

## Lab : Deploy on Clouds

01418471 Introduction to Software Engineering

---

เราจะมาเรียนรู้การ deploy application บน cloud โดยเราจะเลือกใช้ Heroku เนื่องจากมีบัญชีแบบฟรีให้เราลองใช้ database โดยไม่ต้องใส่ข้อมูลบัตรเครดิต

### Heroku

<https://www.heroku.com>

Heroku เป็นผู้ให้บริการคลาวด์แบบ platform-as-a-service เจ้าแรก ๆ ที่ให้ developer สามารถ deploy application ขึ้นไปทำงานบนคลาวด์ได้ รองรับภาษาโปรแกรมที่หลากหลาย เช่น Java, Node.js, PHP, Python, etc.. สามารถอ่าน document ในการพัฒนามบน Heroku ได้ที่ลิงก์นี้

<https://devcenter.heroku.com/>

### I. ทำความเข้าใจกับ application ที่เราจะ deploy

1. เราจะมาเรียนรู้การ deploy application โดยใช้ Spring Boot application 2 แอป
  - o Wisdom Book Web
    - <https://github.com/ladyusa/wisdom-book-front>
  - o Wisdom Book API
    - <https://github.com/ladyusa/wisdom-book-api>
2. ให้นิสิต **fork** และ **clone** จาก repo ทั้งสองด้านบน
  - o จากนั้น เปิด IntelliJ เพื่อทำความเข้าใจโค้ด
3. ลองไปที่ pom.xml จะเห็นว่า เราใช้ **java 8** เนื่องจาก Heroku รองรับแค่ java 8 เวอร์ชันเดียว by default (สามารถใช้ version อื่นได้ แต่ต้องมีการ configure ที่ถูกต้อง)
4. Wisdom Book Web
  - o เหมือน Spring web ทั่วไป มี HTML templates, controller, model, service โดย service จะต่อกับ Wisdom Book API
  - o ลองสังเกตในไฟล์ application.properties
    - i. `api.book.url` คือ ตัวแปร URL ที่เราจะใช้ในคลาส `BookService` เพื่อเชื่อมกับ Wisdom Book API
    - ii. `${API_BOOK}` คือ environment variable ที่เราสามารถเปลี่ยนเมื่อ deploy บน Heroku ได้
    - iii. `${API_BOOK:http://localhost:9001}` คือ เราสามารถระบุค่า default ของ environment variable ได้ด้วย หลัง : โดยในเครื่องเราใช้ `http://localhost:9001`
      - `${ENV_VAR:default_value}`

```
server.port=9000
```

```
api.book.url=${API_BOOK:http://localhost:9001}/book
```

- ลองสังเกตในคลาส BookService จะเห็นว่า เรานำตัวแปร `api.book.url` มาใช้เพื่อเชื่อมต่อกับ Wisdom Book API ทำให้เราไม่ต้อง hard-code URL ไว้ในโค้ด

```
@Service
public class BookService {

    @Autowired
    private RestTemplate restTemplate;

    @Value("${api.book.url}")
    private String url;

    public List<Book> getAll() {
        ResponseEntity<Book[]> response =
            restTemplate.getForEntity(url, Book[].class);
        Book[] restaurants = response.getBody();
        return Arrays.asList(restaurants);
    }

    public Book create(Book book) {
        return restTemplate.postForObject(url, book, Book.class);
    }
}
```

## 5. Wisdom Book API

- เหมือน Spring API ทั่วไป มี REST controller, model, repository โดยจะเชื่อมต่อกับ database
- Database จะเป็น PostgreSQL เนื่องจากเป็น database แบบเดียวที่เราใช้ได้โดยไม่ต้องใส่บัตรเครดิตใน Heroku account
- ลองสังเกตในไฟล์ application.properties
  - i. `spring.datasource.url` จะเป็น url ของ PostgreSQL และใช้ environment variable `DB_HOST`, `DB_PORT`, `DB_DB` และมีการกำหนด default เพื่อให้ dev ในเครื่องเราได้ง่าย
  - ii. `spring.datasource` อื่น ๆ ถูกปรับให้ใช้ PostgreSQL และใช้ environment variable

```
server.port=9001

## Datasource PostgreSQL
spring.datasource.url=jdbc:postgresql://${DB_HOST:localhost}:${DB_PORT:5430}/${DB_DB:postgres}
spring.datasource.driverClassName=org.postgresql.Driver
spring.datasource.username=${DB_USERNAME:manager}
spring.datasource.password=${DB_PASSWORD:abc123}
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect

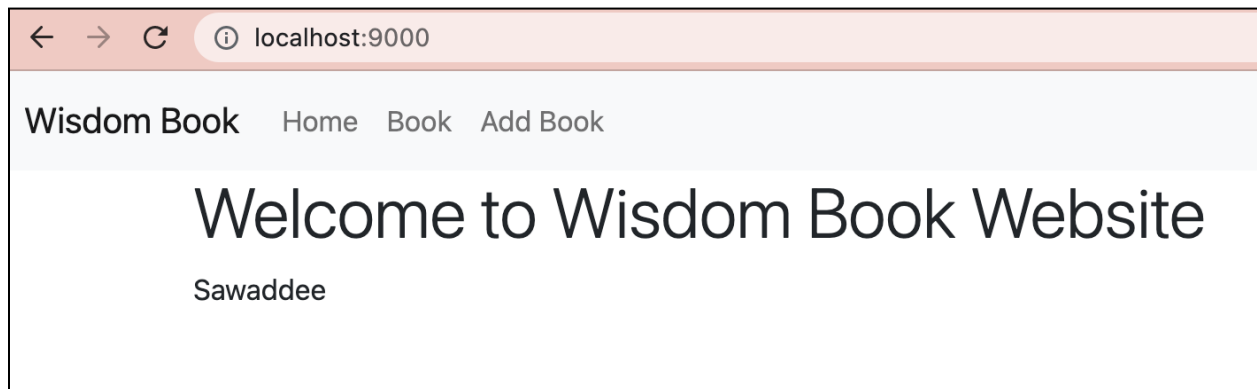
# JPA
```

```
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
```

