***Important concepts in node js:***

***# what is node js:***

* *Node.js is a light weight framework.*
* *Node.js ia run-time environment that allow us to launch frontend and backend.*

**# Types of modules:**

1) core module-core module is the in-built module in node js application

2) local module- local module are modules created locally in your node js application

3) third party module- user-defined module

**#Important core modules:**

* http
* url
* querystring
* path
* fs (file system)
* util

**# export module in node js:**

The module.exports or exports is a special object which is included in every js file in the node.js application by default**.**

**# Various port number:**

* Best port number
* 8081 (react native)
* 4200(angular)
* 3000(express/reactjs)
* 5000(free)
* 4000

**# Javascript callbacks:**

\* Callback functions: "a function is ablock of code that performs a certain task when called."

\* advantage: we can wait for the result of a previous function call and then execute another function call.

**# Set Time out function:**

* This functions calls a (cb) after a specified number of milliseconds (ms)
* Time out must be in the range of 1-2,147,483,647 inclusive.
* If the value is outside that range, its changed to 1 millisecond (ms)

setTimeout (function,milliseconds);

* Function- a function containing a block of code.
* Milliseconds- the time after which the function is executed.

**# clearTimeout(t)**

* The clearTimeout() is used to cancel a timeout that was set with setTimeout().the callback will not execute.

**# setInterval(cb,ms)**

* Setinterval() calls a function (cb) repeatedly at specified intervals(in milliseconds (ms))
* The interval must be in the range of 1-2,147,483,647 inclusive.
* If the value is outside that range, its changed to 1ms.

**# clearInterval(t)**

* The clearInterval() is used to stop a timer that was set with setInterval(). The callback will not execute.

***# Promises :***

* A promise is basically an advancement of callbacks in node. While developing an application you may encounter that you are using a lot of nested callback.
* A promise is an object that allows you to handle asynchronous operations. It’s an alternative to plain old callbacks.
* Promises have many advantages over callbacks.

1. Make the async code easier to read.
2. Provide combined error handling
3. Better control flow. You can have sync actions execute in parallel or series.
4. Promises are used to handle asynchronous http requests.

.then (fulfillement)

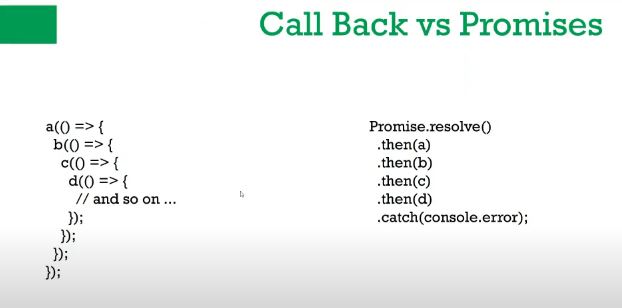
**fulfill**

.catch (on rejection)

promises

reject

**# Callback vs Promises:**



* SYNTAX:
* THEN() : is invoked when a prmise is either resolved or rejected.
* CATCH(): is invoked when a promise is either rejected or some error has occurred in execution

.then(function(result) {

//Handle success

}, function (error){

//handle error

})

***# File Handling Concepts :***

* Blocking: When javascript execution in node.js process(each program) has to wait until a non-js operation completes is called blocking.
* NON-Blocking: This is opposite of the blocking i.e. js execution do not wait until the non-js operation completes.
* **Synchronous**: Synchronous Code is also known as “blocking” because it halts the program until all the resources are available.
* **Asynchronous**: Asynchronous code is also known as “non-blocking” because the program continues executing and doesn’t wait for external resorces(I/O) to be available.

**# Advantages of non-blocking code:**

* More performant.
* Blocking code waste 90% of CPU cycles waiting for the network or disk to get the data.
* Non-blocking code is a straight forward way to have concurrency without having to deal with execution threads.

**# FS Modules:**

It helps to create,read,delete and append file.

