



ECSE 428 Assignment B


Jeffery Tang 260791531



Zhao Liang 260781081


Note

[ECSE 428] Extension for Assignment B 

 **Jeffery Tang**
(No message text) Tue 11/2/2021 12:04 AM

 **Jeffery Tang**
Thank you very much sir! Tue 11/2/2021 12:03 AM

 **Robert Sabourin** <robsab@gmail.com>
Mon 11/1/2021 10:22 PM
To: Jeffery Tang
Sure, that's fine.


 **Jeffery Tang**
Mon 11/1/2021 8:38 PM
To: Robert Sabourin <robsab@gmail.com>
Dear Prof. Sab,

This weekend is my busiest since the start of the semester, I have 55% worth of stuff due on Sunday and the day after, I was wondering if I could get a two-day extension on assignment B, say Nov 9 at 11:30pm. If you can grant it, I would appreciate it a lot!

Sincerely,
Jeffery

How to Install and Launch the Application

Check README.md, should be in the same directory

List of Tests

ID	Name (maybe more info)	Valid Domain Logic?
01	Standard envelope, lightweight $140 \leq L \leq 245$ $90 \leq W \leq 156$ $3 \leq Wt \leq 30$ \$0.49	Yes
02	Standard envelope, regular weight $140 \leq L \leq 245$ $90 \leq W \leq 156$ $30 < Wt \leq 50$ \$0.8	Yes
03	Non-standard envelope, due to heaviness $140 \leq L \leq 245$ $90 \leq W \leq 156$ $50 < Wt \leq 100$ \$0.98	Yes
04	Non-standard envelope, due to heaviness, very heavy $140 \leq L \leq 245$ $90 \leq W \leq 156$ $100 < Wt \leq 500$ \$2.4	Yes
05	Non-standard envelope, due to length $245 < L \leq 380$ $90 \leq W \leq 156$ $3 \leq Wt \leq 100$ \$0.98	Yes
06	Non-standard envelope, due to length and weight $245 < L \leq 380$ $90 \leq W \leq 156$ $100 < Wt \leq 500$ \$2.4	Yes
07	Non-standard envelope, due to width $140 \leq L \leq 380$	Yes

	$156 < W \leq 270$ $3 \leq Wt \leq 100$ \$0.98	
08	Non-standard envelope, due to width and weight $140 \leq L \leq 380$ $156 < W \leq 270$ $100 < Wt \leq 500$ \$2.4	Yes
09	Missing required arguments	No
10	Length too short <140mm	No
11	Length too long >380mm	No
12	Width too short <90mm	No
13	Width too long >270mm	No
14	Weight too light <3g	No
15	Weight too heavy >500g	No
16	Distance unit not in mm or in	No
17	Weight unit not in g or oz	No
18	Invalid request payload syntax	No
19	Length is not a number	No
20	Width is not a number	No
21	Weight is not a number	No

Note: For the valid domain logic tests above, for the constraints specified, length is in mm and weight in g. Those tests also check for other available units. Same with ID10-15.

Step by Step TDD with Screenshots

Setup

Mock request and response:

```
const res = {
  code: 200,
  text: "",
  send(response) {
    this.text = response;
  },
  status(code) {
    this.code = code;

    return this;
  },
  sendStatus(code) {
    this.status(code).send(code);
  },
};

const req = {
  body: {
    length: 200,
    width: 156,
    weight: 10,
  },
};
```

Helper functions for assertion:

```
function expected(req, status, text) {
  ramp(req, res);

  if (text === undefined) {
    // 400 status code
    expect(res.code).toBe(status);
  } else {
    expect({ status: res.code, text: res.text }).toEqual({ status, text });
  }
  // reset it
  res.code = 200;
  res.text = "";
}
```

```

function assertForAllUnits(length_mm, width_mm, weight_g, postalRate) {
  const req = {};
  // Asserts for every combination of units
  for (const distanceUnit of [null, "mm", "in"]) {
    for (const weightUnit of [null, "g", "oz"]) {
      let length = length_mm;
      let width = width_mm;
      let weight = weight_g;
      if (distanceUnit === "in") {
        length = mm_to_in(length);
        width = mm_to_in(width);
      }
      if (weightUnit === "oz") {
        weight = g_to_oz(weight);
      }
      req.body = { length, width, weight };

      if (distanceUnit !== null) {
        req.body.distanceUnit = distanceUnit;
      }
      if (weightUnit !== null) {
        req.body.weightUnit = weightUnit;
      }

      expected(req, postalRate === undefined ? 400 : 200, postalRate);
    }
  }
}

```

Empty code:

```

function ramp(req, res) {

}

module.exports = ramp

```

ID01

Test:

```
describe("Valid Domain Logic", () => {
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});
```

Fail:

```
FAIL ./ramp.test.js
Valid Domain Logic
  ✖ ID01: Standard envelope, lightweight (4 ms)

  ● Valid Domain Logic › ID01: Standard envelope, lightweight

    expect(received).toEqual(expected) // deep equality

    Expected: {"status": 200, "text": 0.49}
    Received: undefined

      36 |     expect(resReturn.status).toBe(status);
      37 |   } else {
    > 38 |     expect(resReturn).toEqual({ status, text });
         |                           ^
      39 |   }
      40 | }
      41 |

    at expected (ramp.test.js:38:23)
    at assertForAllUnits (ramp.test.js:103:7)
    at ramp.test.js:75:5

Test Suites: 1 failed, 1 total
Tests: 1 failed, 1 total
Snapshots: 0 total
Time: 0.434 s, estimated 1 s
Ran all test suites.
```