6. ถ้านิสมิมี docker ในเครื่อง สามารถลองรันโปรแกรมทั้งสองบนเครื่อง local ได้ โดยให้รัน **PostgreSQL container** ด้วยคำสั่งนี้

```
docker run --name book-postgres -e POSTGRES_USER=manager -e
POSTGRES_PASSWORD=abc123 -p 5430:5432 -d postgres
```

7. จากนั้น ให้รัน Wisdom Book API และ Wisdom Book Front แล้วลองเพิ่มหนังสือ



## II. Signup and login to Heroku

1. Signup Heroku <https://www.heroku.com>

2. Download Heroku CLI

<https://devcenter.heroku.com/articles/getting-started-with-java#set-up>

In this step you'll install the **Heroku Command Line Interface (CLI)**. You use the CLI to manage and scale your applications, provision add-ons, view your application logs, and run your application locally.

Download and run the installer for your platform:

**macOS**

Download the installer

Also available via Homebrew:

```
$ brew install heroku/brew/heroku
```

**Windows**

Download the appropriate installer for your Windows installation:

64-bit installer

32-bit installer

### 3. เปิด CMD / Terminal แล้วใช้ heroku login command เพื่อ login to Heroku

Use the `heroku login` command to log in to the Heroku CLI:

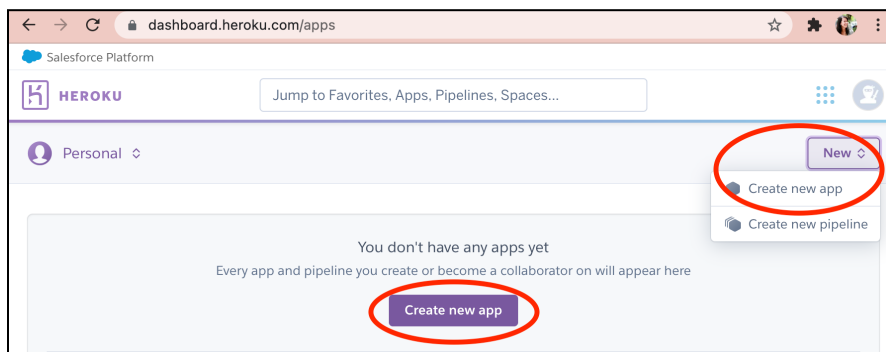
```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit
> Warning: If browser does not open, visit
> https://cli-auth.heroku.com/auth/browser/**
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

This command opens your web browser to the Heroku login page. If your browser is already logged in to Heroku, simply click the Log in button displayed on the page.

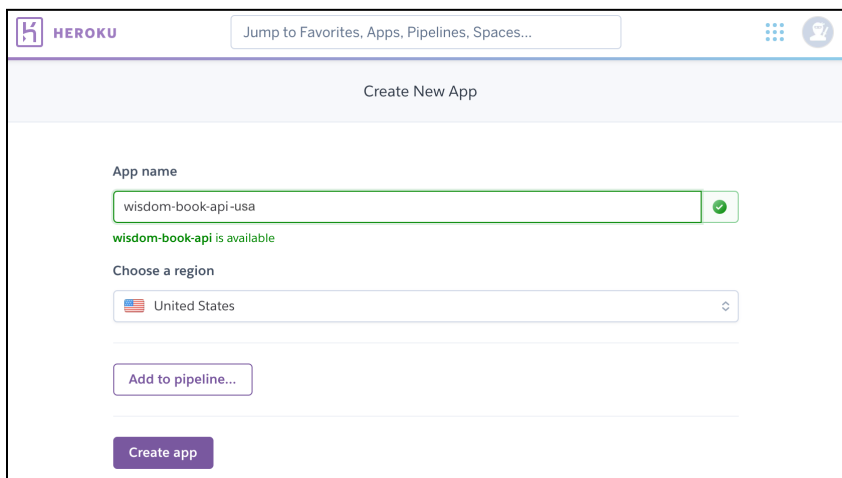
## III. การ deploy application ที่ใช้ database

### A. Create app บน heroku

1. เราจะมาเรียนรู้การ deploy application ที่ต้องใช้ database โดยเราจะ deploy โปรแกรม Wisdom Book API กันก่อน
2. ให้ไปที่หน้า dashboard (เมื่อ login จะไปที่หน้า dashboard อัตโนมัติ)
3. กดปุ่ม Create new app หรือไปที่เมนู New ข้างบน แล้วกด Create new app



4. ตั้งชื่อ app ตามต้องการ เช่น wisdom-book-api-ชื่อนิสิต (เช่น อาจารย์ใช้ wisdom-book-api-usa ชื่อจะได้ไม่ซ้ำกับเพื่อน) แล้วกดปุ่ม Create app



## B. Deploy app ขึ้น heroku

### 5. เปิด CMD / Terminal แล้ว

- `cd` ไปที่โฟลเดอร์ `wisdom-book-api` ที่เรา clone มา
- `heroku git:remote -a <App-name>`
  - บอก heroku ว่า repo นี้จะเชื่อมกับ app ไบน heroku (ให้ใส่ชื่อเดียวกับ App name ที่ตั้งไปด้านบน)
- `git push heroku main`
  - คือ การ deploy ขึ้น heroku ดังนั้น ถ้าหลังจากนี้มีการแก้ไขโค้ด ให้รันแค่คำสั่งนี้

