

AuthSkipList

```

- head: Node<String, ByteArray>
- level: Int
- size: Int
- deltas: MutableList<Triple<String, String, Any?>>
- stack: Stack<Pair<Node<String, ByteArray>, String>>

- findNext(e: String, current: Node<String, ByteArray>): Node<String, ByteArray>
- find(e: String, current: Node<String, ByteArray>): Node<String, ByteArray>
- find(e: String): Node<String, ByteArray>
- containsKey(k: String): Boolean
- applyAdd(k: String, value: Any?): ByteArray
- applyDel(k: String): ByteArray
- clone(): ADS<ByteArray, String, Any>
- lessThan(x: String, y: String): Boolean
- storeHashes(): HashMap<String, Pair<ByteArray?, ByteArray?>>
- height(node: Node<String, ByteArray>): Int
- gap(node: Node<String, ByteArray>, next: Node<String, ByteArray>): Int

+ getValue(k: String): Any?
+ getWithProof(k: String): Pair<Any?, Proof<ByteArray, String, Any>>
+ getProof(keys: List<String>): Proof<ByteArray, String, Any>
+ applyDeltas(): ADS<ByteArray, String, Any>
+ add(k: String, v: Any): ByteArray
+ del(k: String): ByteArray
+ rootHash(): ByteArray
+ printADS()
+ isEmpty(): Boolean

```

<<Interface>>

ADS<K: Comparable<K>, H, V>

```

+ getWithProof(k: K): Pair<V?, Proof<H, K, V>>
+ getProof(keys: List<K>): Proof<H, K, V>
+ getValue(k: K): V?
+ add(k: K, v: V): H
+ del(k: K): H
+ applyDeltas(): ADS<H, K, V>
+ rootHash(): H
+ isEmpty(): Boolean
+ printADS()

```

Node<K: Comparable<K>, H>

```

- firstHash: H?
- secondHash: H?
- key: K
- next, prev, up: Node<K, H>?
- down: Any?

+ isPlateau(): Boolean

```

create

call

proof

<<Interface>>

Proof<H, K: Comparable<K>, V>

```

+ keys: MutableSet<K>

+ rootHash(): H
+ union(proof: Proof<H, K, V>): Proof<H, K, V>
+ rootHash(list: List<Pair<K, V?>>): Pair<Proof<H, K, V>, H>
+ printProof()

```

SkipListProof

```

- head: Node<String, ByteArray>
- baseNodes: HashMap<String, Node<String, ByteArray>>
+ keys: MutableSet<String>

- lessThan(x: String, y: String): Boolean
- clone(): Proof<ByteArray, String, Any>

+ rootHash(): ByteArray
+ union(proof: Proof<ByteArray, String, Any>): Proof<ByteArray, String, Any>
+ rootHash(list: List<Pair<String, Any?>>): Pair<Proof<ByteArray, String, Any>, ByteArray>
+ printProof()

```