
HyperledgerIndy Documentation

Release 1.0

Michael McKean

Aug 07, 2019

1	Key Characteristics	3
2	Our Repositories	5
2.1	Documentation README	5



HYPERLEDGER INDY

Hyperledger Indy provides tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any other silo.

If you are a developer, begin your journey in our [getting-started guide](#).

Key Characteristics

- Distributed ledger purpose-built for decentralized identity
- Correlation-resistant by design
- DIDs (Decentralized Identifiers) that are globally unique and resolvable (via a ledger) without requiring any centralized resolution authority
- Pairwise Identifiers create secure, 1:1 relationships between any two entities
- Verifiable Claims are interoperable format for exchange of digital identity attributes and relationships currently in the standardization pipeline at the W3C
- Zero Knowledge Proofs which prove that some or all of the data in a set of Claims is true without revealing any additional information, including the identity of the Prover

- Distributed Ledger:
 - `indy-node`
 - `indy-plenum`
- Client Tools:
 - `indy-sdk`
 - `indy-agent`
- Shared Components:
 - `indy-hipe`
 - `indy-crypto`

2.1 Documentation README

2.1.1 How to Add Documentation

For new features and pull requests, maintainers should make sure that the **contributor has added an explanation for their changes in the docs folder before merging the PR.**

Contributors should write an addition to a current file or add a new file to the `docs/source/` folder that explains what their feature is and how it works. If needed, they may also add a link to more technical README's located nearer to the code.

Whenever additions are made to the docs, make sure to update the `index.rst` in whichever folder the file has been added, and build the docs locally to confirm they work (TODO: add the `sphinx-build` command to our CI/CD flow).

For example, if I wanted to add another file to the `indy-sdk docs/` folder named `glossary.md`, I would create the file, and then add a reference to it in the `index.rst`:

```
.. toctree::
  :maxdepth: 1
  :hidden:

  getting-started/index.rst
  ...
  other files
  ...
  glossary.md .. <-- this is your new file!
```

To add a new file to a subfolder, simply update the subfolder's `index.rst` with the relative link to your file.

If you'd like to link to a file outside of the `docs/` folder, you'll need to provide an external github link (this is by design, to keep our docs organized into a single folder)

2.1.2 Building the docs on your machine

Here are the quick steps to achieve this on a local machine without depending on ReadTheDocs. Note: Instructions may differ depending on your OS. Run these commands within the repository folder

```
pip install Sphinx
pip install sphinx_rtd_theme
pip install recommonmark==0.4.0
cd docs/source # Be in this directory. Makefile sits there.
make html
```

This will generate all the html files in `docs/source/_build/html` which you can then browse locally in your browser. Every time you make a change to the documentation you will need to rerun `make html`.

2.1.3 Additional Instructions

This section is to be used for repo maintainers to add additional documentation guidelines or instructions.