

## Setup

```
$ npm install express-generator -g  
$ npm install nodemon
```

## Generator app

```
$ mkdir (folder name)  
$ cd (folder name)  
$ express --view=ejs crown2_12(app name)  
$ code crown2_12 (app name)
```

## Install package

```
$ npm i
```

## Start app

```
npm start
```

## Modify to nodemon

1. open package.json
2. modify

```
"scripts": {  
  "start": "node ./bin/www"  
}  
> to  
"scripts": {  
  "start": "nodemon ./bin/www"  
}
```

## Modify public

1. open folder - public
2. delete all
3. add crown\_themes

<div> <span>←</span> <span>→</span> <span>⌵</span> <span>⬆</span> <span>📁</span> &gt; 1102DB_HW &gt; crown2_12 &gt; public         </div>				
名稱	修改日期	類型	大小	
📁 assets	2022/4/6 下午 06:58	檔案資料夾		
📁 css	2022/4/6 下午 06:58	檔案資料夾		
📁 img	2022/4/6 下午 06:58	檔案資料夾		
🌐 hats	2022/4/6 下午 06:58	Chrome HTML D...	6 KB	
🌐 index	2022/4/6 下午 06:58	Chrome HTML D...	4 KB	
🌐 jackets	2022/4/6 下午 06:58	Chrome HTML D...	4 KB	
🌐 mens	2022/4/6 下午 06:58	Chrome HTML D...	4 KB	
🌐 overview	2022/4/6 下午 06:58	Chrome HTML D...	10 KB	
🌐 sneakers	2022/4/6 下午 06:58	Chrome HTML D...	5 KB	
🌐 womens	2022/4/6 下午 06:58	Chrome HTML D...	5 KB	

## Modify views

1. add a folder crown2\_12
2. copy index.html to this folder
3. change the file type by renaming the file to index.ejs

## Create Router

// change the ui

1. open app.js
2. then adding

```

app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/crown2_12', crown2_12_Router); // this line
// and
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

const crown2_12_Router = require('./routes/crown2_12') // this line

var app = express();

```

3. then adding file crown2\_12.js
4. copy the code of file ./index.js
5. then change the code to

```
router.get('/', function(req, res, next) {
  res.render('crown2_12/index', { id: '409650412', title: 'crown2' });
});
```



```
routes > crown2_12.js > ...
1  var express = require('express');
2  var router = express.Router();
3
4  GET home page. */
5  router.get('/', function(req, res, next) {
6    res.render('index', { title: 'Express-2' });
7  });
8
9  module.exports = router;
10
```

## Add name and id

1. go to the /views/crown2\_12/index.ejs
2. add this

```
<div class="homepage">
  <div> // this line
    <h3> <%= title %> <%= id %> </h3> // this line
  </div> // this line
  <div class="directory-menu">
```

