ProvenDB Builder’s Fest

Getting Started

You can view a full digital version of this guide with further detail at:

https://rebrand.ly/1ck5dj

1. Go to ProvenDB.com and sign up for a free account, once you’re in the dashboard, create your new ProvenDB service (this may take some time).

2. In the mean time download our ProvenDB Shell Helpers, these integrate with the native MongoDB Shell. You can download these from https://rebrand.ly/g5jtok

4. Now connect to one of our premade services, the URI will be in the following format:

```
# On Linux and Mac
export BID=NN # Where NN is your builder ID, for example 01, 02 03 etc...
export PDB_DB=tst_f$BID
export PDB_URI=mongodb://fest:fest@provendb_tst_f$BID.provendb.io/$PDB_DB?ssl=true
```

```
# Windows
set BID=NN # Where NN is your builder ID, for example 01, 02 03 etc...
set PDB_DB=tst_f%BID%
set PDB_URI=mongodb://fest:fest@provendb_tst_f%BID%.provendb.io/%PDB_DB%?ssl=true
```

Working with Versions

First lets check what version we are working on.

```
db.getVersion()
```

OR

```
db.runCommand({getVersion: 1})
```

Let’s have a look at version **5**

```
db.setVersion(4)
```

OR

```
db.runCommand({setVersion: 4})
```

Let’s return to the latest version.

```
db.setVersion()
```

OR

```
db.runCommand({setVersion:
‘current’})
```

Point in time history

By leveraging our versioning system, we can see the whole history of a document:

```
db.docHistory(‘accounts’,{name:
‘Guy’} )
```

OR

```
db.runCommand({docHistory:
{collection: ‘accounts’, filter:
{name: ‘Guy’}}})
```

Using MongoFiles

Let’s add some data we want to prove. In our case we’ll use MongoFile to add our source code, but you can simply run **db.test.insertOne({ ... })** instead if you like.

```
# On Mac or Linux
mongofiles --uri $PDB_URI --db $PDB_DB put myFiles.tgz

# On Windows
mongofiles --uri %PDB_URI% --db %PDB_DB% put myFiles.tgz
```

Working with Proofs

Let’s look at the file we want to prove.

```
db.fs.files.find({}, fileName: 1)
```

Great, what version does this file exist in?

```
db.getVersion()
```

OR

```
db.runCommand({getVersion: 1})
```

Now create a new proof for this version.

```
db.submitProof(12)
```

OR

```
db.runCommand({submitProof: 12})
```

Let’s look at what a previously completed proof looks like.

```
db.getProof(4)
```

OR

```
db.runCommand({getProof:
‘0bfc8b10-8cda-11e9-a57b-01ae91ab539e’
})
```

We can even get the proof information for a single document.

```
db.getDocumentProof(‘accounts’,
{name: ‘Guy’}, 4)
```

OR

```
db.runCommand({getDocumentProof:
{collection: ‘accounts’, filter:
{name: ‘Guy’}, version: 4}})
```

And finally we will verify that our data hasn’t been tampered with

```
db.verifyProof(4)
```

OR

```
db.runCommand({verifyProof: 4})
```

Next Steps!

Congratulations, you’ve just become a Blockchain Developer**!** And the best thing is that if you already knew **MongoDB**, you know **ProvenDB**.
But if you need a little help getting started, here are a few places to begin:

- View our documentation at **ProvenDB.Readme.io.**
- Read some of our blogs and guides at **Medium.com/ProvenDB.**
- See our open source sample application at **ProvenDocs.com.**