Quiz #3

Q1 NodeJS's EventEmitter

5 Points

Node.js uses several programming patterns to help handle blocking functionality in the single-threaded JavaScript runtime. The patterns include using callback functions, promises, and the listener/emitter pattern (e.g. Node.js's EventEmitter module - `require('events')`).

Assume you are given a NodeJS's EventEmitter object that documents the following events:

Event	Parameters
EventA	`(arg1, arg2)`
EventB	`()`

Your project lead is far more comfortable with callback functions than EventEmitters and asks you to write a function that converts the above EventEmitter to use callback functions.

Write a `ConvertEmitterToCallback` routine that takes the emitter and a callback function and turns emitted events into calls to the callback function. The first parameter to the callback function should be the event that was emitted with the remaining arguments being parameters to the event. For example, `emit('EventA', 1, 2)` would generate a `callback('EventA', 1, 2)` call.

Demonstrate your understanding of EventEmitters by showing the code needed to perform this conversion to callbacks:

```
function ConvertEmitterToCalllback(emitter, callback) {
```

Enter your answer here

```
}
```

Save Answer

Q2 EventEmitter mistakes

4 Points

Sometimes you can just look at code and tell the programmer likely made a mistake. For example, the code might obviously generate an error (e.g. `let o; o.prop = 1;`) or the code might not do anything useful (e.g. `x = 1; x = 3;`). For each of the following pairs of statements, identify if there looks to be a likely mistake or if it could be correct code.

Assume you have an EventEmitter named `myEmitter` in scope:

Q2.1

1 Point

```
myEmitter.emit('X', 'Hi');
myEmitter.emit('Y', 'Bye');
```

○ code will generate an error

○ code doesn't do anything useful

○ code could be correct

Save Answer

Q2.2

1 Point

```
myEmitter.emit('X',  (x) => console.log('one',x));
myEmitter.emit('Y',  (x) => console.log('two',x));
```

○ code will generate an error

○ code doesn't do anything useful

○ code could be correct

Save Answer

Q2.3

1 Point

```
myEmitter.on('X', 'Hi');
myEmitter.on('Y', 'Bye');
```

○ code will generate an error

○ code doesn't do anything useful

○ code could be correct

Save Answer

Q2.4

1 Point

```
myEmitter.on('X',  (x) => console.log('one',x));
myEmitter.on('Y',  (x) => console.log('two',x));
```

○ code will generate an error

○ code doesn't do anything useful

○ code could be correct

Save Answer

Q3 async module

3 Points

The node async module (i.e. `require('async')`) usage looks like:

```
async.each(items, function iteratee(item, callback) {  ...
```

where the first argument (`items`) is the collection and the second argument is the iteratee function that is called on each item. The `iteratee` function is defined with two parameters `item` and `callback`. `item` is the item from the collection being processed and `callback` is a function.

Describe what the function `callback` does when called:

Enter your answer here

Save Answer

Q4 ExpressJS

5 Points

An ExpressJS handler is always passed three arguments: two objects (`httpRequest` and `httpResponse`) and a function (`next`) like:

```
function (httpRequest, httpResponse, next)
```

Express handlers access properties and call methods on the `httpRequest` and `httpResponse` objects and sometimes will call the `next` function. Show your understanding by answering the following questions about this:

Q4.1

3 Points

Explain why a handler might contain code that modifies the incoming `httpRequest` object and then subsequently calls the `next` function. Include an example of why this ordering might be done.

Enter your answer here

Save Answer

Q4.2

2 Points

Explain why ExpressJS handlers that call `httpResponse.send()` rarely subsequently call `next`.

Enter your answer here

Save Answer

Q5 Databases

3 Points

Object relational mapping (ORM) converts from the object data model to the relational data model. Object definition languages (ODL) are found in ORM systems and Mongoose help making mapping to the relational model possible. Explain why without an ODL like Mongoose, mapping of MongoDB objects (i.e. documents) to the relational model is difficult.

Enter your answer here

Save Answer

Q6 Session Hijack Attack

3 Points

Web application backends using the ExpressJS session middleware leave the web application exposed to attacks that steal the session cookie but not attacks involving forging session cookies. Explain how forging session cookies is made difficult by the ExpressJS session module.

Enter your answer here

Save Answer

Q7 Session state

3 Points

If you had a web application structured like our photo app and you suddenly lost all the session state in your backend, what ill effects would you expect the users of your web applications to see? Be specific.

Enter your answer here

Save Answer

Q8 Input Validation

4 Points

Answer the following questions about input validation.

Q8.1

2 Points

Explain why validation needs to be done in the backend even if your frontend can do all the same validation of input.

Enter your answer here

Save Answer

Q8.2

2 Points

Explain why validation is done in the frontend even if we have complete validation in the backend.

Enter your answer here

Save Answer

Q9 Sanitization

3 Points

Explain what is meant by a module that sanitizes HTML to make sure it is free of cross site scripting attacks. Include a description of what the sanitization process must do.

Enter your answer here

Save Answer

Q10 Attacks and defenses

6 Points

When talking about the various attacks that web applications might be subjected to, we presented several helpful technologies including encryption/decryption, message authentication codes, and HTTPS certificates. For each of the following attack types, select which of these technologies can be helpful.

Q10.1

1.5 Points

Eavesdropper attacks

□ encryption/decryption

□ message authentication codes

□ HTTPS certificates

□ None of the above

Save Answer

Q10.2

1.5 Points

Denial of service attacks

□ encryption/decryption

□ message authentication codes

□ HTTPS certificates

□ None of the above

Save Answer

Q10.3

1.5 Points

Phishing attacks

□ encryption/decryption

□ message authentication codes

□ HTTPS certificates

□ None of the above

Save Answer

Q10.4

1.5 Points

SQL injection attacks

□ encryption/decryption

□ message authentication codes

□ HTTPS certificates

□ None of the above

Save Answer

Q11 Scale-out

3 Points

Explain why scale-out architectures are known for handling failures better than scale-up architectures.

Enter your answer here

Save Answer

Q12 CDN

3 Points

A Content Distribution Network (CDN) doesn't work well for data that needs to be updated frequently. Explain what the problem is with updating information in a CDN.

Enter your answer here

Save Answer

Save All Answers

Submit & View Submission ›