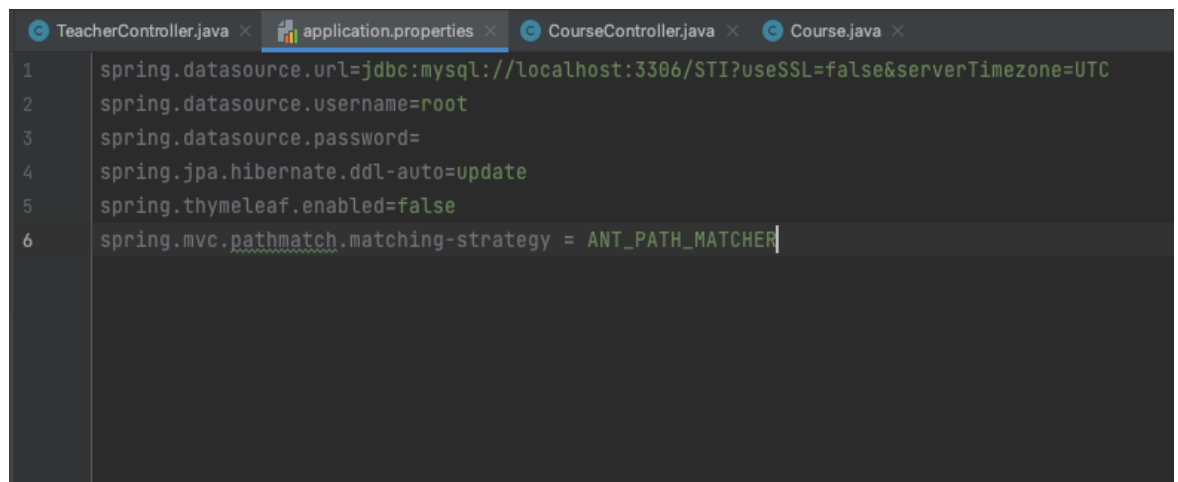


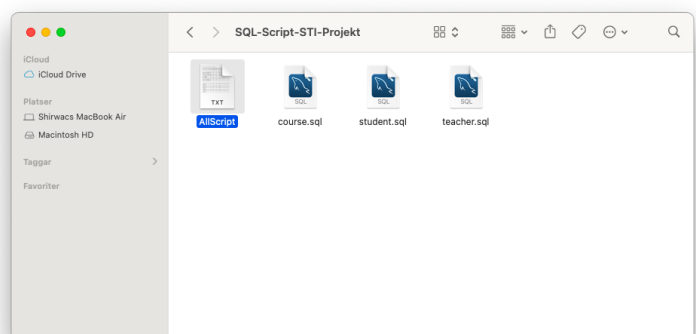
## ***Några steg du behöver göra för att programmet ska fungera för dig.***

1. Du behöver skapa en databas som heter STI innan du kör igång programmet, det kan du göra via terminalen. Sedan måste du konfigurera i Application.properties ligger sti-Dao under resources, sen är det bara att konfigurera med någon användare eller root. Prova logga in på mysql på terminalen, använd samma inloggning här med. Du kan sedan prova starta programmet under mappen app, om programmet körs så är allt som det ska vara.

A screenshot of an IDE window showing the 'application.properties' file. The file contains the following configuration: spring.datasource.url=jdbc:mysql://localhost:3306/STI?useSSL=false&serverTimezone=UTC, spring.datasource.username=root, spring.datasource.password=, spring.jpa.hibernate.ddl-auto=update, spring.thymeleaf.enabled=false, and spring.mvc.pathmatch.matching-strategy = ANT\_PATH\_MATCHER. The tabs at the top are TeacherController.java, application.properties, CourseController.java, and Course.java.

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/STI?useSSL=false&serverTimezone=UTC
2 spring.datasource.username=root
3 spring.datasource.password=
4 spring.jpa.hibernate.ddl-auto=update
5 spring.thymeleaf.enabled=false
6 spring.mvc.pathmatch.matching-strategy = ANT_PATH_MATCHER
```

2. Det finns en nackdel med vårt program då knappt finns några validering för data som kommer in, kan det lätt dyka upp massa error. Därför har jag genererat lite data i databasen. Som du kan använda dig av för att testa funktionerna i programmet. Jag kommer både bifoga en txt fil med SQL funktionerna ifall du vill använda i terminalen direkt eller SQL fil om du använder WorkBench. Bifogar den i Projektet mappen.



3. Så här ser databasen ut,  
(Använde mig av random generator)

```
mysql> select * from student;
```

id	course_code	first_name	last_name	person_number
1	5	Chalmers	Beecraft	2302823176
2	5	Denny	Fawdery	7784016468
3	1	Ruggiero	Hubber	5599739430
4	5	Kurtis	Franses	5271588033
5	4	Franny	Jerrones	4970250839
6	2	Bevan	Rappport	6613938122
7	1	Sargent	Mawby	3907051270
8	3	Ev	Gladyer	7746836909
9	3	Jaymie	Crumley	2629518574
10	2	Mabelle	Tarrier	6030189557

```
10 rows in set (0,00 sec)
```

```
mysql> select * from teacher;
```

id	course_code	first_name	hourly_rate	last_name	person_number
1	5	Benjamin	189	Pogosian	0345868935
2	4	Ulrick	190	Wrought	6839431517
3	5	George	121	Zanicchi	5077309027
4	2	Standford	185	D'Abbot-Doyle	2405496216
5	1	Manolo	130	Cottel	0308465962
6	5	Trudy	122	Clery	2238953957
7	1	Hayward	130	Berns	1362049948
8	5	August	144	Idenden	9800039082
9	3	Stanwood	177	Mallabar	1559869593
10	3	Osbourn	184	Peppard	6933907309

```
10 rows in set (0,00 sec)
```

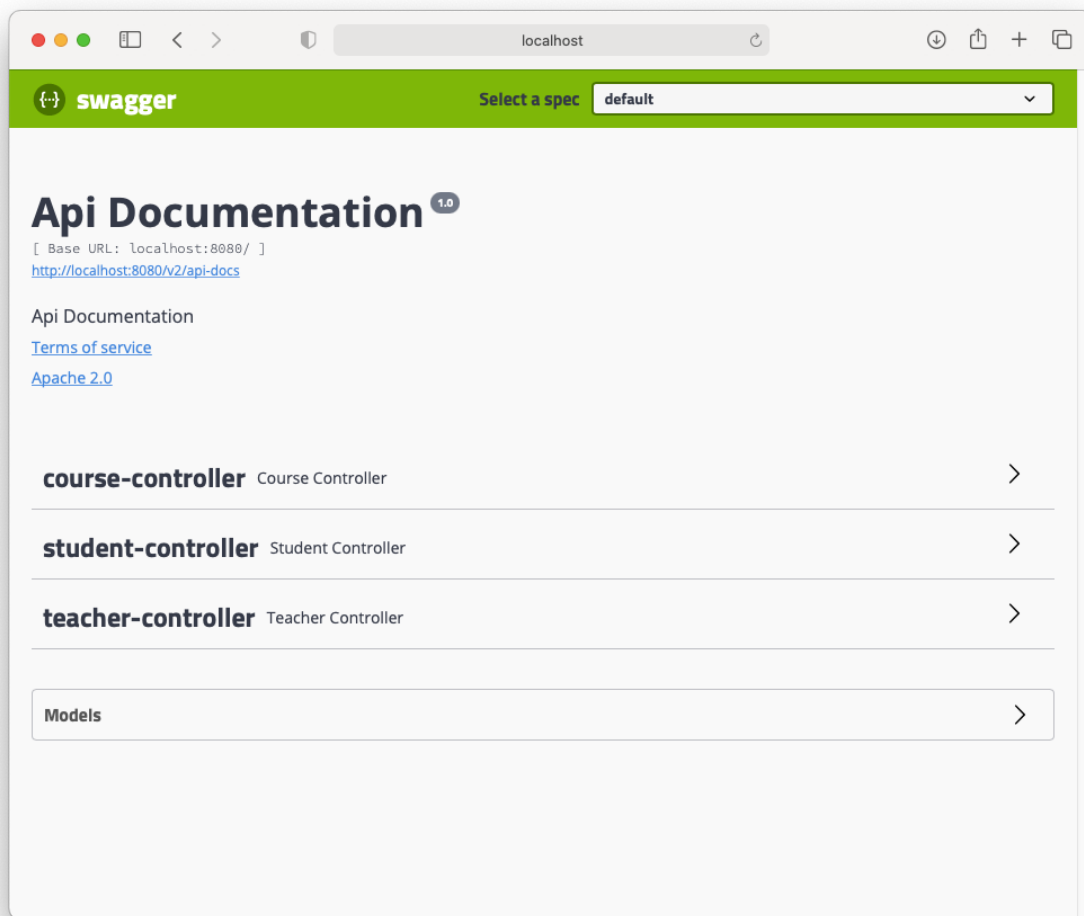
```
mysql> Select * from course;
```

id	course_code	course_name	total_hours	yh_points
1	2999	Java	90	50
2	2526	SQL	40	50
3	2657	CSS	60	40
4	1707	HTML	40	25
5	1352	JavaScript	90	25

```
5 rows in set (0,00 sec)
```

```
mysql> █
```

- Nu när SpringBoot är uppe och rullar, kan du prova funktionerna via Swaggern, så är det bara att slå i den här i webbläsaren <http://localhost:8080/swagger-ui.html>. Du borde få upp den här sidan.



Nu är det bara att prova funktionerna, vi använde oss istället id nummer som vi generar istället för att personnummer, för jag försökte få det funka genom att skapa en *findByPersonNumber* funktion i *repon* men det strulade alltid och fick alltid att repo var null.

Sen med Junit, blir det väldigt komplicerat att testa dao, eftersom vi använder oss av JPA repositories funktionen för att hämta in data då vi inte implementerar någon kod alls själv.

Hoppas den här guiden hjälpte.