Code:

```
function ramp(req, res) {
  const { length, width, weight, distanceUnit, weightUnit } = req.body;
  var postalRate = 0.49;
  res.send(`${postalRate}`);
}

module.exports = ramp;
```

Pass:

```
> ecse428_assignmentb@1.0.0 test /Users/zhaoliang/Documents/ECSE 428/ECSE428_AssignmentB
> jest

PASS ./ramp.test.js
  Valid Domain Logic
    ✓ ID01: Standard envelope, lightweight (5 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.96 s, estimated 1 s
Ran all test suites.
```

ID02

Test:

```
describe("Valid Domain Logic", () => {
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],
    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});
```

Fail:

```
FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (4 ms)
  ✗ ID02: Standard envelope, regular weight (9 ms)

● Valid Domain Logic > ID02: Standard envelope, regular weight

expect(received).toEqual(expected) // deep equality

- Expected - 1
+ Received + 1

Object {
  "status": 200,
-  "text": "0.8",
+  "text": "0.49",
}

34 |     expect(res.code).toBe(status);
35 |   } else {
> 36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
    |                                     ^
37 |   }
38 |   // reset it
39 |   res.code = 200;

at expected (ramp.test.js:36:50)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:81:5

Test Suites: 1 failed, 1 total
Tests: 1 failed, 1 passed, 2 total
Snapshots: 0 total
Time: 0.954 s, estimated 1 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
```

Code (every line but the first and last inside the function is different):

```
function ramp(req, res) {
  const { length, width, weight, distanceUnit, weightUnit } = req.body;
  var postalRate = 0;
  var real_weight = weight;

  if (weightUnit == "oz") {
    real_weight = oz_to_g(weight);
  }

  if (real_weight > 30) {
    postalRate = 0.8;
  } else {
    postalRate = 0.49;
  }

  res.send(`${postalRate}`);
}
```


Pass:

```
> ecse428_assignment@1.0.0 test /Users/zhaoliang/Documents/ECSE 428/ECSE428_AssignmentB
> jest

PASS ./ramp.test.js
  Valid Domain Logic
    ✓ ID01: Standard envelope, lightweight (5 ms)
    ✓ ID02: Standard envelope, regular weight (10 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.134 s
Ran all test suites.
```

ID03

Test:

```
describe("Valid Domain Logic", () => {
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],
    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],
    ["ID03: Non-standard envelope, due to heaviness", 200, 156, 60, "0.98"],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});
```

Fail:

```
> ecse428_assignmentB@1.0.0 test /Users/zhaojiang/Documents/ECSE 428/ECSE428_AssignmentB
> jest

FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (8 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✗ ID03: Non-standard envelope, due to heaviness (6 ms)

● Valid Domain Logic › ID03: Non-standard envelope, due to heaviness

    expect(received).toEqual(expected) // deep equality

    - Expected  - 1
    + Received  + 1

    Object {
      "status": 200,
    -   "text": "0.98",
    +   "text": "0.8",
    }

      34 |         expect(res.code).toBe(status);
      35 |       } else {
    >   36 |         expect({ status: res.code, text: res.text }).toEqual({ status, text });
         |                                     ^
      37 |       }
      38 |       // reset it
      39 |       res.code = 200;

      at expected (ramp.test.js:36:50)
      at assertForAllUnits (ramp.test.js:109:7)
      at ramp.test.js:81:5

Test Suites: 1 failed, 1 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:   0 total
Time:        0.877 s, estimated 2 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.
```

Code (lines 10-12, 15, 22-24):

```

1 |   const { mm_to_in, in_to_mm, g_to_oz, oz_to_g } = require("./helper");
2 |   function ramp(req, res) {
3 |       const { length, width, weight, distanceUnit, weightUnit } = req.body;
4 |       var isStandard = false;
5 |       var postalRate = 0;
6 |       var real_weight = weight;
7 |       if (weightUnit == "oz") {
8 |         real_weight = oz_to_g(weight);
9 |       }
10 |      if (3 <= real_weight && real_weight <= 50) {
11 |        isStandard = true;
12 |      }
13 |
14 |      //calculate postalRate
15 |      if (isStandard) {
16 |        if (real_weight > 30) {
17 |          postalRate = 0.8;
18 |        } else {
19 |          postalRate = 0.49;
20 |        }
21 |      }
22 |      else {
23 |        postalRate = 0.98;
24 |      }
25 |      res.send(`${postalRate}`);
26 |    }

```

Pass:

```

> ecse428_assignmentB@1.0.0 test /Users/zhaoliang/Documents/ECSE 428/ECSE428_AssignmentB
> jest

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (12 ms)
  ✓ ID02: Standard envelope, regular weight (4 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (10 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        1.8 s
Ran all test suites.

```

ID04

Test:

```

describe("Valid Domain Logic", () => {
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],

    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],

    ["ID03: Non-standard envelope, due to heaviness", 200, 156, 60, "0.98"],

    [
      "ID04: Non-standard envelope, due to heaviness, very heavy",
      200,
      156,
      300,
      "2.4",
    ],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});

```

Fail:

```

> ecse428_assignmentb@1.0.0 test /Users/zhaojiang/Documents/ECSE 428/ECSE428_AssignmentB
> jest

FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (5 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (2 ms)
  ✗ ID04: Non-standard envelope, due to heaviness, very heavy (11 ms)

● Valid Domain Logic > ID04: Non-standard envelope, due to heaviness, very heavy

    expect(received).toEqual(expected) // deep equality

    - Expected   - 1
    + Received   + 1

    Object {
      "status": 200,
    -   "text": "2.4",
    +   "text": "0.98",
    }

      34 |     expect(res.code).toBe(status);
      35 |   } else {
    >  36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
         |                                             ^
      37 |   }
      38 |   // reset it
      39 |   res.code = 200;

      at expected (ramp.test.js:36:50)
      at assertForAllUnits (ramp.test.js:109:7)
      at ramp.test.js:81:5

Test Suites: 1 failed, 1 total
Tests:       1 failed, 3 passed, 4 total
Snapshots:   0 total
Time:        1.064 s, estimated 2 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.

```

Code (lines 22-26):

```
1 | const { mm_to_in, in_to_mm, g_to_oz, oz_to_g } = require("./helper");
2 | function ramp(req, res) {
3 |   const { length, width, weight, distanceUnit, weightUnit } = req.body;
4 |   var isStandard = false;
5 |   var postalRate = 0;
6 |   var real_weight = weight;
7 |   if (weightUnit == "oz") {
8 |     real_weight = oz_to_g(weight);
9 |   }
10 |   if (3 <= real_weight && real_weight <= 50) {
11 |     isStandard = true;
12 |   }
13 |
14 |   //calculate postalRate
15 |   if (isStandard) {
16 |     if (real_weight > 30) {
17 |       postalRate = 0.8;
18 |     } else {
19 |       postalRate = 0.49;
20 |     }
21 |   } else {
22 |     if (real_weight <= 100) {
23 |       postalRate = 0.98;
24 |     } else {
25 |       postalRate = 2.4;
26 |     }
27 |   }
28 |   res.send(`${postalRate}`);
29 | }
```

Pass:

```
> ecse428_assignment@1.0.0 test /Users/zhaoliang/Documents/ECSE 428/ECSE428_AssignmentB
> jest

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (5 ms)
  ✓ ID02: Standard envelope, regular weight (1 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (3 ms)

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        0.88 s, estimated 1 s
Ran all test suites.
```

ID05

Test:

```
describe("Valid Domain Logic", () => {
  Debug
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],

    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],

    ["ID03: Non-standard envelope, due to heaviness", 200, 156, 60, "0.98"],

    [
      "ID04: Non-standard envelope, due to heaviness, very heavy",
      200,
      156,
      300,
      "2.4",
    ],

    ["ID05: Non-standard envelope, due to length", 330, 156, 10, "0.98"],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});
```

Fail:

```

FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (17 ms)
  ✓ ID02: Standard envelope, regular weight (4 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (4 ms)
  ✗ ID05: Non-standard envelope, due to length (14 ms)

● Valid Domain Logic > ID05: Non-standard envelope, due to length

expect(received).toEqual(expected) // deep equality

- Expected   - 1
+ Received   + 1

Object {
  "status": 200,
-  "text": "0.98",
+  "text": "0.49",
}

34 |     expect(res.code).toBe(status);
35 |   } else {
> 36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
    |                                         ^
37 |   }
38 |   // reset it
39 |   res.code = 200;

at expected (ramp.test.js:36:50)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:81:5

Test Suites: 1 failed, 1 total
Tests:       1 failed, 4 passed, 5 total
Snapshots:  0 total
Time:        1.71 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.

```

Code (8-10, 15):

```

1 | const { mm_to_in, in_to_mm, g_to_oz, oz_to_g } = require("./helper");
2 | function ramp(req, res) {
3 |   const { length, width, weight, distanceUnit, weightUnit } = req.body;
4 |   var isStandard = false;
5 |   var postalRate = 0;
6 |   var real_weight = weight;
7 |   var real_length = length;
8 |   if (distanceUnit == "in") {
9 |     real_length = in_to_mm(length);
10 |   }
11 |   if (weightUnit == "oz") {
12 |     real_weight = oz_to_g(weight);
13 |   }
14 |   if (3 <= real_weight && real_weight <= 50) {
15 |     if (140 <= real_length && real_length <= 245) {
16 |       isStandard = true;
17 |     }
18 |   }
19 |
20 |   //calculate postalRate
21 |   if (isStandard) {
22 |     if (real_weight > 30) {
23 |       postalRate = 0.8;
24 |     } else {
25 |       postalRate = 0.49;
26 |     }
27 |   } else {
28 |     if (real_weight <= 100) {
29 |       postalRate = 0.98;
30 |     } else {
31 |       postalRate = 2.4;
32 |     }
33 |   }
34 |   res.send(`${postalRate}`);
35 | }

```

Pass:

```

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (8 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
  ✓ ID05: Non-standard envelope, due to length (2 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        1.399 s
Ran all test suites.

```


ID06

Test:

```
describe("Valid Domain Logic", () => {
  Debug
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],

    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],

    ["ID03: Non-standard envelope, due to heaviness", 200, 156, 60, "0.98"],

    [
      "ID04: Non-standard envelope, due to heaviness, very heavy",
      200,
      156,
      300,
      "2.4",
    ],

    ["ID05: Non-standard envelope, due to length", 330, 156, 10, "0.98"],

    [
      "ID06: Non-standard envelope, due to length and weight", //should pass
      300,
      100,
      150,
      "2.4",
    ],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});
```

Fail:

It passed directly without changing any code from the previous test because we have already checked length and weight

Pass:

```
PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (5 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
  ✓ ID05: Non-standard envelope, due to length (2 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (3 ms)

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        1.055 s, estimated 2 s
Ran all test suites.
```

ID07

Test:

```
describe("Valid Domain Logic", () => {
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],

    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],

    ["ID03: Non-standard envelope, due to heaviness", 200, 156, 60, "0.98"],

    [
      "ID04: Non-standard envelope, due to heaviness, very heavy",
      200,
      156,
      300,
      "2.4",
    ],

    ["ID05: Non-standard envelope, due to length", 330, 156, 10, "0.98"],

    [
      "ID06: Non-standard envelope, due to length and weight", //should pass
      300,
      100,
      150,
      "2.4",
    ],

    ["ID07: Non-standard envelope, due to width", 200, 200, 10, "0.98"],
  ])("%s", (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});
```

Fail:

```
FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (4 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (8 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (5 ms)
  ✓ ID05: Non-standard envelope, due to length (1 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (2 ms)
  ✗ ID07: Non-standard envelope, due to width (46 ms)

● Valid Domain Logic > ID07: Non-standard envelope, due to width

expect(received).toEqual(expected) // deep equality

- Expected   - 1
+ Received   + 1

Object {
  "status": 200,
-  "text": "0.98",
+  "text": "0.49",
}

   34 |     expect(res.code).toBe(status);
   35 |   } else {
>  36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
      |                                     ^
   37 |   }
   38 |   // reset it
   39 |   res.code = 200;

at expected (ramp.test.js:36:50)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:81:5

Test Suites: 1 failed, 1 total
Tests:       1 failed, 6 passed, 7 total
Snapshots:  0 total
Time:        1.326 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.
```

Code (line 11, 18):

```

2  function ramp(req, res) {
3      const { length, width, weight, distanceUnit, weightUnit } = req.body;
4      var isStandard = false;
5      var postalRate = 0;
6      var real_weight = weight;
7      var real_length = length;
8      var real_width = width;
9      if (distanceUnit == "in") {
10         real_length = in_to_mm(length);
11         real_width = in_to_mm(width);
12     }
13     if (weightUnit == "oz") {
14         real_weight = oz_to_g(weight);
15     }
16     // check if this is a standard envelop
17     if (140 <= real_length && real_length <= 245) {
18         if (90 <= real_width && real_width <= 156) {
19             if (3 <= real_weight && real_weight <= 50) {
20                 isStandard = true;
21             }
22         }
23     }
24
25     //calculate postalRate
26     if (isStandard) {
27         if (real_weight > 30) {
28             postalRate = 0.8;
29         } else {
30             postalRate = 0.49;
31         }
32     } else {
33         if (real_weight <= 100) {
34             postalRate = 0.98;
35         } else {
36             postalRate = 2.4;
37         }
38     }
39     res.send(`${postalRate}`);

```

Pass:

```

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (7 ms)
  ✓ ID02: Standard envelope, regular weight (10 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (2 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (13 ms)
  ✓ ID05: Non-standard envelope, due to length (14 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (15 ms)
  ✓ ID07: Non-standard envelope, due to width (12 ms)

Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        1.904 s
Ran all test suites.

```

ID08

Test:

```

describe("Valid Domain Logic", () => {
  Debug
  test.each([
    // test name, length (mm), width (mm), weight (g), postal rate
    ["ID01: Standard envelope, lightweight", 200, 156, 10, "0.49"],

    ["ID02: Standard envelope, regular weight", 200, 156, 45, "0.8"],

    ["ID03: Non-standard envelope, due to heaviness", 200, 156, 60, "0.98"],

    [
      "ID04: Non-standard envelope, due to heaviness, very heavy",
      200,
      156,
      300,
      "2.4",
    ],

    ["ID05: Non-standard envelope, due to length", 330, 156, 10, "0.98"],

    [
      "ID06: Non-standard envelope, due to length and weight", //should pass
      300,
      100,
      150,
      "2.4",
    ],

    ["ID07: Non-standard envelope, due to width", 200, 200, 10, "0.98"],

    [
      "ID08: Non-standard envelope, due to width and weight", // should pass
      200,
      200,
      150,
      "2.4",
    ],
  ])(`%s`, (testName, length_mm, width_mm, weight_g, postalRate) => {
    assertForAllUnits(length_mm, width_mm, weight_g, postalRate);
  });
});

```

Fail:

It passed directly without changing any code from the previous test because we have already checked width and weight

Pass:

```
PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (4 ms)
  ✓ ID02: Standard envelope, regular weight (1 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
  ✓ ID05: Non-standard envelope, due to length (1 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (1 ms)
  ✓ ID07: Non-standard envelope, due to width (1 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (1 ms)

Test Suites: 1 passed, 1 total
Tests:      8 passed, 8 total
Snapshots:  0 total
Time:       1.031 s, estimated 2 s
Ran all test suites.
```

ID09

Test:

```
test("ID09: Missing required arguments", () => {
  const req = {
    body: {
      length: 200,
    },
  };

  expected(req, 400);
});
```

Fail:

```

FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (6 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (3 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
  ✓ ID05: Non-standard envelope, due to length (20 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (2 ms)
  ✓ ID07: Non-standard envelope, due to width (2 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (7 ms)
Invalid Domain Logic
  ✗ ID09: Missing required arguments (17 ms)

● Invalid Domain Logic > ID09: Missing required arguments

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

   32 |     if (text === undefined) {
   33 |         // 400 status code
>  34 |         expect(res.code).toBe(status);
      |                             ^
   35 |     } else {
   36 |         expect({ status: res.code, text: res.text }).toEqual({ status, text });
   37 |     }

at expected (ramp.test.js:34:22)
at Object.<anonymous> (ramp.test.js:140:5)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 8 passed, 9 total
Snapshots:  0 total
Time:        2.144 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.