## Heroku

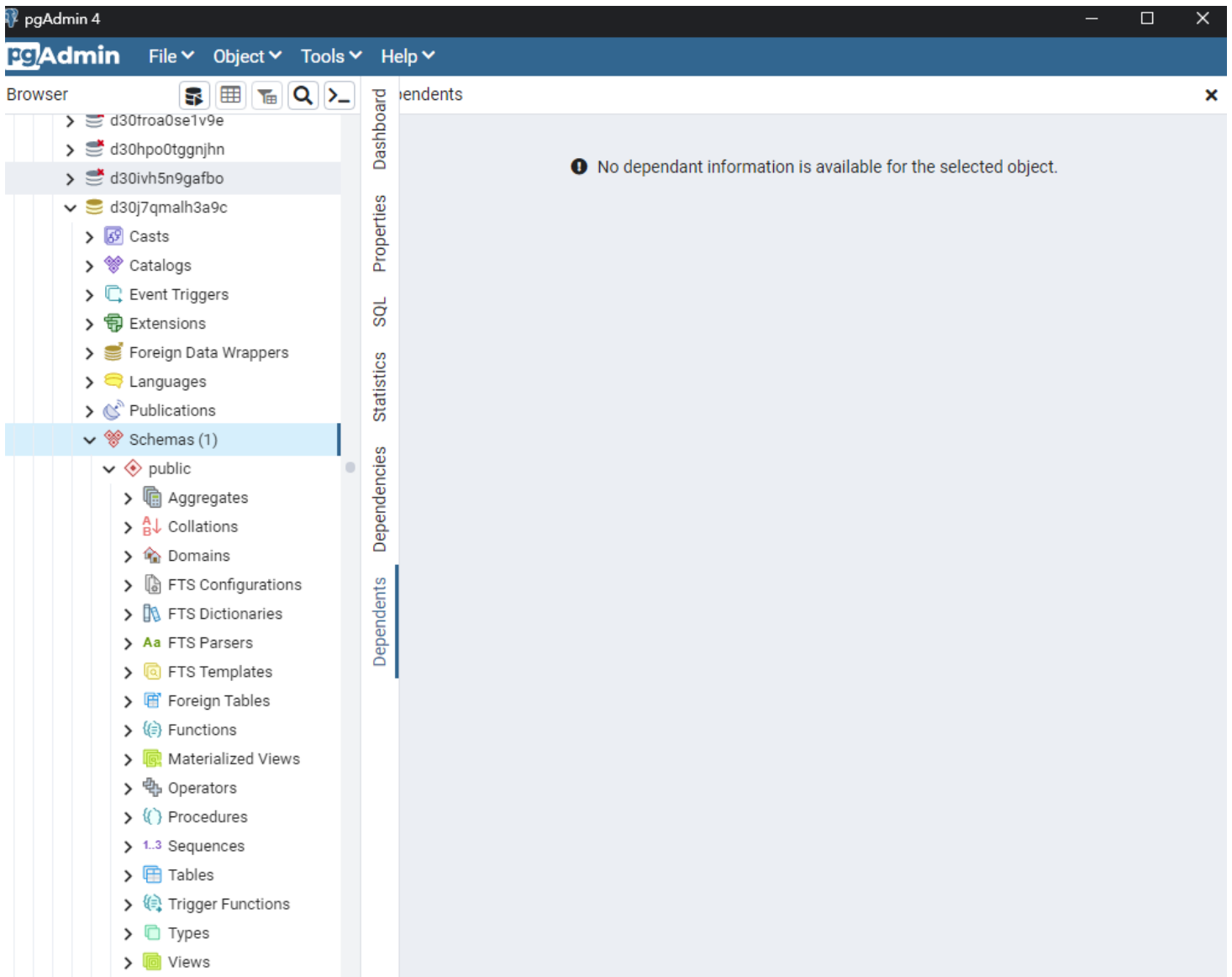
1. login
2. create new app > name: crown2-1102db-hw
3. => Resources page => Add-ons
4. type "Herok Postgres" at the quick add
5. => Setting => Config Vars => Reveal Config Vars
6. copy

```
postgres://
rsiuqaahnockfn: // user.name
3e81cac92db030ea18463bbd38c5c93c1febb6aa1bcbc519c0f51d304e91b950@ // password
ec2-52-21-136-176.compute-1.amazonaws.com: // hostname
5432/ // port
d30j7qmalh3a9c // maintenance
```

pgAdmin

pgAdmin	Heroku
Host name	ec2-52-21-136-176.compute-1.amazonaws.com
Port	5432
Maintenance	d30j7qmalh3a9c
Password	3e81cac92db030ea18463bbd38c5c93c1febb6aa1bcbc519c0f51d304e91b950

Search Server





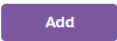
## Add Node\_Env

### Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

### Config Vars

Hide Config Vars

DATABASE_URL	postgres://rsiuqaahnockfn:3e81cac92db030e	 
NODE_ENV	production	

## Connect Heroku to Github

1. => Deploy page
2. search repo
3. connect repo

Personal > crown2-1102db-hw ☆ Open app More

Overview Resources **Deploy** Metrics Activity Access Settings

**Add this app to a pipeline**

Create a new pipeline or choose an existing one and add this app to a stage in it.

**Add this app to a stage in a pipeline to enable additional features**

Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

Choose a pipeline

**Deployment method**

Heroku Git  
Use Heroku CLI

GitHub  
Connect to GitHub

Container Registry  
Use Heroku CLI

**Connect to GitHub**

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

chanyu020816 1102DB\_HW Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

## 4. enable auto deploy

**Automatic deploys**

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Automatic deploys from `main` are enabled

Every push to `main` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. [Learn more.](#)

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Disable Automatic Deploys

## 5. press deploy at "Manual Deploy"

// if error then

```
$ npm install heroku
```

## Modify script

- => package.json
- modify this

```
"scripts": {
  "start": "nodemon ./bin/www",
  "dev": "nodemon ./bin/www" // add this line
},
```