```
$ cd wisdom-book-api
$ heroku git:remote -a wisdom-book-api-usa
$ git push heroku main
```

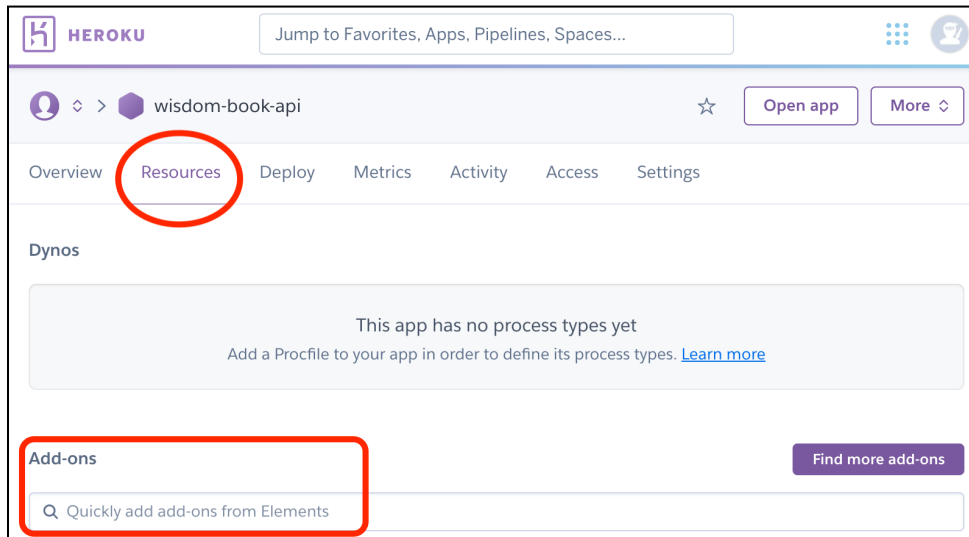
**หมายเหตุ** คำสั่ง heroku จะใช้ได้กับ **git repository** เท่านั้น เนื่องจากเรา clone repo ลงมา จึงเป็น git repo อยู่แล้ว หากนิสิตทำโปรแกรมขึ้นมาใหม่ จะต้อง `git init/add/commit` ให้เรียบร้อยก่อนใช้คำสั่ง heroku git และ git push

### 6. จะเห็นการ build, deploy และควรได้ output ดังนี้ ซึ่งถือว่า deploy สำเร็จแล้ว

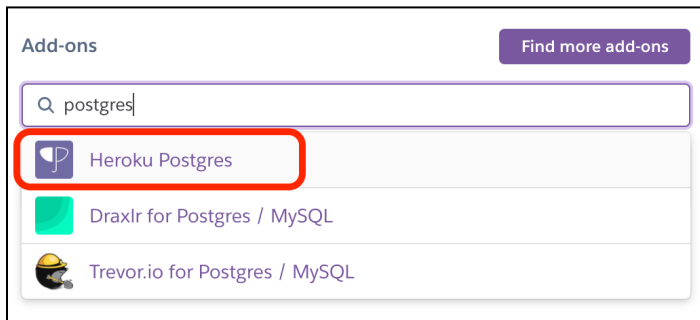
```
remote:      [INFO] BUILD SUCCESS
remote:      [INFO] -----
remote:      [INFO] Total time:  16.191 s
remote:      [INFO] Finished at: 2021-10-12T12:25:33Z
remote:      [INFO] -----
remote: ----> Discovering process types
remote:      Procfile declares types    -> (none)
remote:      Default types for buildpack -> web
remote: ----> Compressing...
remote:      Done: 86.4M
remote: ----> Launching...
remote:      Released v5
remote:      https://wisdom-book-api.herokuapp.com/ deployed to Heroku
remote: Verifying deploy... done.
To https://git.heroku.com/wisdom-book-api.git
* [new branch]    main -> main
```

### C. เพิ่ม database ใน heroku

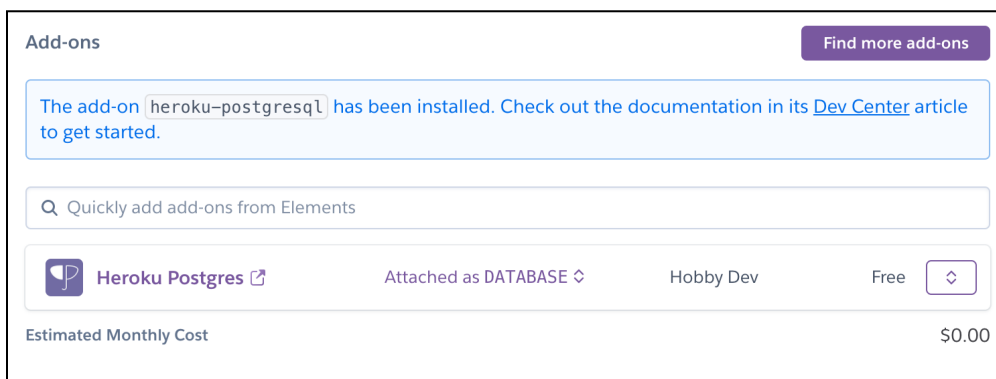
7. ไปที่ Heroku ที่ app wisdom-book-api-ชื่อนิสิต ในเว็บ
8. แล้วไปที่เมนู Resources และสังเกตช่อง Add-ons



9. จากนั้น พิมพ์ postgres ที่ช่อง Add-ons และเลือก Heroku Postgres

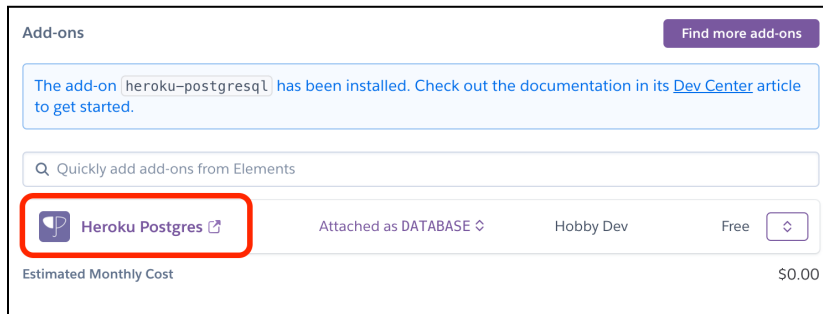


