

```

File Edit View Search Terminal Help
edureka@localhost ~]$ hdfs fsck /
16/10/28 15:56:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
lthin-java classes where applicable
Connecting to namenode via http://localhost:50070
FSCK started by edureka (auth:SIMPLE) from /127.0.0.1 for path / at Fri Oct 28 15:56:10 IST 2016
.....Status: HEALTHY
Total size:      549646 B
Total dirs:      52
Total files:      38
Total symlinks:      0
Total blocks (validated):      38 (avg. block size 14464 B)
Minimally replicated blocks:      38 (100.0 %)
Over-replicated blocks:      0 (0.0 %)
Under-replicated blocks:      0 (0.0 %)
Mis-replicated blocks:      0 (0.0 %)
Default replication factor:      1
Average block replication:      1.0
Corrupt blocks:      0
Missing replicas:      0 (0.0 %)
Number of data nodes:      1
Number of racks:      1
FSCK ended at Fri Oct 28 15:56:10 IST 2016 in 17 milliseconds

```

HDFS client

This library allow to connect to the Hadoop datalab cluster without any system installation (except Java).

* A maven dependency can be import in your Java application for use hdfs

* A command line interface can be used on your machine `_hdfs dfs_`

HDFS in Java application

```
``xml
```

```
<dependency>
```

```
    <groupId>com.tony.hdfs</groupId>
```

```
    <artifactId>HdfsClient</artifactId>
```

```
    <version>1.0</version>
```

```
</dependency>
```

```
...
```

Configuration

Define your hadoop.properties in your project

```
...
```

```
hadoop.cluster=clustername
```

```
hadoop.failoverProxy=org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyPr
vider
```

```
hadoop.namenodes=nn1,nn2
```

```
hadoop.rpcAddress=[DNS_NAMENODE1]:[PORT_RPC],[DNS_NAMENODE2]:[PORT_RPC]
```

```
hadoop.httpAddress=[DNS_NAMENODE1]:[PORT_HTTP],[DNS_NAMENODE2]:[PORT_HTTP]
```

```
hadoop.krb5Url=hadoop/krb5.conf
```

```
hadoop.jaasConfUrl=hadoop/jaas.conf
...
```

Note: Example `__krb5.conf__` and `__jaas.conf__` are embedded in jar and must be overridden

Usage

```
``java
Properties prop = new Properties();

ClassLoader classLoader = getClass().getClassLoader();

InputStream input = new FileInputStream("./hadoop.properties");
prop.load(input);

client = new HadoopClient();

client.setHadoopCluster(prop.getProperty("hadoop.cluster"));
client.setNamenodes(prop.getProperty("hadoop.namenodes"));
client.setHttpAaddress(prop.getProperty("hadoop.httpAddress"));
client.setRpcAddress(prop.getProperty("hadoop.rpcAddress"));
client.setHadoopProxy(prop.getProperty("hadoop.failoverProxy"));

# For use internal krb5 and jaas files
URL jaas = classLoader.getResource(prop.getProperty("hadoop.jaasConfUrl"));
URL krb5 = classLoader.getResource(prop.getProperty("hadoop.krb5Url"));

# For use external krb5 and jaas files
#URL jaas = new File(prop.getProperty("hadoop.jaasConfUrl")).toURL();
#URL krb5 = new File(prop.getProperty("hadoop.krb5Url")).toURL();

client.setJaasConfUrl(jaas);
client.setKrbConfUrl(krb5);

String keytabPath = new File("xxx.keytab").getPath();

FileSystem fs = client.hadoopConnectionWithKeytab(keytabPath, "xxx@xxx.CORP");

// or with user/password
FileSystem fs = client.hadoopConnectionWithUserPassword("xxx@xxx.CORP", "xxx");
...
```