Example: var fs = require(‘fs’);

***# NPM :***

* NPM means node package manager
* The npm program is installed on our computer when we install node.js.
* NPM install packages in two modes local and global.
* **PACKAGE.JSON** will store all information regarding application and its name etc. In node.js

***Comparing the scenario with android:***

***Androird -> Mobile->Application-> Whatsapp (click install)***

***->Facebook (click Update)***

In this, we have to simply click button but same thing doesn’t happen in node.js

* ***Update: LocalPackage.json compare with Live Application Number***
* ***Local vs live***

***NODE JS:*** Library 5000 files

***a.*** js 800 mb

***b.***js 900 mb

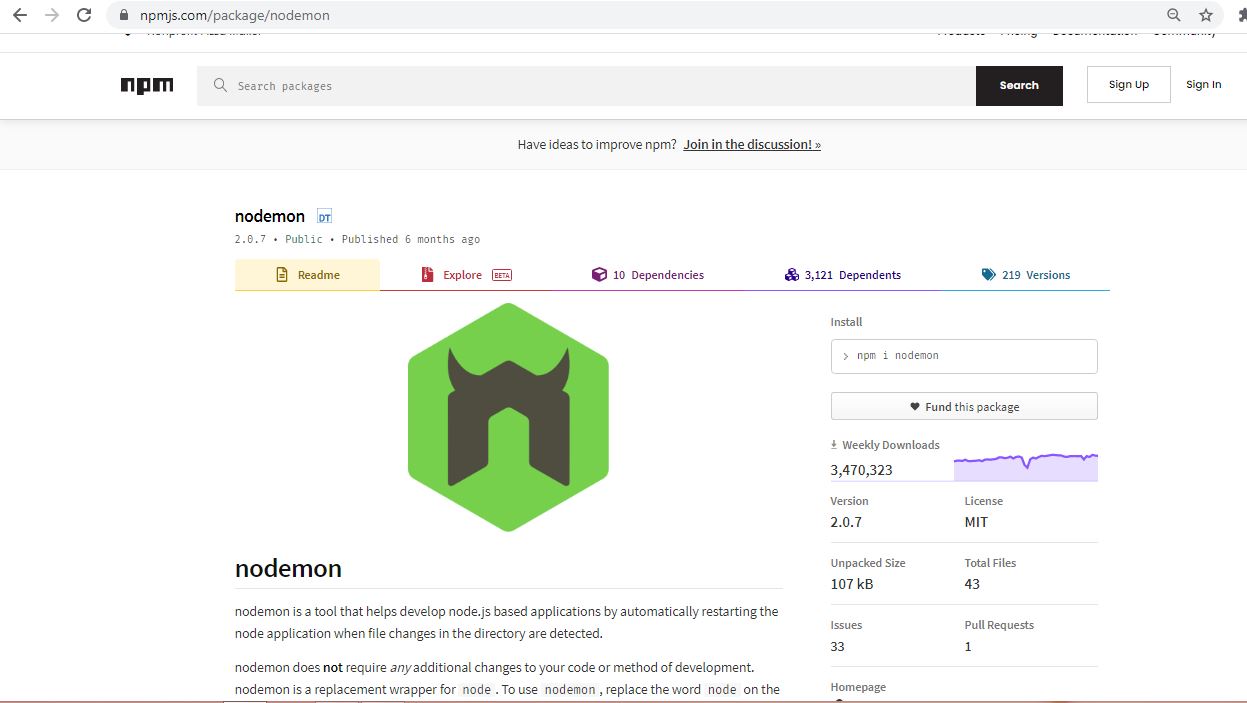
* IN order to save bandwidth or work fast, the concept of global & local must be clear
* If some some error occurs in node.js, then also it can’t be solved by deleting & re-installing npm package.

***What is the use of npm?***

* Npm makes it easy for JS developers to share and reuse code, and itbmakes it easy to update the code that you’re sharing.
* Just like php has composer for downloading decencies which are just code in files.
* Just like python has pip, ruby has gems etc.
* To find various NPM packages visit [**www.npmjs.com**](http://www.npmjs.com)

**Example:** Nodemon package **-> *npm i nodemon*** command is used to install this package

**Others are node restart, chokidar, react, angular, react-native etc.**

****