10. เราจะได้ database สำหรับ Wisdom Book API



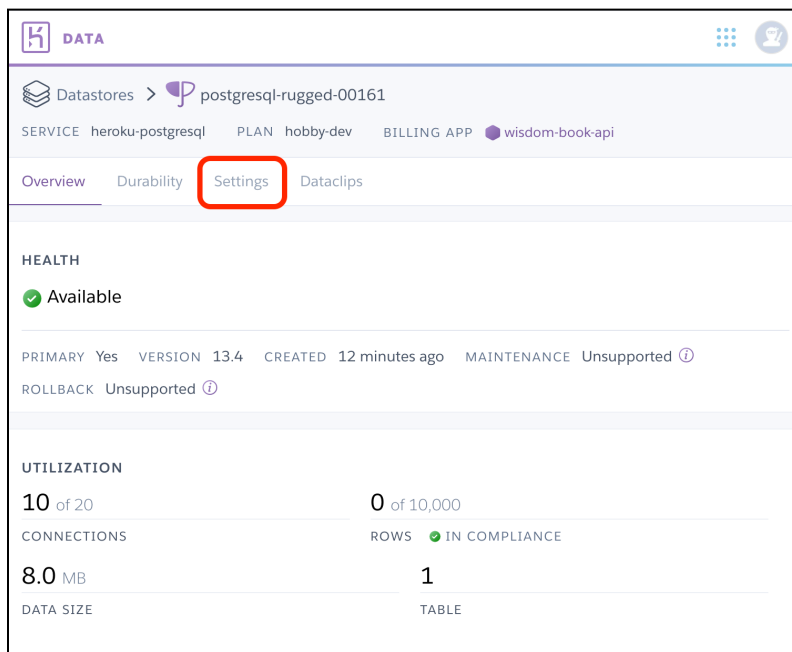
## D. ตั้งค่า environment variable

### 11. คลิกไปที่ Heroku Postgres



The screenshot shows the Heroku Add-ons page. At the top, a message states: "The add-on heroku-postgresql has been installed. Check out the documentation in its Dev Center article to get started." Below this is a search bar with the text "Quickly add add-ons from Elements". A red rectangle highlights the "Heroku Postgres" add-on card. The card shows the Heroku Postgres logo, the name "Heroku Postgres", a link to documentation, and details: "Attached as DATABASE", "Hobby Dev", "Free", and a dropdown arrow. At the bottom of the card, it says "Estimated Monthly Cost" and "\$0.00".

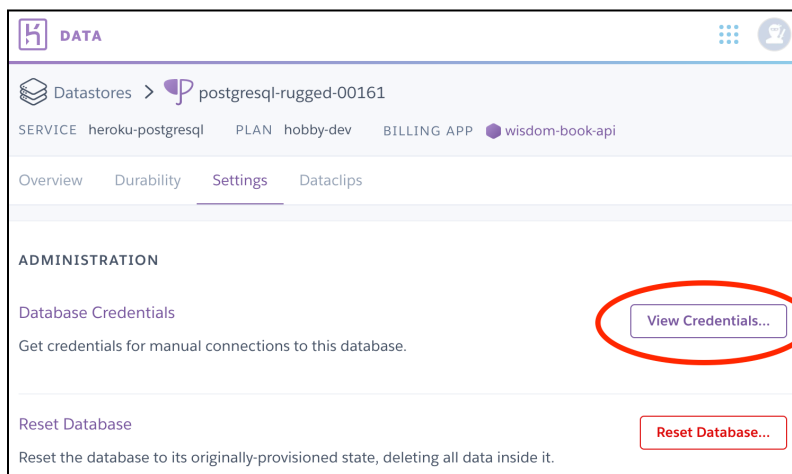
### 12. จะได้หน้านี้ ให้ไปที่แท็บ Settings



The screenshot shows the Heroku Postgres Settings page for the instance "postgresql-rugged-00161". The "Settings" tab is highlighted with a red rectangle. The page displays the following information:

- HEALTH:** Available (indicated by a green checkmark).
- PRIMARY:** Yes
- VERSION:** 13.4
- CREATED:** 12 minutes ago
- MAINTENANCE:** Unsupported (with an info icon)
- ROLLBACK:** Unsupported (with an info icon)
- UTILIZATION:** 10 of 20 connections, 0 of 10,000 rows. Rows are in compliance (indicated by a green checkmark).
- DATA SIZE:** 8.0 MB
- TABLE:** 1

