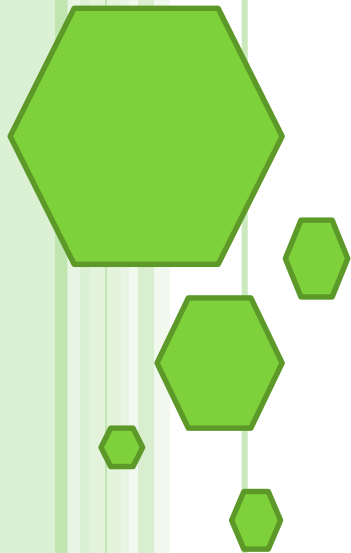


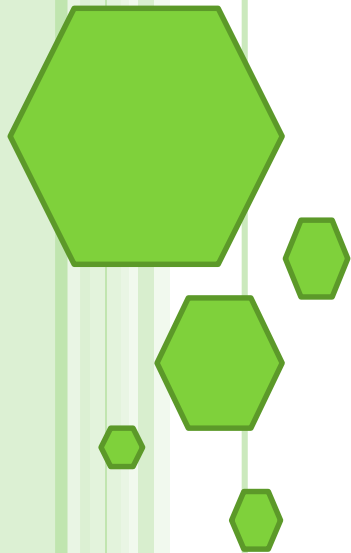


The Server-side JavaScript



Eng. Niveen Nasr El-Den
SD & Gaming CoE.

iTi



DAY 2

EVENTS MODULE

- All objects that emit events are instances of the EventEmitter class.
- These objects expose an `eventEmitter.on()` function that allows one or more Functions to be attached to named events emitted by the object.
- When the EventEmitter object emits an event, all of the Functions attached to that specific event are called *synchronously*
- Any object can become an EventEmitter through inheritance using:
 - **`util.inherits()`** method.
 - **ES6 extends**



EVENTS MODULE

```
//1)require event Module
var myEvent = require("events");

//2)inherit from
class Emitter extends myEvent {}

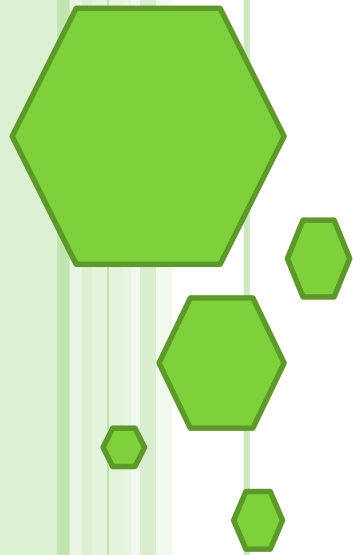
//3)create event obj
var myEmitter = new Emitter();

//listen
▼ myEmitter.on("lookup", function () {
    console.log("lookup event fired");
});

▼ myEmitter.once("lookup", function () {
    console.log("lookup Abdallah");
});

myEmitter.emit("lookup");
```





NODE.JS MODULES

Event Module Demo

HTTP MODULE

```
var http = require("http");

http.createServer(function(request, response) {

    console.log("request recieved");
    response.writeHead(200); //status code in header
    response.write("welcom to nodeJS world!!"); //response body

    //to close the connection
    response.end(); // so client knows it has recieved all data

});

//http.listen(3000, "127.0.0.1");
http.listen(3000);

// to ensure that server is running
console.log("listening on port 3000...");
```

MEDIA | MIME TYPE

- MIME stands for Multipurpose Internet Mail Extensions
- IANA (Internet Assigned Numbers Authority) is responsible for all official MIME types https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Complete_list_of_MIME_types
- MIME types for most common files
 - html → 'text/html'
 - txt → 'text/plain'
 - css → 'text/css'
 - gif → 'image/gif'
 - png → 'image/png'
 - jpg | jpeg → 'image/jpeg'
 - js → 'application/javascript' | 'text/javascript'
 - json → 'application/json' | 'text/json'
 - etc.. <https://www.sitepoint.com/mime-types-complete-list/>
- Browsers use the MIME type, to determine how to process a URL, not file's extension
- web servers send the correct MIME type in its response header via "Content-Type"



HTTP HEADERS

- HTTP headers allow the client and the server to pass additional information with the request or the response.
- Example:
 - Content-Type
 - Set-Cookie
 - Access-Control-Allow-Origin
 - etc..



HTTP VERBS

○ GET

- get data

○ POST

- create data

○ PUT

- update data

○ DELETE

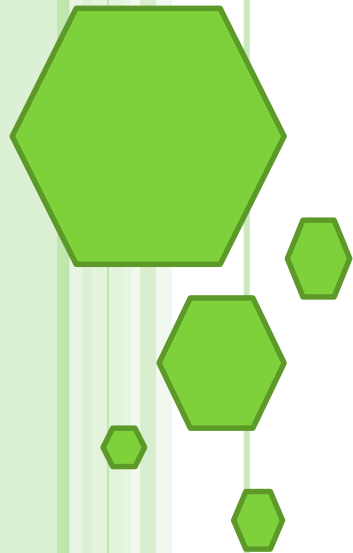
- delete data

○ PATCH

- update part of data

○ ...





NODE.JS MODULES

HTTP server Demo

CUSTOM MODULE

- Creating a module is easy, just put your JavaScript code in a separate .js file and include it in your code by using keyword `require`, like:

```
var myModule = require("./myModule.js");
```

- To require local created module, we have to pass relative path to its .js file; so nodejs knows that its not going to looking for it in `node_module` folder.



CUSTOM MODULE

- **require()** looks for module based upon passed parameter
 - “../module_nm” – looks in parent directory
 - “./module_nm” – looks in same directory
 - “/module_nm” – looks in that directory (absolute)
 - “module_nm” – search for a folder with this name and use “index.js” as the required module (it presented) if not; it searches inside node-module of application directory then search globally
- Inside node-module there is a directory for each module



CUSTOM MODULE & SCOPE

- `module.exports` useful to be used with constructor function or where the entire module is a single function otherwise use `exports`

```
var items=[];

function addItem (item,price){
    items.push({
        item_nm:item,
        item_price:price
    });
}

function total(){
    return items.reduce(function(prev,curr){
        return prev+curr.price;
    },0);
}

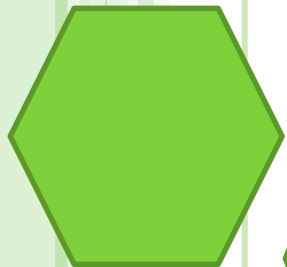
exports.addItem = addItem;
exports.total = total;
```

```
function Cart(){
    this.items=[];
}

Cart.prototype.addItem =function(item,price){
    this.items.push({
        item_nm:item,
        item_price:price
    });
}

Cart.prototype.total=function(){
    return this.items.reduce(function(prev,curr){
        return prev+curr.item_price;
    },0);
}

module.exports = Cart;
```



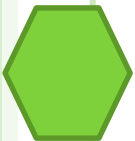
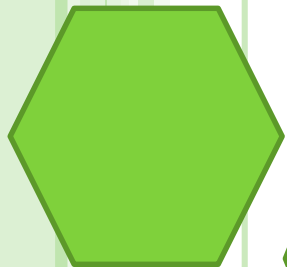
CUSTOM MODULES

Demo

ES6-MODULE IN NODEJS

- Both module exports and its importing files should be with extension **.mjs**
- To run the file importing your module you should run the following cmd
node --experimental-module fnm.mjs





ASSIGNMENTS