3. then open app on Heroku

## Open app

## Open pgAdmin

1. create table

The screenshot shows the pgAdmin 4 interface. On the left, the 'Databases (4)' tree is expanded, showing 'crown2' > 'Schemas (1)' > 'public' > 'Tables (1)' > 'category\_12'. The main panel displays the 'Query Editor' with the following SQL query:

```
1 SELECT * FROM public.category_12
2 ORDER BY id ASC
```

The 'Data Output' tab shows the results of the query, which are 5 rows of data from the 'category\_12' table:

	id [PK] integer	name character varying	size character varying	remote_url character varying	local_url character varying	link_url character varying
1	1	hats	[null]	https://i.libb.co/cvpntL1/hats.png	/img/homepage/hats.png	/crown2_12/shop_12/hats
2	2	jackets	[null]	https://i.libb.co/px2tCc3/jackets.png	/img/homepage/jackets.png	/crown2_12/shop_12/jackets
3	3	sneakers	[null]	https://i.libb.co/0jqHnp/sneakers.png	/img/homepage/sneakers.png	/crown2_12/shop_12/sneakers
4	4	womens	large	https://i.libb.co/GCCdy8t/womens.png	/img/homepage/womens.png	/crown2_12/shop_12/womens
5	5	mens	large	https://i.libb.co/R70vBrQ/men.png	/img/homepage/mens.png	/crown2_12/shop_12/mens

**\*\*Connect with database**

1. => 1102DB\_HW/CROWN2\_12 => adding folder utilis
2. adding database.js
3. install package

```
$ npm install pg
```

4. => utilis/database.js

```
const { Pool } = require('pg');
const { connectionString } = require('pg/lib/defaults');

const isProduction = process.env.NODE_ENV === 'production';

let pool;

if(isProduction) {
  pool = new Pool (
    {
      connectionString: process.env.DATABASE_URL,
      ssl: { rejectUnauthorized: false }
    }
  )
} else {
  pool = new Pool ({
    user: 'postgres',
    host: 'localhost',
    port: '5432',
    database: 'crown2',
    password: '0000'
  });
}

module.exports = pool;
```

## pgAdmin

1. => pgAdmin find "d30j7qmalh3a9c"
2. Restore crown.tar

## Connect to database

1. => Create new folder models
2. create "Category\_12.js"
3. Typing



```

const db = require('../utilis/database');

const Category_12 = class Category_12 {

  constructor(id, name, size, remote_url, local_url, link_url) {
    this.id = id;
    this.name = name;
    this.remote_url = remote_url;
    this.local_url = local_url;
    this.link_url = link_url;
  }

  static async fetchAll() {
    try {
      let results = await db.query(`SELECT * from category_12`);
      // console.log('results', JSON.stringify(results.rows));
      return results.rows;
    } catch (err) {
      console.log('error', err);
    }
  }
}

//testing
//const test = async () => {
//  let results = await Category_12.fetchAll();
//  console.log('test result', JSON.stringify(results))
//}
//test();

module.exports = Category_12.js;

```

4. => /routes/crown2\_12.js

5. modify the code

```

var express = require('express');
const Category_12 = require('../models/Category_12'); //this line
var router = express.Router();

/* GET home page. */
router.get('/', async function(req, res, next) {
  try {
    let results = await Category_12.fetchAll(); // this line
    console.log('results', JSON.stringify(results)); // this line
    res.render('crown2_12/index', { // this line
      data: results, // this line
      id: '409650412', // this line
      title: 'crown2' }); // this line
  } catch(err){ // this line
    console.log('error', err) // this line
  }

});

module.exports = router;

```

## for loop

1. => /views/crown2\_12/index.ejs
2. modify menu-items

```

41 </div>
42
43 <div class="homepage">
44   <div>
45     <h3> <%= title %> <%= id %> </h3>
46   </div>
47   <div class="directory-menu">
48 >     <div class="menu-item">...
54   </div>
55 >     <div class="menu-item">...
61   </div>
62 >     <div class="menu-item">...
68   </div>
69 >     <div class="large menu-item">...
75   </div> You, 4 hours ago • 3/01 ...
76
77   <% for(let i=0; i<data.length; i++){ %>
78     <div class="large menu-item">
79       
80       <a href="./mens.html" class="content">
81         <h1 class="title">MENS</h1>
82         <span class="subtitle">SHOP NOW</span>
83       </a>
84     </div>
85   <% } %>
86
87

