and install `vc_redist.x64.exe` from the Visual Studio 2017 section on the <https://support.microsoft.com/en-ae/help/2977003/the-latest-supported-visual-c-downloads> page
  - b. Rerun `indy-cli.exe` to see if it works as described in previous step.
5. Download the GENESIS files to the `indy-cli` directory for later use:
  - . Open a command prompt. (This will work differently if you use Windows Terminal)
  - a. `cd` to the directory where you unzipped the `indy-cli` package.
    - i. For example, if you unzipped directly in your 'downloads' directory like I did you would type: **`cd \Users\LBendixsen\Downloads\indy-cli_1.14.2`**
    - b. Download the following two files:
    - c. This one is for the Sovrin BuilderNet (be sure to copy the entire command from the three lines):

- d. `curl -O https://raw.githubusercontent.com/sovrin-foundation/sovrin/master/sovrin/pool\_transactions\_builder\_genesis`
- e. And this one is for the Sovrin StagingNet:
- f. `curl -O https://raw.githubusercontent.com/sovrin-foundation/sovrin/stable/sovrin/pool\_transactions\_sandbox\_genesis`
6. Create a file named `cliconfig` in the same directory with `indy-cli.exe` and add to it the following contents:

```
{  
  "plugins": "<Full Path to sovtoken.dll.lib>/sovtoken.dll.lib:sovtoken_init",  
  "taaAcceptanceMechanism": "for_session"  
}
```
7. Run `indy-cli` using a parameter:
  - a. **`indy-cli --config cliconfig`**
  - b. A full path for `cliconfig` is needed if you do not run `indy-cli` from its directory.

## Ubuntu:

To install the CLI on Ubuntu, perform the following steps from the ubuntu command line:

1. **`sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys CE7709D068DB5E88`**
2. **`sudo add-apt-repository "deb https://repo.sovrin.org/sdk/deb xenial stable"`**
3. **`sudo add-apt-repository "deb https://repo.sovrin.org/deb xenial stable"`**
4. **`sudo apt-get update -y`**
5. **`sudo apt-get upgrade -y`**
6. **`sudo apt-get install -y indy-cli libsovtoken`**
7. **`cd ~`**
8. Download the Genesis files for later use:
9. This one is for the Sovrin BuilderNet (be sure to copy the entire command from the three lines)
10. `curl -O https://raw.githubusercontent.com/sovrin-foundation/sovrin/master/sovrin/pool\_transactions\_builder\_genesis`
11. And this one is for the Sovrin StagingNet
12. `curl -O https://raw.githubusercontent.com/sovrin-foundation/sovrin/stable/sovrin/pool\_transactions\_sandbox\_genesis`
13. Create a `cliconfig` file in your home directory and add the following contents:

```
{  
  "plugins": "/usr/lib/libsovtoken.so:sovtoken_init",  
  "taaAcceptanceMechanism": "for_session"  
}
```
14. Run `indy-cli`:
  - a. **`indy-cli --config cliconfig`**
  - b. A full path for `cliconfig` is needed if you do not run `indy-cli` from its directory

## Mac:

Since there is not a prepackaged version of the CLI prepared for the Mac, the following steps will help you to create an environment, build, and run the CLI in a Mac terminal.

1. Open a Terminal
2. Run the following commands in the terminal:
  - a. `cd ~`
  - b. `mkdir github`
  - c. `cd github`
  - d. `git clone https://github.com/hyperledger/indy-sdk.git`
  - e. `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
  - f. `curl https://sh.rustup.rs -sSf | sh`
- i. Follow onscreen instructions to install rust
  - g. `brew install pkg-config`  
`https://raw.githubusercontent.com/Homebrew/homebrew-core/65effd2b617bade68a8a2c5b39e1c3089cc0e945/Formula/libodium.rb automake autoconf cmake openssl zeromq zmq`
  3. NOTE: the openssl path needs to match what you currently have on your system  
Run `> ls /usr/local/Cellar/openssl/`
    - a. Note the name of the directory shown (mine is 1.0.2p but latest is 1.0.2q)
    - b. Use this directory in place of the one listed below in your .profile file
  4. Add the following lines to your ~/.profile file (making the correction shown in the previous step if needed)

```
export PATH="$HOME/.cargo/bin:$PATH:~/github/indy-  
sdk/libindy/target/debug:~/github/indy-sdk/cli/target/debug"  
export PKG_CONFIG_ALLOW_CROSS=1  
export CARGO_INCREMENTAL=1  
export RUST_LOG=indy=trace  
export RUST_TEST_THREADS=1  
export OPENSSL_DIR=/usr/local/Cellar/openssl/1.0.2p #use your path  
export LIBRARY_PATH=~/github/indy-sdk/libindy/target/debug/  
export LIBINDY_DIR=~/github/indy-sdk/libindy/target/debug/
```
5. Run the following commands from your terminal to build the CLI and other dependencies:
  - a. `source ~/.profile`
  - b. `cd ~/github/indy-sdk/libindy`
  - c. `cargo build`
  - d. `cd ../cli`
  - e. `cargo build`
  - f. `cd ../libsovtoken/libsovtoken`
  - g. `cargo build`
6. You can now run indy-cli to make sure it starts from within a terminal by typing:  
.  
`indy-cli`
  - a. `exit` (To exit from the indy-cli prompt when you are done)
7. Download the Genesis files using the following commands:  
.  
`cd ~`

- a. This one is for the Sovrin BuilderNet (be sure to copy the entire command from the three lines):
- b. `curl -O https://raw.githubusercontent.com/sovrin-foundation/sovrin/master/sovrin/pool\_transactions\_builder\_genesis`
- c. And this one is for the Sovrin StagingNet:
- d. `curl -O https://raw.githubusercontent.com/sovrin-foundation/sovrin/stable/sovrin/pool\_transactions\_sandbox\_genesis`
8. Create a cliconfig file in your home directory and add the following contents:

```
{  
  "plugins": "$HOME/github/libsovtoken/libsovtoken/target/debug/libsovtoken.dylib:sovtoken_init",  
  "taaAcceptanceMechanism": "for_session"  
}
```
9. Run indy-cli:
  - a. **indy-cli --config cliconfig**
  - b. A full path for cliconfig is needed if you do not run indy-cli from its directory