edX

Course > Expres... > Modul... > Mod02...

Mod02 Assessment

🔖 **Bookmark this page**

Question 1

1/1 point (graded)

What statement(s) is(are) true about Express? (Choose four that apply)

☑ With Express developers can implement server for static content (CSS, HTML, JS)

☑ Express offers better code organization and modularization

☑ Express can be used for building APIs (GraphQL, REST, etc.)

☑ Express has a built-in mechanism to render HTML using template engines

☐ Express framework is installed with `npm i -g express`

✔️

Submit	You have used 1 of 1 attempt

✔ Correct (1/1 point)

Question 2

1/1 point (graded)

```
express -c styl express-styl
```

What will the above command do?

○ Launch an Express server

◉ Generate Express project in express-styl folder with support of Stylus
   ✔

○ Install express-styl plugin for Express

○ Create Express route express-styl

○ Order kombucha drone delivery

| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)

Question 3

1/1 point (graded)

If the current version of Express is 4.16.2, what commands will install Express with this version as a project dependency? (Choose three that apply)

☐ `npm install -g express@4.16.2`

☑ `npm i express@latest -SE`

☑ `npm i express@4.16`

☐ `npm install`

☑ `npm i express --save-dev`

✔

Submit   You have used 1 of 1 attempt

✔ Correct (1/1 point)

Question 4

1/1 point (graded)

```
const express = require('express')
const app = express()

app.get('/', (req, res) => {
  //TODO
})

app.listen(3000)
```

What would be the best code to be put in place of "TODO" so the application can become a working HTTP server that returns 'hello world'?
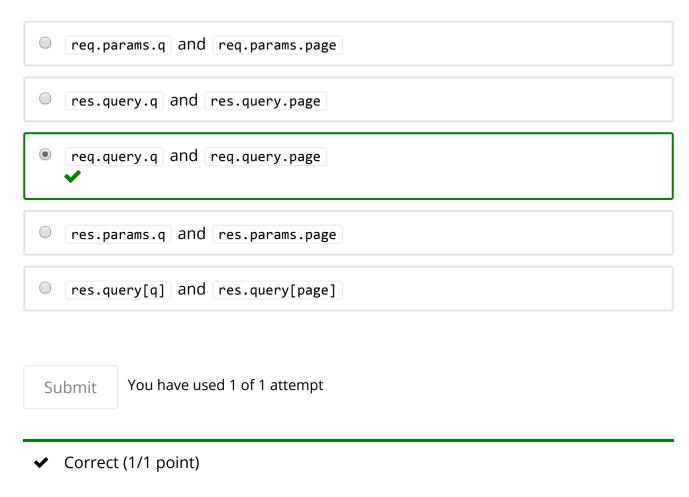
○ `response.send('hello world')`

○ `res.write('hello world')`

◉ `res.send('hello world')`
   ✔

○ `req.send(200)`

○ Nothing. The code above is wrong so nothing in place of "TODO" will fix it.

Submit   You have used 1 of 1 attempt

✔ Correct (1/1 point)

Question 5

1/1 point (graded)

URL: `http://node.university/p/aws-intro/q=express&page=1`

What is the right way to access the data of "q" and "page" shown in the URL above in Express?

- ○ `req.params.q` and `req.params.page`

- ○ `res.query.q` and `res.query.page`

- ◉ `req.query.q` and `req.query.page`
  ✔

- ○ `res.params.q` and `res.params.page`

- ○ `res.query[q]` and `res.query[page]`

| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)

Question 6

1/1 point (graded)

URL: `http://node.university/p/aws-intro/?q=express&page=1`

What is the right way to access the data of "aws-intro" shown in the URL above in Express?

- ○ `/param=name` as URL pattern and `req.params.name` in request handler

- ○ `/p/:name` as URL pattern and `res.params.name` in request handler

- ○ `/p/:name` as URL pattern and `req.param.name` in request handler

○ `/p/:name` as URL pattern and `res.param.name` in request handler

◉ `/p/:name` as URL pattern and `req.params.name` in request handler
   ✔

| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)

Question 7

1/1 point (graded)

What is the valid response method in Express to *complete* the HTTP response? (Choose two that apply)

☐ `res.status()`

☐ `res.write()`

☑ `res.end()`

☐ `res.kill()`

☑ `res.sendfile()`

✔

| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)

Question 8

1/1 point (graded)

What is a valid way to define a PUT route to update one message by ID?

⚪ `app.put.message(updateMessage)`

⚪ `appPut('/messages', updateMessage)`

🔘 `app.put('/messages/:id', updateMessage)`
✔️

⚪ `app.updateMessage(id)`

⚪ `app.update.message(id)`

| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)

Question 9

1/1 point (graded)

```
const dbRef = db.connect()
app.use((request, response, next) => {
  request.database = dbRef
})
```

What is wrong with this request handler? (Choose two that apply)

☐ `db` is undefined

☑ It will timeout

☐ No `error` argument

☑ No `response.send()`

☐ No semicolons

✔

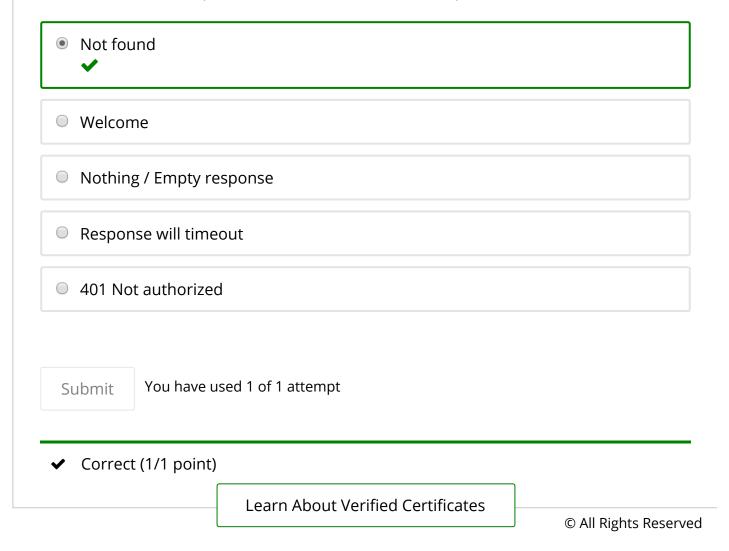| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)
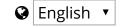

Question 10

1/1 point (graded)

```
const express = require('express')
let app = express()

let store = {accounts: require('./data/accounts.json')}
const bodyParser = require('body-parser')

app.use(bodyParser.json())

app.get('/', (req, res) => {
  res.send('Welcome')
})

app.all('*', (req, res) => {
  res.status(404).send('Not found')
})

app.get('/accounts', (req, res) => {
  res.status(200).send(store.accounts)
})

app.put('/accounts/:id', (req, res) => {
  store.accounts[req.params.id] = req.body
  res.status(200).send(store.accounts[req.params.id])
})

app.listen(3000)
```

What could be the response off a PUT /accounts/123 request?

⊙ Not found
✔

○ Welcome

○ Nothing / Empty response

○ Response will timeout

○ 401 Not authorized

| Submit | You have used 1 of 1 attempt |

✔ Correct (1/1 point)

Learn About Verified Certificates

© All Rights Reserved

🌐 English ▾

POWERED BY
OPENedX®