Block Chain = = Blockchain ??

First of all, let’s clear one thing Blockchain is not a Bitcoin and Bitcoin is not a Blockchain.

Blockchain is a concept which has been used in cryptocurrency or digital currency.

The first work on a cryptographically secured chain of blocks was described in 1991 by Stuart Haber and W. Scott Stornetta and became famous in 2008 when an unknown person  [Satoshi Nakamoto](https://en.wikipedia.org/wiki/Satoshi_Nakamoto) used it in one of most famous cryptocurrency today ie Bitcoin.

Satoshi used Blochchain as the public [ledger](https://en.wikipedia.org/wiki/Ledger) for all transactions of Bitcoin on the network. The invention of the blockchain for bitcoin made it the first digital currency to solve the [double-spending](https://en.wikipedia.org/wiki/Double-spending) problem without the need of a trusted authority or central [server](https://en.wikipedia.org/wiki/Server_(computing)). The bitcoin design has been the inspiration for other applications

each of the [Big Four accounting firms](https://en.wikipedia.org/wiki/Big_Four_accounting_firms) is testing blockchain technologies in various formats. [Ernst & Young](https://en.wikipedia.org/wiki/Ernst_%26_Young) has provided [cryptocurrency wallets](https://en.wikipedia.org/wiki/Cryptocurrency_wallets) to all (Swiss) employees,[[80]](https://en.wikipedia.org/wiki/Blockchain#cite_note-80) has installed a bitcoin ATM in their office in Switzerland, and accepts bitcoin as payment for all its consulting services.

A **blockchain**,[[1]](https://en.wikipedia.org/wiki/Blockchain#cite_note-te20151031-1)[[2]](https://en.wikipedia.org/wiki/Blockchain#cite_note-fortune20160515-2)[[3]](https://en.wikipedia.org/wiki/Blockchain#cite_note-nyt20160521-3) originally **block chain**,[[4]](https://en.wikipedia.org/wiki/Blockchain#cite_note-primer-4)[[5]](https://en.wikipedia.org/wiki/Blockchain#cite_note-obmh-5) is a continuously growing list of [records](https://en.wikipedia.org/wiki/Record_(computer_science)), called *blocks*, which are linked and secured using [cryptography](https://en.wikipedia.org/wiki/Cryptography). The words *block* and *chain* were used separately in Satoshi Nakamoto's original paper, but were eventually popularized as a single word, *blockchain,* by 2016.

**Blockchain is a Linked List**

If we compare blockchain with Linkedlist, yes it’s a linked list but with the assurance of secured data.it acts like a Linkedlist with where each node has three major component as:

1. **Current Hash** is a String element which is calculated (Hashed) from **Data** and **Previous Hash**.
2. **Previous Hash** holds the **Current Hash** of the previous block (node)
3. **Data**, can be any object, which represents the node

**Header**

**Previous Node Address**

**Next Node Address**

**Data**

**Previous Hash**

**+**

**Data**

**=**

**Current Hash**

**Current Hash**

**Previous Hash**

**Data**

**Current Hash**

**Current Hash**

**Data**

**Previous Hash**

**Previous Hash**

**Data**

As we can see in above diagram its very much similar with Linkedlist but there is no linking between nodes using node address instead it has previous hash which is the current hash of previous node and current hash which is the combination of previous hash and data.

Unless linked list it has fully secured data because if we try to alter the data the current hash would also get changed because onece we create the hash of certain data we get the same hash again and again if our data is same.

Hence no one can cheat by changing the data in any of the node because changing data will cause different hash and different hash will cause invailid block hence it ensured the integrity of the BlockChain and reduces chances of tampering with old blocks.

Blockchain provides four important features:  
  
**1. Decentralization (No individual administrator)**  
**2. Integrity of data (No tampering)**  
**3. Smart Contracts**

Blockchain can be used in a public peer to peer network where participants are not aware about each other or in a private business where participants know each other and trust each other.

**Creating Blockchain**

**Class Block**

**public** **class** Block {

**public** **int** index;

**public** String timestamp;

**public** Data data;

**public** String hash;

**public** String previousHash;

}

Block class is a class which contains index,timestamp,data,hash and previousHash.

**Index:** is the index of a block in a linkedlist

**Timestamp:** timestamp is used to keep the track of block was created.

**Data:** data which has to be stored in block.

**Hash:** unique hash code generated by data + previoushash .

**Previoushash:** hash of previous block.

**Constructor of Block**

**public** Block(Data data) {

**this**.timeStamp = ""+**new** Date().getTime();

**this**.data = data;

}

**Class Data**

**package** com.piyush.app.blockchain;

**public** **class** Data {

String name;

**int** balance;

**public** Data(String name,**int** balance){

**this**.name=name;

**this**.balance=balance;

}

@Override

**public**

String toString() {

**return** **this**.name+" "+**this**.balance;

}

}

Data class is the class holds our data like **name** and the **balance** of a customer. This is the data which should be secured from tempering and blockchain gives us security that there would be no data tempering.