### 13. กด View Credentials



The screenshot shows the Heroku Postgres Administration page for the instance "postgresql-rugged-00161". The "Settings" tab is selected. Under the "ADMINISTRATION" section, there is a "Database Credentials" subsection. A red rectangle highlights the "View Credentials..." button. Below this, there is a "Reset Database" subsection with a "Reset Database..." button.

#### 14. ให้ copy ข้อมูล Host, Database, User, Port, Password นี้ไว้

[Overview](#) [Durability](#) [Settings](#) [Dataclips](#)

ADMINISTRATION

Database Credentials Cancel

Get credentials for manual connections to this database.

Please note that **these credentials are not permanent**.

Heroku rotates credentials periodically and updates applications where this database is attached.

Host	ec2-35-174-56-18.compute-1.amazonaws.com
Database	dd3uucc5tr9mua
User	kgiwhngnydytrm
Port	5432
Password	3030f653e4f304e2ac30b6bf07bd089c5ac082a58293a86253a918e6e9724bb
URI	postgres://kgiwhngnydytrm:3030f653e4f304e2ac30b6bf07bd089c5ac082a58
Heroku CLI	heroku pg:psql postgresql-rugged-00161 --app wisdom-book-api

#### 15. มาที่หน้าของ wisdom-book-api-ชื่อนิสิต แล้วคลิกแท็บ Settings

**HEROKU**

> wisdom-book-api ☆ Open app More ▾

[Overview](#) [Resources](#) [Deploy](#) [Metrics](#) [Activity](#) [Access](#) [Settings](#)

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). Hide Create a Heroku Pipeline

