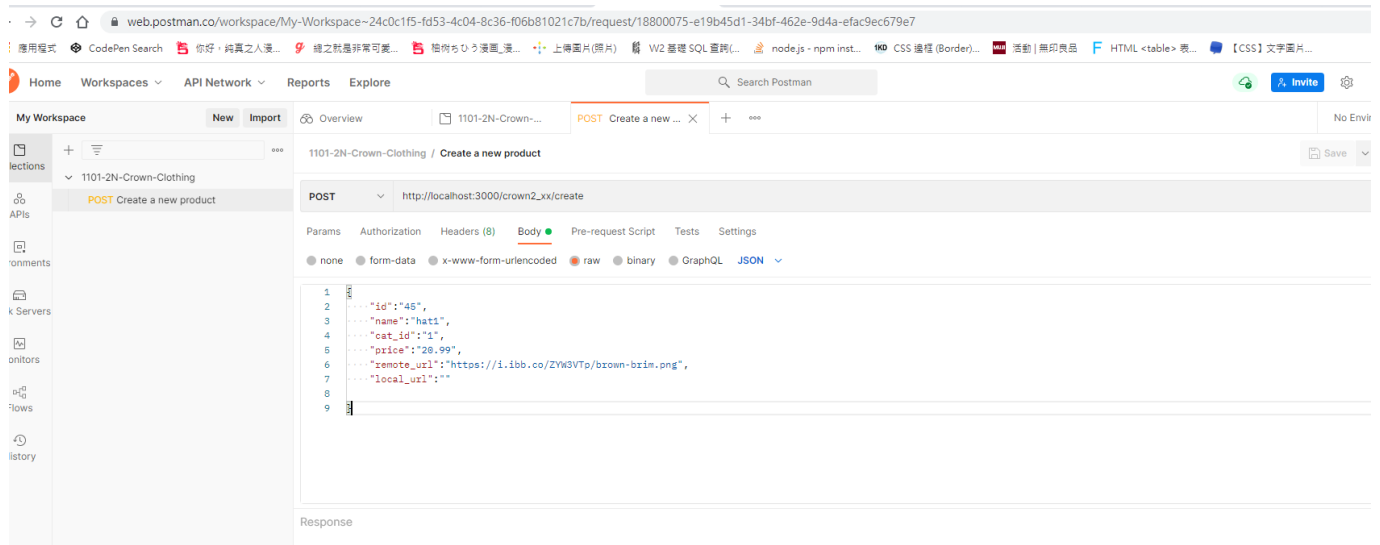


P1: Create json data of a product from Postman, and send it by POST request via route /crown2_xx/create

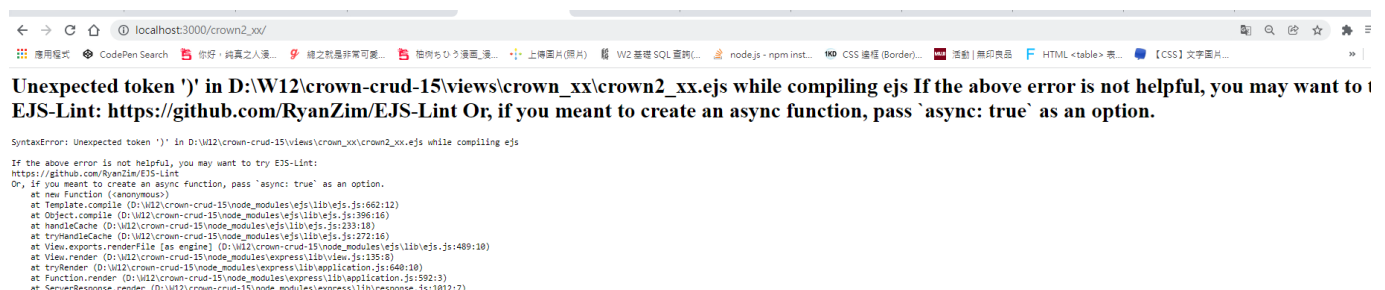
=> Create a new POST request from Postman with a new json data of product



=> code for handling route /crown2_xx/create with console output



=> 不知道ejs出了什麼問題 localhost跑不出來 postman也無法顯示



P2: Create code for creating a product, and verify

=> crown2Controller code

```
2
3  /* CREATE */
4
5  exports.createProducts = async (req, res) => {
6    console.log('body', req.body);
7    res.json('Create body received')
8    try{
9      let results = await serviceCrown2Controller_xx.create(req.body);
10     console.log('results', JSON.stringify(results));
11     res.json('data:{msg: creating successful}');
12   }catch(err){
13
14   }
15 }
16
17 /* READ */
```

=> serviceController code

```
JS crown2Controller_xx.js M  w12.md U  JS serviceCrown2Controller_xx.js M X
controllers > crown_xx > JS serviceCrown2Controller_xx.js > ...
1  const Category_xx = require('../../models/crown_xx/Category_xx');
2  const Shop_xx = require('../../models/crown_xx/Shop_xx');
3
4  //CREATE
5  exports.create = async(body) => {
6    try {
7      return await Shop_xx.create(body);
8    } catch(err){
9      console.log('create',err);
10   }
11 }
12
13
14 //READ
15
```