```

Code:

To make this test correct, we need to add an if statement at the beginning of our code.

```

function ramp(req, res) {
  const { length, width, weight, distanceUnit, weightUnit } = req.body;
  if (length == null || width == null || weight == null) {
    return res.status(400);
  }
}

```

Pass:


```

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (4 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (10 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
  ✓ ID05: Non-standard envelope, due to length (2 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (2 ms)
  ✓ ID07: Non-standard envelope, due to width (1 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (2 ms)
Invalid Domain Logic
  ✓ ID09: Missing required arguments (1 ms)

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        0.948 s, estimated 2 s
Ran all test suites.

```

ID10-15

Test:

```

test.each([
  // task name, property, invalid value
  ["ID10: Length too short", "length", 70],
  ["ID11: Length too long", "length", 381],
  ["ID12: Width too short", "width", -2],
  ["ID13: Width too long", "width", 300],
  ["ID14: Weight too light", "weight", 89.9],
  ["ID15: Weight too heavy", "weight", 1000 * 1000],
])( "%s", (taskName, property, invalidValue) => {
  const req = cloneValidReq();
  req.body[property] = invalidValue;
  assertForAllUnits(req.length, req.width, req.weight, undefined);
});

```

Fail:

FAIL ./ramp.test.js

Valid Domain Logic

- ✓ ID01: Standard envelope, lightweight (5 ms)
- ✓ ID02: Standard envelope, regular weight (5 ms)
- ✓ ID03: Non-standard envelope, due to heaviness (4 ms)
- ✓ ID04: Non-standard envelope, due to heaviness, very heavy (4 ms)
- ✓ ID05: Non-standard envelope, due to length (2 ms)
- ✓ ID06: Non-standard envelope, due to length and weight (21 ms)
- ✓ ID07: Non-standard envelope, due to width (2 ms)
- ✓ ID08: Non-standard envelope, due to width and weight (1 ms)

Invalid Domain Logic

- ✓ ID09: Missing required arguments (1 ms)
- ✗ ID10: Length too short (11 ms)
- ✗ ID11: Length too long (1 ms)
- ✗ ID12: Width too short (1 ms)
- ✗ ID13: Width too long (7 ms)
- ✗ ID14: Weight too light (1 ms)
- ✗ ID15: Weight too heavy (23 ms)

● Invalid Domain Logic > ID10: Length too short

```
expect(received).toBe(expected) // Object.is equality
```

Expected: 400

Received: 200

```
32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                       ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }
```

```
at expected (ramp.test.js:34:22)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:154:5
```

● Invalid Domain Logic > ID11: Length too long

```
expect(received).toBe(expected) // Object.is equality
```

Expected: 400

Received: 200

```
32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                       ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }
```

```
at expected (ramp.test.js:34:22)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:154:5
```

● Invalid Domain Logic > ID12: Width too short

```
expect(received).toBe(expected) // Object.is equality
```

Expected: 400

Received: 200

```
32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                       ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }
```

```

32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                        ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }

at expected (ramp.test.js:34:22)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:154:5

```

● Invalid Domain Logic > ID13: Width too long

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

```

32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                        ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }

at expected (ramp.test.js:34:22)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:154:5

```

● Invalid Domain Logic > ID14: Weight too light

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

```

32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                        ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }

at expected (ramp.test.js:34:22)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:154:5

```

● Invalid Domain Logic > ID15: Weight too heavy

expect(received).toBe(expected) // Object.is equality

● Invalid Domain Logic > ID15: Weight too heavy

```
expect(received).toBe(expected) // Object.is equality
```

Expected: 400

Received: 200

```
32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                       ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }
```

```
at expected (ramp.test.js:34:22)
at assertForAllUnits (ramp.test.js:109:7)
at ramp.test.js:154:5
```

Test Suites: 1 failed, 1 total

Tests: 6 failed, 9 passed, 15 total

Snapshots: 0 total

Time: 1.438 s

Ran all test suites.

npm ERR! Test failed. See above for more details.

Code (line 22-43, importantly 43):

```
1 | const { mm_to_in, in_to_mm, g_to_oz, oz_to_g } = require("../helper");
2 | function ramp(req, res) {
3 |   const { length, width, weight, distanceUnit, weightUnit } = req.body;
4 |   if (length == null || width == null || weight == null) {
5 |     return res.status(400);
6 |   }
7 |   var isStandard = false;
8 |   var isNonStandard = false;
9 |   var postalRate = 0;
10 |   var real_weight = weight;
11 |   var real_length = length;
12 |   var real_width = width;
13 |   if (distanceUnit == "in") {
14 |     real_length = in_to_mm(length);
15 |     real_width = in_to_mm(width);
16 |   }
17 |   if (weightUnit == "oz") {
18 |     real_weight = oz_to_g(weight);
19 |   }
20 |
21 |   // check if this is a non-standard envelop
22 |   if (140 <= real_length && real_length <= 245 && 90 <= real_width && real_width <= 156 && 3 <= real_weight && real_weight <= 50) {
23 |     isStandard = true;
24 |   }
25 |   // check if this is a standard envelop
26 |   else if (140 <= real_length && real_length <= 380 && 90 <= real_width && real_width <= 270 && 3 <= real_weight && real_weight <= 500) {
27 |     isNonStandard = true;
28 |   }
29 | }
```

```

29 | //calculate postalRate
30 | if (isStandard) {
31 |   if (real_weight > 30) {
32 |     postalRate = 0.8;
33 |   } else {
34 |     postalRate = 0.49;
35 |   }
36 | } else if (isNonStandard) {
37 |   if (real_weight <= 100) {
38 |     postalRate = 0.98;
39 |   } else {
40 |     postalRate = 2.4;
41 |   }
42 | } else {
43 |   return res.status(400);
44 | }
45 | res.send(`${postalRate}`);
46 | }
47 |
48 | module.exports = ramp;
49 |

