Lab 9.1 - Implement a Service That Is Not Vulnerable to Parameter Pollution

The labs-1 folder contains the following files:

- `package.json`
- `app.js`
- `validate.js`

The `package.json` file contains the following:

```
{
  "name": "labs-1",
  "version": "1.0.0",
  "scripts": {
    "start": "node app.js"
  },
  "license": "UNLICENSED",
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

Note that Express is a dependency of the project. Install the project dependency with the following command, executed within the labs-1 folder:

```
npm install
```


The `app.js` file contains the following:

```
'use strict'
const express = require('express')
const app = express()
const router = express.Router()
const { PORT = 3000 } = process.env

router.get('/', (req, res) => {
  setTimeout(() => {
    res.send((req.query.un || '').toUpperCase())
  }, 1000)
})

app.use(router)

app.listen(PORT, () => {
  console.log(`Express server listening on ${PORT}`)
})
```

This is a small Express service that uppercases any input sent via a **un** query string parameter, but it waits one second before sending the response.

This service is vulnerable to parameter pollution. A URL such as http://localhost:3000/?un=a&un=b will cause the service to crash, assuming the service is listening on port 3000.

Fix it, without changing any of the current functionality.

The parameter pollution attack may be handled as seen fit. For instance upper casing all forms, or sending a 400 Bad Request, or any kind of response. The only thing that must not happen is the service crashing and requests containing query-strings with a single **un** parameter must continue to respond with the uppercased version of that value.

Run the `validate.js` file as follows, to validate the fix:

```
node validate.js
```

If successful this should output something similar to the following: