

Język: **frontend** ionic-angular typescript javascript, **backend** kotlin

Środowisko:

- IntelliJ Community
- PostgreSQL (hasła, użytkownik, ścieżka do bazy muszą się zgadzać z tymi przyjętymi w projekcie, więc Szymon musi przekazać)
- pgAdmin – zarządzanie bazą
- nodeJs
- w CMD → npm install -g ionic

Front-end:

Stawianie projektu:

CMD → wejść do folderu gdzie ma się wygenerować projekt

npm install -g ionic

ionic start clearApp blank

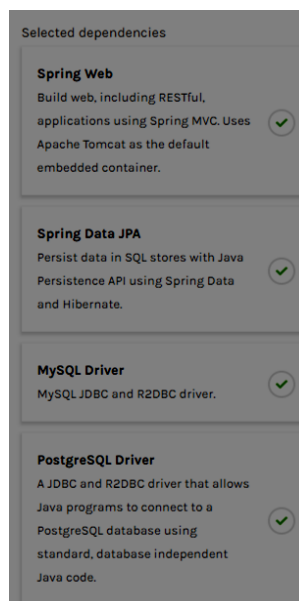
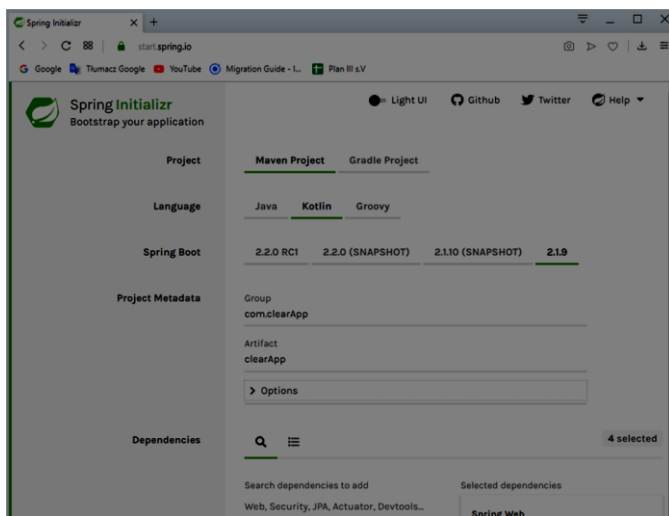
Uruchamianie:

‘open folder’ z lab_dream_frontend w intelliJ, otwórz w programie terminal i tam wpisz **ionic serve**, po tym poleceniu każda wprowadzona zmiana będzie aktualizować automatycznie wygląd, więc nie trzeba tego robić ręcznie powinna się sama otworzyć strona z apką, ale jak nie to wpisać w przeglądarkę **localhost:8100**

Back-end:

Stawianie projektu:

start.spring.io



Uruchamianie:

‘open folder’ z lab_dream_backend w intelliJ i wystartuj program strzałką z górnego prawego rogu, jeśli nie jest aktywny to trzeba ustawić główną klasę ręcznie src/main/kotlin/lab_dream/LabDreamApplication.kt prawym przyciskiem myszy i wybrać ‘Run’

W pliku application.properties trzeba dodać link i dane logowania do bazy danych żeby można było się z nią łączyć (są podane, ale mogą się nie zgadzać, więc trzeba je poprawić wg tego co poda Szymon)

Działanie sprawdza się na **np. localhost:8080/ścieżka/ścieżka**, w kontrolerach tworzy się metody które przypisuje się do danej ścieżki i to co dana metoda zwraca lub wykonuje, po wejściu w tą ścieżkę powinno się wykonać return zwraca na ekran przeglądarki, voidy powinny wykonać daną czynność np. usunąć usera z bazy (odsylam do MusicRepo, bo to tam jest i łatwiej zrozumieć jak się widzi, teraz piszemy w kotlinie, ale działanie takie samo) folder ‘controller’ → pliki z kontrolerami- funkcjonalność, ‘model’ → pliki z modelami obiektów, ‘repository’ → pliki z interfejsami łączącymi z bazą