Installed add-ons \$0.00/month [Configure Add-ons](#)

**Heroku Postgres**  
Hobby Dev  
postgresql-rugged-00161

**usa.sa@ku.th:** Detach  
HEROKU\_POSTGRESQL\_CRIMSON  
(@ref:postgresql-curly-10867)  
Today at 7:35 PM · v8



## 16. กดปุ่มที่ Config Vars

HEROKU Jump to Favorites, Apps, Pipelines, Spaces...

wisdom-book-api ☆ Open app More ▾

Overview Resources Deploy Metrics Activity Access Settings

App Information

App Name wisdom-book-api

Region United States

Stack heroku-20

Framework Java

Slug size 86.4 MiB of 500 MiB

Heroku git URL <https://git.heroku.com/wisdom-book-api.git>

Config Vars

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

Reveal Config Vars

## 17. ใส่ variable DB\_HOST, DB\_PORT, DB\_DB, DB\_USERNAME, DB\_PASSWORD ใน KEY และใส่ค่า VALUE ตาม Postgres settings ในข้อ 14

- (เราไม่ใช่ DATABASE\_URL ถ้า heroku สร้างมาให้ ก็ปล่อยไว้แบบนั้นได้ค่ะ)

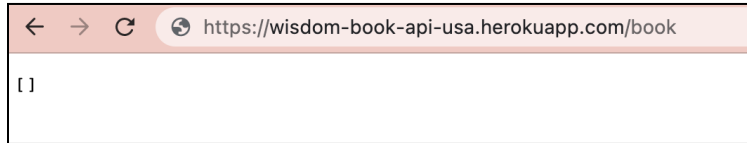
Config Vars Hide Config Vars

DATABASE_URL	postgres://kgiwhngnydytrm	✎ ✕
DB_HOST	ec2-35-174-56-18.compute-	✎ ✕
DB_PORT	5432	✎ ✕
DB_DB	dd3uucc5tr9mua	✎ ✕
DB_USERNAME	kgiwhngnydytrm	✎ ✕
DB_PASSWORD	3030f653e4f304e2ac30b6bf0	✎ ✕

KEY VALUE Add

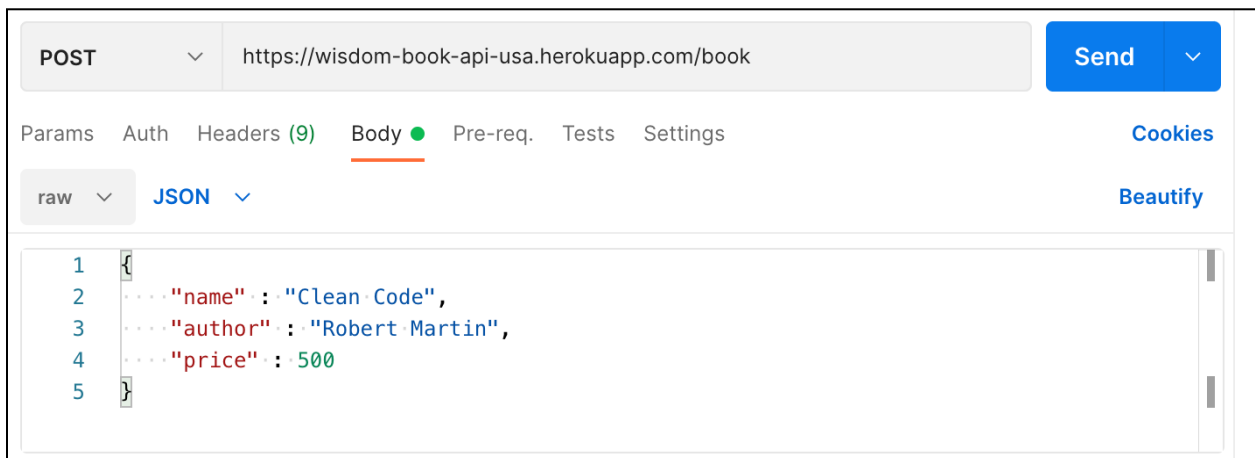
18. ลองไปที่หน้าเว็บ แล้วดูว่า ได้ empty list ของ book หรือไม่ ถ้าได้ แสดงว่าแอปทำงานถูกต้อง

<https://wisdom-book-api-usa.herokuapp.com/book>



19. นิสิตสามารถลอง post จาก Postman ได้