**Class BlockChain**

**public** **class** BlockChain {

**public** Block generateBlock(Block block,List list) **throws** Exception{

**try** {

block.previousHash=Utils.*getPreviousHash*(list);

}

**catch** (Exception e) {

**throw** **new** Exception("previous hash null or empty");

}

String hashCode=Utils.*generateHash*(block);

block.setHash(hashCode);

**return** block;

}

}

**Class Utils**

**public** **class** Utils {

**public** **static** String generateHash(Block block) {

String sha256hex = org.apache.commons.codec.digest.DigestUtils

.*sha256Hex*(block.previousHash + "" + block.index + "" + block.timeStamp + "" + block.data);

**return** sha256hex;

}

**public** **static** <E> String getPreviousHash(List<E> list) **throws** Exception {

**if** (list.size() != 0) {

Block block = (Block) list.get(list.size() - 1);

String previousHash = block.getHash();

**if** (previousHash != **null** && !previousHash.equals("")) {

**return** previousHash;

}

**else** {

**throw** **new** Exception("previous hash null or empty");

}

}

**else** {

**return** "firsthash";

}

}

**public** **static** Map isBlockChainValid(List<Block> blockChainList) **throws** Exception {

**if** (!blockChainList.isEmpty()) {

**if** (blockChainList.size() > 1) {

**return** *validate*(blockChainList);

}

**else** {

**throw** **new** Exception("block chain is empty");

}

}

**else** {

**throw** **new** Exception("block chain is empty");

}

}

**private** **static** Map validate(List<Block> blockChainList) {

Block current;

Block previous;

Map<String, Object> result = **new** HashMap();

**for** (**int** i = 1; i < blockChainList.size(); i++) {

current = blockChainList.get(i);

previous = blockChainList.get(i - 1);

**if** (!previous.getHash().equals(current.getPreviousHash())) {

result.put("block", blockChainList.get(i));

result.put("index", i);

}

}

**return** result;

}

}

**generateHash:** this is the main and most import part of the blockchain where we generate the hash. We have lots of implementation available to generate hash but I have used sha256Hex.

Hash is the combination of **data+previoushash**.

We have to add one below dependency to get this implementation though there are lots of api and other methods available on internet we can use any one of them.

**Gradle:**

compile **group**: 'commons-codec', **name**: 'commons-codec', **version**: '1.11'

**Maven:**

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.11</version>

</dependency>

**getPreviousHash:**

previousHash also plays very important role in Blockchain because without previous hash we can not create current hash and we can not link our blocks together.

**isBlockChainValid:**

whenever we add any block in our blockchain here we check if the Blochchain is valid or not by checking the previous and current hash of each block.

**validate:**

we validate the each block by generating currenthash of block by previousblock and data and match it with already present currenthash if both are same then data id not altered else data hash been altered and block chain is invalid.

**When Data is altered**

Block Chain is altered at index2

hash 3cd627b352c68a2d4a3664806355a6f3fcf3d378b336380b907fcb71e41edf5f

previous hash firsthash

data piyush1 28

hash 1dd182aa0d8466541a97bd780571728b3169ec7a8df97c9bb526a35d2fffa8a0

previous hash 80430d2efe27badd4e73dd52e65893f5d8aaba87fb1f186643c12c5d7f830074

data piyush2 29

hash 043dfeb5ea9cd0337a1d386b5b58bcf850248f84644a0b73bbfc1f762e5fe1ef

previous hash 3cd627b352c68a2d4a3664806355a6f3fcf3d378b336380b907fcb71e41edf5f

data piyush2 28

hash 80430d2efe27badd4e73dd52e65893f5d8aaba87fb1f186643c12c5d7f830074

previous hash 043dfeb5ea9cd0337a1d386b5b58bcf850248f84644a0b73bbfc1f762e5fe1ef

data piyush3 28

**No data altered**

hash 33bd892bad70ff4fa7ab0f3ec648df8a40608f872337e8a8cfc8b5c8e87e1c49

previous hash firsthash

data piyush1 28

hash 41169e01eb6160f0947640340540c246e5654c2db02223a6a44d24c016337d3a

previous hash 33bd892bad70ff4fa7ab0f3ec648df8a40608f872337e8a8cfc8b5c8e87e1c49

data piyush2 28

hash 48441e40a3cd6384a2ce09edf3f3acefa4f18d852608fd65bc51ead861411c96

previous hash 41169e01eb6160f0947640340540c246e5654c2db02223a6a44d24c016337d3a

data piyush3 28

Please go through **Dataauthorization** blog to know more about how to prevent you data to get tempered.