Storage

# First comments (Date unknown)

I have been able to connect to a peer from the node ID in the tutorial. I have no idea if this is a full working node or just some random key.

I Can't store the value with the function. Maybe it's because the key/Value are wrong. Maybe the key needs to be a hash of the value or something.

# 28/01/2019

The value is being stored on my own machine under the path specified in the Node object. The iterativeStore function still returns a zero meaning it didn’t STORE any of the values in any peers. The storing should be doable with two peers.

I also tried hashing the value with SHA 256 but this did nothing.

I tried retrieving the value with the iterativeFindValue function but it returns nothing. This makes sense if it’s on my storage as I don’t think my own Node ID exists in the routing table.

I need to set this up with two nodes that I know are able to store items. The only way to do this is to run two servers in my own network or run another on someone else’s machine.

# 30/01/2019

I have successfully implemented two different nodes on the network and they are able to store files. The iterativeFindValue returns the node ID along with the details of K nodes meaning that the file is stored on all them or that it can’t find the value in them (probably the former). I need to figure out a way of getting the ID of the nodes in a string so I can implement a way of transferring IOTA to their addresses.

# 03/02/2019

I have been able to parse the response and get the Node’s address. I tried to use the send function but it doesn’t work properly. This is probably because I used the address instead of the node object. I should also write a proper handler and place the function in a module.

# 08/02/2019

The program will now return the correct value. When you first store the values they appear in the database as a regular text file. The next time you publish something this file becomes an encrypted file you cannot read. When the files are encrypted you can then access the values and publishers with a key.

The next thing I need to do is figure out a way to get the nodes to send each other their Iota addresses so they can be paid for storing the files.

# 10/02/2019

So the value that is returned when the file isn’t encrypted yet is actually an array of contacts. A contact itself is an array that contains a node’s ID and the details required to send a message to it.   
The send function that I was trying to use to send a message requesting the IOTA address required the node’s contact not just its ID.   
So if I get the node IDs of all those that stored the data I can request their IOTA address and send them Iota. I will return to using the key directly after it’s stored even though I can’t get the value back. This should work perfectly.

# 10/02/2019 Part 2

The file sharing is completely working and Iotas can be sent to any node that stores a file. The function used to send iota isn’t very efficient. In the future it would be better to write more efficient code or to even create a module that can be accessed.