```
{  
  "name" : "Clean Code",  
  "author" : "Robert Martin",  
  "price" : 500  
}
```

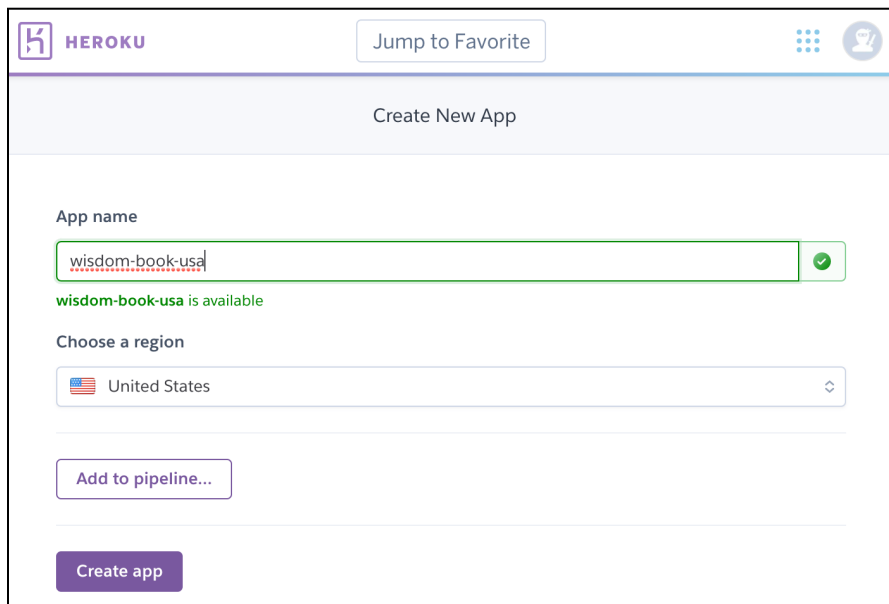


### III. การ deploy application ที่ไม่ต้องใช้ database

จะง่ายขึ้น เนื่องจากไม่ต้องเพิ่ม Add-on แต่กระบวนการอื่นเหมือนเดิม โดยเราจะ deploy โปรแกรม wisdom-book-front

#### A. Create app บน heroku

1. ให้ไปที่หน้า dashboard
2. ไปที่เมนู New ซ้ายบน แล้วกด Create new app
3. ตั้งชื่อ app ตามต้องการ เช่น wisdom-book-ชื่อนิสิต (อาจารย์ใช้ wisdom-book-usa) แล้วกดปุ่ม Create app



#### B. Deploy app ขึ้น heroku

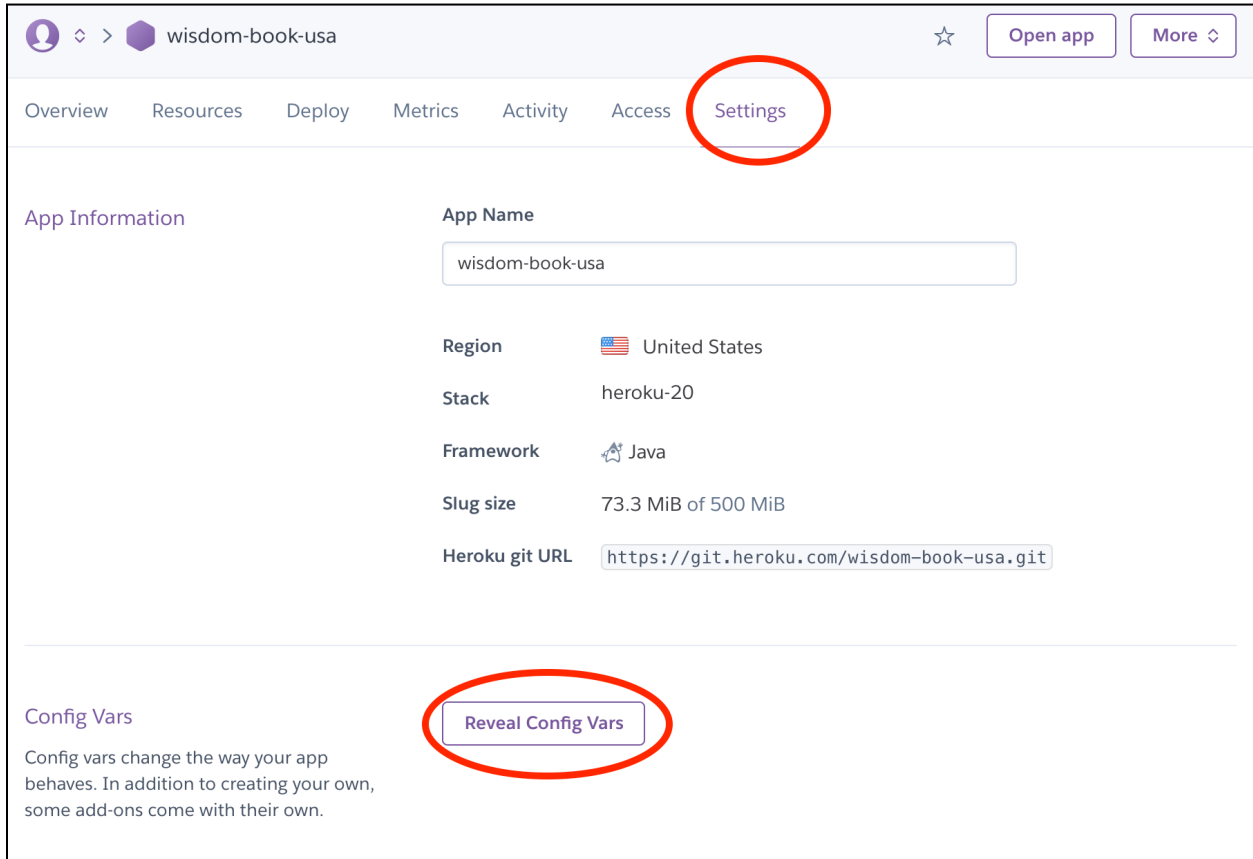
4. เปิด CMD / Terminal แล้ว
  - cd ไปที่โฟลเดอร์ wisdom-book-front ที่เรา clone มา
  - `heroku git:remote -a <App-name>`
    - i. บอก heroku ว่า repo นี้จะเชื่อมกับ app ใดบน heroku (ให้ใส่ชื่อเดียวกับ App name ที่ตั้งไปด้านบน)
  - `git push heroku main`
    - i. คือ การ deploy ขึ้น heroku ดังนั้น ถ้าหลังจากนี้มีการแก้ไขโค้ด ให้รันแค่คำสั่งนี้