=> model Shop_xx code

```

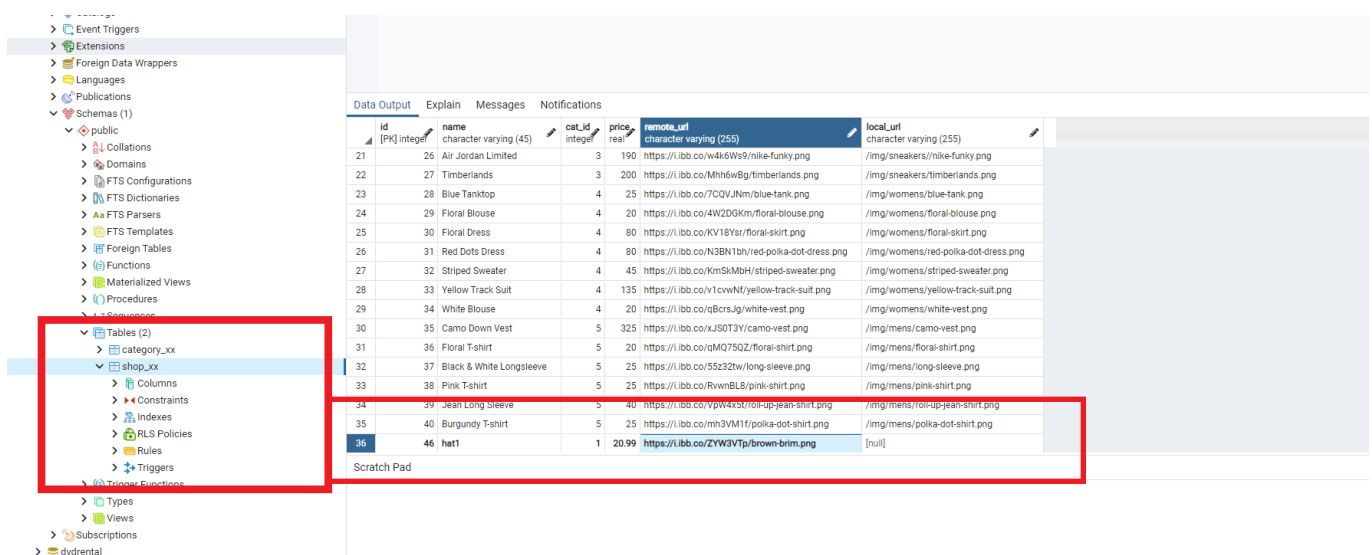
JS crown2Controller_xx.js M X w12.md U JS Shop_xx.js M X JS serviceCrown2Controller_xx.js M
models > crown_xx > JS Shop_xx.js > ...
1  const db = require('../../utils/database');
2
3  const Shop_xx = class Shop_xx {
4    constructor(id, name, cat_id, price, remote_url, local_url) {
5      this.id = id;
6      this.name = name;
7      this.cat_id = cat_id;
8      this.price = price;
9      this.remote_url = remote_url;
10     this.local_url = local_url;
11   }
12
13
14   //CREATE
15   static async create(body) {
16     const {id,name,cat_id,price,remote_url,local_url} =body;
17     const query = {
18       text: `INSERT INTO shop_xx(id,name,cat_id,price,remote_url,local_url)
19         VALUES ($1,$2,$3,$4,$5,$6)`,
20       values:[id,name,cat_id,price,remote_url,local_url]
21     }
22   }
23   return db.query(query);
24 }
25
26 // get all categories

```

=> Postman request, json data and result

!p2-4.png]((https://i.imgur.com/jwtSlBm.png))

=> Verify by pgAdmin



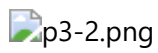
id	name	cat_id	price	remote_url	local_url
21	Air Jordan Limited	3	190	https://i.libb.co/w4k6W69/nike-funky.png	/img/sneakers/nike-funky.png
22	Timberlands	3	200	https://i.libb.co/Mth6wBg/timberlands.png	/img/sneakers/timberlands.png
23	Blue Tanktop	4	25	https://i.libb.co/7CQVJNm/blue-tank.png	/img/womens/blue-tank.png
24	Floral Blouse	4	20	https://i.libb.co/4W2DGKm/floral-blouse.png	/img/womens/floral-blouse.png
25	Floral Dress	4	80	https://i.libb.co/KV18Ysr/floral-skirt.png	/img/womens/floral-skirt.png
26	Red Dots Dress	4	80	https://i.libb.co/N3BN1bn/red-polka-dot-dress.png	/img/womens/red-polka-dot-dress.png
27	Striped Sweater	4	45	https://i.libb.co/KmSkMbH/striped-sweater.png	/img/womens/striped-sweater.png
28	Yellow Track Suit	4	135	https://i.libb.co/v1cwnf/yellow-track-suit.png	/img/womens/yellow-track-suit.png
29	White Blouse	4	20	https://i.libb.co/qBcrsJg/white-vest.png	/img/womens/white-vest.png
30	Camo Down Vest	5	325	https://i.libb.co/xJ50T3V/camo-vest.png	/img/mens/camo-vest.png
31	Floral T-shirt	5	20	https://i.libb.co/qMQ75QZ/floral-shirt.png	/img/mens/floral-shirt.png
32	Black & White Longsleeve	5	25	https://i.libb.co/5S23Zhw/long-sleeve.png	/img/mens/long-sleeve.png
33	Pink T-shirt	5	25	https://i.libb.co/RvwnBL8/pink-shirt.png	/img/mens/pink-shirt.png
34	Jean Long Sleeve	5	40	https://i.libb.co/vPW4t3t/roll-up-jean-shirt.png	/img/mens/roll-up-jean-shirt.png
35	Burgundy T-shirt	5	25	https://i.libb.co/mh3VM1f/polka-dot-shirt.png	/img/mens/polka-dot-shirt.png
36	hat1	1	20.99	https://i.libb.co/ZYW3Vtp/brown-brim.png	[null]

P3: Code for updating a product, and verify

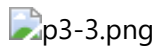
=> crown2Controller code

p3-1.png

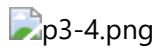
=> **serviceController code**



=> **model Shop_xx code**



=> **Postman request, json data and result**



=> **Verify by pgAdmin**