```

Pass:

```

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (4 ms)
  ✓ ID02: Standard envelope, regular weight (3 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (6 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (4 ms)
  ✓ ID05: Non-standard envelope, due to length (3 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (8 ms)
  ✓ ID07: Non-standard envelope, due to width (22 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (2 ms)
Invalid Domain Logic
  ✓ ID09: Missing required arguments (19 ms)
  ✓ ID10: Length too short (9 ms)
  ✓ ID11: Length too long (5 ms)
  ✓ ID12: Width too short (1 ms)
  ✓ ID13: Width too long (6 ms)
  ✓ ID14: Weight too light (1 ms)
  ✓ ID15: Weight too heavy (2 ms)

Test Suites: 1 passed, 1 total
Tests:       15 passed, 15 total
Snapshots:   0 total
Time:        1.181 s, estimated 2 s
Ran all test suites.

```

ID16-17

Test:

```

test.each([
  // task name, property, invalid value
  ["ID16: Distance unit not in mm or in", "distanceUnit", "cm"],
  ["ID17: Weight unit not in g or oz", "weightUnit", "qirjgpoiqrjgpoiqrjg"],
])(`${taskName}, ${property}, invalid value => {
  const req = cloneValidReq();
  req.body[property] = invalidValue;

  expected(req, 400);
});

```

Fail:

```

FAIL ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (5 ms)
  ✓ ID02: Standard envelope, regular weight (9 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (2 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (3 ms)
  ✓ ID05: Non-standard envelope, due to length (2 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (2 ms)
  ✓ ID07: Non-standard envelope, due to width (2 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (1 ms)
Invalid Domain Logic
  ✓ ID09: Missing required arguments
  ✓ ID10: Length too short (1 ms)
  ✓ ID11: Length too long (1 ms)
  ✓ ID12: Width too short (1 ms)
  ✓ ID13: Width too long (3 ms)
  ✓ ID14: Weight too light (2 ms)
  ✓ ID15: Weight too heavy (1 ms)
  ✗ ID16: Distance unit not in mm or in (10 ms)
  ✗ ID17: Weight unit not in g or oz (1 ms)

● Invalid Domain Logic › ID16: Distance unit not in mm or in

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

   32 |     if (text === undefined) {
   33 |       // 400 status code
>  34 |       expect(res.code).toBe(status);
      |                           ^
   35 |     } else {
   36 |       expect({ status: res.code, text: res.text }).toEqual({ status, text });
   37 |     }

```

```

    at expected (ramp.test.js:34:22)
    at ramp.test.js:166:5

● Invalid Domain Logic › ID17: Weight unit not in g or oz

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

   32 |     if (text === undefined) {
   33 |       // 400 status code
>  34 |       expect(res.code).toBe(status);
      |                           ^
   35 |     } else {
   36 |       expect({ status: res.code, text: res.text }).toEqual({ status, text });
   37 |     }

    at expected (ramp.test.js:34:22)
    at ramp.test.js:166:5

Test Suites: 1 failed, 1 total
Tests:      2 failed, 15 passed, 17 total
Snapshots:  0 total
Time:       1.279 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.

```

Code:

To test the validity of units, we only need to add an if statement near the beginning of the code.

```
function ramp(req, res) {
  const { length, width, weight, distanceUnit, weightUnit } = req.body;
  if (length == null || width == null || weight == null) {
    return res.status(400);
  }
  if ((distanceUnit != "mm" && distanceUnit != "in" && distanceUnit !== undefined)
    || (weightUnit != "g" && weightUnit != "oz" && weightUnit !== undefined)) {
    return res.status(400);
  }
  var isStandard = false;
}
```

Pass:

```
PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (5 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (16 ms)
  ✓ ID05: Non-standard envelope, due to length (9 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (1 ms)
  ✓ ID07: Non-standard envelope, due to width (21 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (3 ms)
Invalid Domain Logic
  ✓ ID09: Missing required arguments (32 ms)
  ✓ ID10: Length too short (2 ms)
  ✓ ID11: Length too long (2 ms)
  ✓ ID12: Width too short (24 ms)
  ✓ ID13: Width too long (23 ms)
  ✓ ID14: Weight too light (15 ms)
  ✓ ID15: Weight too heavy (1 ms)
  ✓ ID16: Distance unit not in mm or in (5 ms)
  ✓ ID17: Weight unit not in g or oz

Test Suites: 1 passed, 1 total
Tests: 17 passed, 17 total
Snapshots: 0 total
Time: 0.925 s, estimated 1 s
Ran all test suites.
```

ID18

Test:

```
test("ID18: Invalid request payload syntax", () => {
  const req = {
    body: '{"wrth"lqrthength": 200,rhqrtehqrth"qwrthidth": 156,qwretgqergqet"weight":qwrth 10qrth}',
  };

  expected(req, 400);
});
```

Fail:

This test passed directly without changing any code from previous test. Due to how Javascript destructuring works, the variables for req.body are just set to undefined. And then it's interpreted as too few arguments and sends a 400 regardless. Also in Express.js, we put a middleware given to us by the framework that checks for a valid JSON payload before even passing it to the ramp function: `app.use(express.json());`

Pass:


```

PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (8 ms)
  ✓ ID02: Standard envelope, regular weight (3 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (2 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
  ✓ ID05: Non-standard envelope, due to length (20 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (1 ms)
  ✓ ID07: Non-standard envelope, due to width (1 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (1 ms)
Invalid Domain Logic
  ✓ ID09: Missing required arguments
  ✓ ID10: Length too short (2 ms)
  ✓ ID11: Length too long (1 ms)
  ✓ ID12: Width too short (5 ms)
  ✓ ID13: Width too long (1 ms)
  ✓ ID14: Weight too light
  ✓ ID15: Weight too heavy (1 ms)
  ✓ ID16: Distance unit not in mm or in
  ✓ ID17: Weight unit not in g or oz
  ✓ ID18: Invalid request payload syntax (2 ms)

Test Suites: 1 passed, 1 total
Tests: 18 passed, 18 total
Snapshots: 0 total
Time: 0.799 s, estimated 1 s
Ran all test suites.

```

ID19-21

Test:

```

test.each([
  // task name, property, invalid value
  ["ID19: Length is not a number", "length", "200"],
  ["ID20: Width is not a number", "width", "200"],
  ["ID21: Weight is not a number", "weight", "200"],
])("%s", (taskName, property, invalidValue) => {
  const req = cloneValidReq();
  req.body[property] = invalidValue;

  expected(req, 400);
});

```

Fail:

FAIL ./ramp.test.js

Valid Domain Logic

- ✓ ID01: Standard envelope, lightweight (9 ms)
- ✓ ID02: Standard envelope, regular weight (2 ms)
- ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
- ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
- ✓ ID05: Non-standard envelope, due to length (1 ms)
- ✓ ID06: Non-standard envelope, due to length and weight (3 ms)
- ✓ ID07: Non-standard envelope, due to width (1 ms)
- ✓ ID08: Non-standard envelope, due to width and weight (1 ms)

Invalid Domain Logic

- ✓ ID09: Missing required arguments (1 ms)
- ✓ ID10: Length too short (1 ms)
- ✓ ID11: Length too long (1 ms)
- ✓ ID12: Width too short
- ✓ ID13: Width too long (1 ms)
- ✓ ID14: Weight too light (2 ms)
- ✓ ID15: Weight too heavy
- ✓ ID16: Distance unit not in mm or in
- ✓ ID17: Weight unit not in g or oz (1 ms)
- ✓ ID18: Invalid request payload syntax
- ✗ ID19: Length is not a number (6 ms)
- ✗ ID20: Width is not a number (1 ms)
- ✗ ID21: Weight is not a number (1 ms)

● Invalid Domain Logic > ID19: Length is not a number

expect(received).toBe(expected) // Object.is equality

Expected: 400

Received: 200

```
32 |   if (text === undefined) {
33 |     // 400 status code
> 34 |     expect(res.code).toBe(status);
    |                       ^
35 |   } else {
36 |     expect({ status: res.code, text: res.text }).toEqual({ status, text });
37 |   }
```

at expected (ramp.test.js:34:22)

at ramp.test.js:188:5

● Invalid Domain Logic > ID20: Width is not a number

```

● Invalid Domain Logic > ID20: Width is not a number

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

   32 |     if (text === undefined) {
   33 |       // 400 status code
>  34 |       expect(res.code).toBe(status);
      |                               ^
   35 |     } else {
   36 |       expect({ status: res.code, text: res.text }).toEqual({ status, text });
   37 |     }

    at expected (ramp.test.js:34:22)
    at ramp.test.js:188:5

● Invalid Domain Logic > ID21: Weight is not a number

expect(received).toBe(expected) // Object.is equality

Expected: 400
Received: 200

   32 |     if (text === undefined) {
   33 |       // 400 status code
>  34 |       expect(res.code).toBe(status);
      |                               ^
   35 |     } else {
   36 |       expect({ status: res.code, text: res.text }).toEqual({ status, text });
   37 |     }

    at expected (ramp.test.js:34:22)
    at ramp.test.js:188:5

Test Suites: 1 failed, 1 total
Tests:       3 failed, 18 passed, 21 total
Snapshots:   0 total
Time:        2.122 s
Ran all test suites.
npm ERR! Test failed.  See above for more details.

```

Code:

To test the type of length, width and weight, we only need to add an if statement near the beginning of the code.

```

2   function ramp(req, res) {
3     const { length, width, weight, distanceUnit, weightUnit } = req.body;
4     if (length == null || width == null || weight == null) {
5       return res.status(400);
6     }
7     if (typeof length != "number" || typeof weight != "number" || typeof width != "number") {
8       return res.status(400);
9     }
10    if ((distanceUnit != "mm" && distanceUnit != "in" && distanceUnit !== undefined) || (weightUn
11      return res.status(400);
12    }

```

Pass:

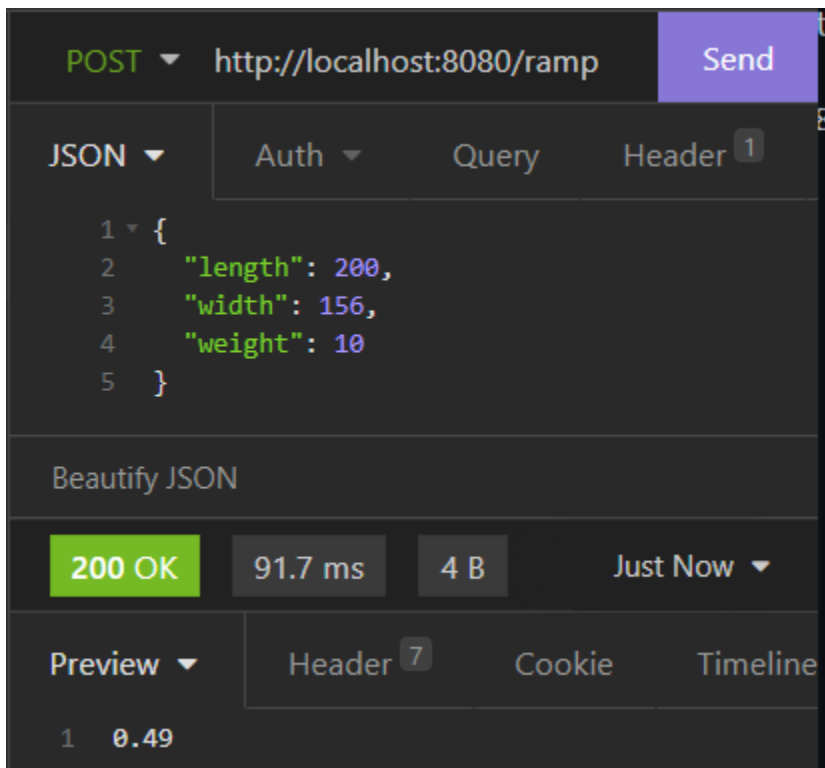
```
PASS ./ramp.test.js
Valid Domain Logic
  ✓ ID01: Standard envelope, lightweight (9 ms)
  ✓ ID02: Standard envelope, regular weight (2 ms)
  ✓ ID03: Non-standard envelope, due to heaviness (17 ms)
  ✓ ID04: Non-standard envelope, due to heaviness, very heavy (15 ms)
  ✓ ID05: Non-standard envelope, due to length (4 ms)
  ✓ ID06: Non-standard envelope, due to length and weight (14 ms)
  ✓ ID07: Non-standard envelope, due to width (26 ms)
  ✓ ID08: Non-standard envelope, due to width and weight (5 ms)
Invalid Domain Logic
  ✓ ID09: Missing required arguments
  ✓ ID10: Length too short (2 ms)
  ✓ ID11: Length too long (3 ms)
  ✓ ID12: Width too short (1 ms)
  ✓ ID13: Width too long (2 ms)
  ✓ ID14: Weight too light (25 ms)
  ✓ ID15: Weight too heavy (8 ms)
  ✓ ID16: Distance unit not in mm or in (1 ms)
  ✓ ID17: Weight unit not in g or oz (1 ms)
  ✓ ID18: Invalid request payload syntax
  ✓ ID19: Length is not a number
  ✓ ID20: Width is not a number
  ✓ ID21: Weight is not a number

Test Suites: 1 passed, 1 total
Tests: 21 passed, 21 total
Snapshots: 0 total
Time: 1.122 s, estimated 2 s
Ran all test suites.
```

Example API Requests

Note: used Insomnia

Standard envelope:



Non-standard envelope:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8080/ramp
- Body:** A JSON object with three properties: "length": 200, "width": 260, and "weight": 10. The JSON is formatted with line numbers 1 through 5.
- Response:** 200 OK, 2.62 ms, 4 B, Just Now.
- Preview:** 0.98

Bad request:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8080/ramp
- Body:** A JSON object with three properties: "length": 200, "width": 260, and "weight": -1. The JSON is formatted with line numbers 1 through 5.
- Response:** 400 Bad Request, 2.99 ms, Just Now.
- Preview:** Bad Request

Note: I had to change all the `return res.status(400);` to `res.sendStatus(400);`, because if we don't send anything, the API will hang. The tests couldn't foresee this issue because we were using mocks. Just to be sure, all the tests still pass afterwards:

PASS `./ramp.test.js`

Valid Domain Logic

- ✓ ID01: Standard envelope, lightweight (5 ms)
- ✓ ID02: Standard envelope, regular weight (1 ms)
- ✓ ID03: Non-standard envelope, due to heaviness (1 ms)
- ✓ ID04: Non-standard envelope, due to heaviness, very heavy (2 ms)
- ✓ ID05: Non-standard envelope, due to length (2 ms)
- ✓ ID06: Non-standard envelope, due to length and weight (3 ms)
- ✓ ID07: Non-standard envelope, due to width (1 ms)
- ✓ ID08: Non-standard envelope, due to width and weight (1 ms)

Invalid Domain Logic

- ✓ ID09: Missing required arguments
- ✓ ID10: Length too short (1 ms)
- ✓ ID11: Length too long
- ✓ ID12: Width too short (1 ms)
- ✓ ID13: Width too long (1 ms)
- ✓ ID14: Weight too light
- ✓ ID15: Weight too heavy
- ✓ ID16: Distance unit not in mm or in
- ✓ ID17: Weight unit not in g or oz
- ✓ ID18: Invalid request payload syntax
- ✓ ID19: Length is not a number (1 ms)
- ✓ ID20: Width is not a number (1 ms)
- ✓ ID21: Weight is not a number (1 ms)

Test Suites: 1 passed, 1 total

Tests: 21 passed, 21 total

Snapshots: 0 total

Time: 2.083 s

Ran all test suites.