```
$ cd wisdom-book-front
$ heroku git:remote -a wisdom-book-usa
$ git push heroku main
```

5. จะเห็นการ build, deploy ถ้าเห็น Build Success ถือว่า deploy สำเร็จแล้ว

### C. ตั้งค่า environment variable

#### 6. มาที่หน้าของ wisdom-book-ใช้นิสิต แล้วคลิกแท็บ Settings และไปที่ Config Vars



The screenshot shows the Heroku dashboard for the application 'wisdom-book-usa'. The 'Settings' tab is selected and highlighted with a red circle. Below the 'App Information' section, the 'Reveal Config Vars' button is highlighted with a red circle.

**App Information**

App Name	wisdom-book-usa
Region	United States
Stack	heroku-20
Framework	Java
Slug size	73.3 MiB of 500 MiB
Heroku git URL	https://git.heroku.com/wisdom-book-usa.git

**Config Vars**

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

[Reveal Config Vars](#)

#### 7. ใส่ environment variable `API_BOOK`

- โดยใส่ url <https://wisdom-book-api-ใช้นิสิต.herokuapp.com> (ไม่มีทับด้านหลังสุด)



The screenshot shows the Heroku Config Vars page. The 'API\_BOOK' key is entered with the value 'https://wisdom-book-api-ใช้นิสิต.herokuapp.com'. The 'Add' button is visible.

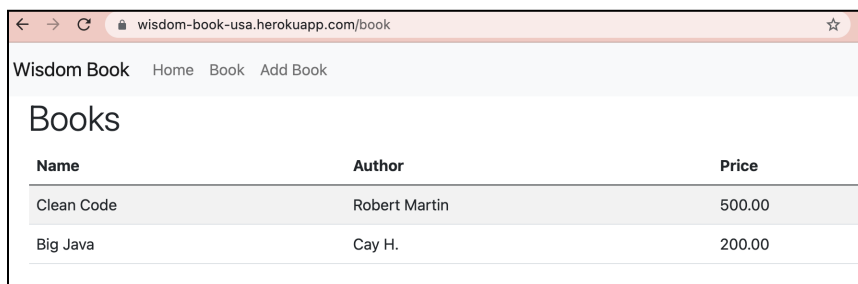
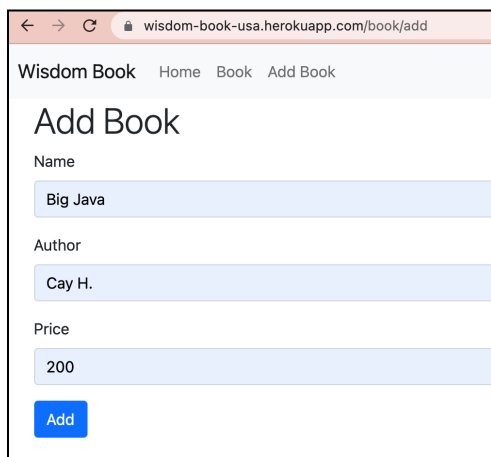
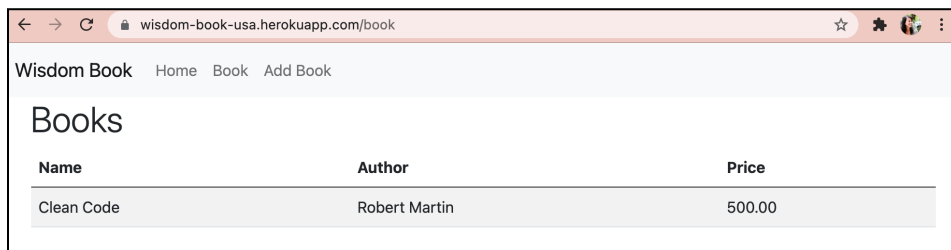
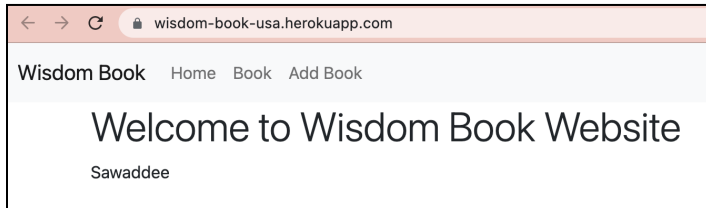
**Config Vars**

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

API_BOOK	https://wisdom-book-api-ใช้นิสิต.herokuapp.com	<a href="#">Hide Config Vars</a>
KEY	VALUE	Add

8. ลองไปที่ลิงก์แอฟ แล้วดูว่า ได้หน้าเว็บ Wisdom Book หรือไม่ ถ้าได้ แสดงว่าแอฟทำงานถูกต้อง
- ลองเพิ่มหนังสือ แล้วไปที่หน้า Book

<https://wisdom-book-ชื่อนิสิต.herokuapp.com/>



9. ลองไปที่ Wisdom Book API แล้วดูว่ามีข้อมูลหนังสือที่เพิ่มมาหรือไม่

