# Introduction to Node.js Exercises

## Create simple storage

### Create Node.js project

Prepare the "**package.json**" file and the IDE configurations.

### Create a module named "storage"

Create a module which is named "**storage**". The purpose of the module is to **store** key-value pairs where the key is always a string. Export from the module the following functions: "**put**", "**get**", "**update**", "**delete**", "**clear**", "**save**" and "**load**".

### Implement the "put" function

The "**put**" function should have two parameters – one for the key and one for the value. If the key is not a string, you should throw an error. If the key already **exists** in the storage, you should throw an error. Otherwise you should save the key-value pair in memory.

function put (key,value){

{{logic}}

return

}

### Implement the "get" function

The "**get**" function should have one parameter – for the key. If the key is not a string, you should throw an error. If the key does not exist in the storage, you should throw an error. Otherwise the function should **return** the **value** corresponding to the provided key.

function get (key){

{{logic}}

return {{return searched value}}

}

### Implement the "getAll" function

The "**getAll**" function should return all key-value pairs from the storage, if the storage is empty it should print appropriate message.

function getAll (){

{{logic}}

return {{return all values}}

}

### Implement the "update" function

The "**update**" function should have two parameters – one for the key and one for the value. If the key is not a string, you should throw an **error**. If the key does not exist in the storage, you should throw an **error**. Otherwise you should **update** the **key-value pair** in memory.

function update (key,value){

{{logic}}

return

}

### Implement the "delete" function

The "**delete**" function should have one parameter – for the key. If the key is not a string, you should throw an error. If the key does not exist in the storage, you should throw an error. Otherwise you should **delete** the **key-value pair** from the memory storage.

function deleteItem (key){

{{logic}}

return

}

### Implement the "clear" function

The "**clear**" function should **delete** **all** saved key-value pairs **in** the **storage**.

function clear (){

{{logic}}

return

}

### Implement the "save" function

The "**save**" function should **save** all key-value pairs **on a file** named "**storage.json**". Use JSON format for saving the data. Every time the "save" function is called the file should be overridden, starting from a blank state.

function save(){

{{logic}}

return

}

### Implement the "load" function

The "**load**" function should read a file named "**storage.json**", parse the data, and load all the key-value pairs in memory. If the file does not exist yet, do nothing.

function load(){

{{logic}}

return

}

### Test the "storage" module through a node script

Create an "**index.js**" file and write tests to see whether or not the storage module is working correctly, you can also use the ones provided below, the first test will **work** if you have used **synchronous functions** for the save and load.

|  |  |  |
| --- | --- | --- |
| Sample code usage |  | Corresponding output |
| storage.load()  storage.put('first','firstValue')  storage.put('second','secondValue')  storage.put('third','thirdValue')  storage.put('fouth','fourthValue')  console.log(storage.get('first'))  console.log(storage.getAll())  storage.delete('second')  storage.update('first','updatedFirst')  storage.save()  storage.clear()  console.log(storage.getAll())  storage.load()  console.log(storage.getAll()) | firstValue  { first: 'firstValue',  second: 'secondValue',  third: 'thirdValue',  fouth: 'fourthValue' }  There are no items in the storage  { first: 'updatedFirst',  third: 'thirdValue',  fouth: 'fourthValue' } |

|  |
| --- |
| **Sample code usage** |
| storage.put('first','firstValue')  storage.put('second','secondValue')  storage.delete('second')  storage.delete('second') |
| storage.put(2,'someValue') |
| storage.put('cat','dog')  storage.put('cat','anotherDog') |

|  |
| --- |
| Corresponding output |
| All examples should throw appropriate Error |

**\*** You **must** **have** variable named **"storage"** that is **connected** to the **storage module,** also you **must delete** the **"storage.json"** file, before testing.