```

3. delete all menu item
4. add this

```

<div class="homepage">
  <div>
    <h3> <%= title %> <%= id %> </h3>
  </div>
  <div class="directory-menu">

    <% for(let i=0; i<data.length; i++) { %> // this line
      <div class="<%= data[i].size %> menu-item"> // this line
         // this line
        <a href="<%= data[i].link_url %>" class="content"> // this line
          <h1 class="title"><%= data[i].name %></h1> // this line
          <span class="subtitle">SHOP NOW</span> // this line
        </a> // this line
      </div> // this line
    <% } %> // this line

  </div>
</div>

```

## Adding new page jackests

1. => pgAdmin
2. create new table

The screenshot shows the pgAdmin interface. On the left, the 'Tables (2)' folder is expanded, showing 'category\_12' and 'jacket\_12'. The 'jacket\_12' table is selected. On the right, the 'Data Output' tab is active, displaying a table with 5 rows of data.

id	name	local_url	prize
1	Black Jean Shearling	https://i.ibb.co/Xzcx...	125
2	Blue Jean Jacket	https://i.ibb.co/mJS...	90
3	Grey Jean Jacket	https://i.ibb.co/N71k...	90
4	Brown Shearling	https://i.ibb.co/s96F...	165
5	Tan Trench	https://i.ibb.co/M6h...	185

3. => models
4. create 'shop\_12.js'
5. copy the code from 'Category\_12'
6. modify the code

```
const db = require('../utilis/database');

const shop_12 = class shop_12 { // this line

  constructor(id, name, local_url, prize) { // this line
    this.id = id; // this line
    this.name = name; // this line
    this.local_url = local_url; // this line
    this.prize = prize; // this line
  }

  static async fetchAll() {
    try {
      let results = await db.query(`SELECT * from jacket_12 ORDER BY id`); // this line
      // console.log('results', JSON.stringify(results.rows));
      return results.rows;
    } catch (err) {
      console.log('error', err);
    }
  }
}

// testing
//const test = async () => {
//  let results = await shop_12.fetchAll(); // this line
//  console.log('test result', JSON.stringify(results))
//}
//test();

module.exports = shop_12;
```

7. => routes
8. create 'jackets.js'
9. copy the code from 'crown2\_12.js'
10. modify the code

```

var express = require('express');
const shop_12 = require('../models/shop_12'); // this line
var router = express.Router();

/* GET home page. */
router.get('/', async function(req, res, next) {
  try {
    let results = await shop_12.fetchAll(); // this line
    console.log('results', JSON.stringify(results));
    res.render('crown2_12/jackets', { // this line
      data: results});
  } catch(err){
    console.log('error', err)
  }

  });

module.exports = router;

```

11. => app.js
12. add routes

```

const crown2_12_Router = require('./routes/crown2_12')
const jackets_Router = require('./routes/jackets') // this line

app.use('/crown2_12', crown2_12_Router);
app.use('/crown2_12/shop_12/jackets', jackets_Router) // this line

```

13. copy 'jackets.html' to views/crown2\_12
14. modify to 'jackets.ejs'
15. modify to for loop

```
<div class="shop-page">
  <div class="collection-page">
    <h1 class="title">Jackets</h1>
    <div class="items">
      <% for(let i=0; i<data.length; i++) { %>
        <div class="collection-item">
          
          <div class="collection-footer">
            <span class="name"><%= data[i].name %></span>
            <span class="price"><%= data[i].prize %></span>
          </div>
          <button class="custom-button">Add to Cart</button>
        </div>
      <% } %>
    </div>
  </div>
</div>
```

```

    
  </div>
  <div class="cart-dropdown">
    <div class="cart-items">
      <div class="cart-item">
        
        <div class="item-details">
          <span class="name">Brown Cowboy</span>
          <span class="price">4 * $35</span>
        </div>
      </div>
      <div class="cart-item">

    </div>
  </div>
  <button class="custom-button">GO TO CHECKOUT</button>
</div>

```

⏪ /div> You, 2 hours ago • tired ...

</div>

```

<div class="shop-page">
  <div class="collection-page">
    <h1 class="title">Jackets</h1>
    <div class="items">
      <% for(let i=0; i<data.length; i++) { %>
        <div class="collection-item">
          
          <div class="collection-footer">
            <span class="name"><%= data[i].name %></span>
            <span class="price"><%= data[i].prize %></span>
          </div>
          <button class="custom-button">Add to Cart</button>
        </div>
      <% } %>
    </div>
  </div>
</div>

```

```

<script>
  function cartToggle(){
    p = document.querySelector('.cart-dropdown');
    p.classList.toggle("show");
  }
</script>

```

</body>



Jackets



Black Jean Shearling125



Blue Jean Jacket90



Grey Jean Jacket90



Brown Shearling165



Tan Trench185