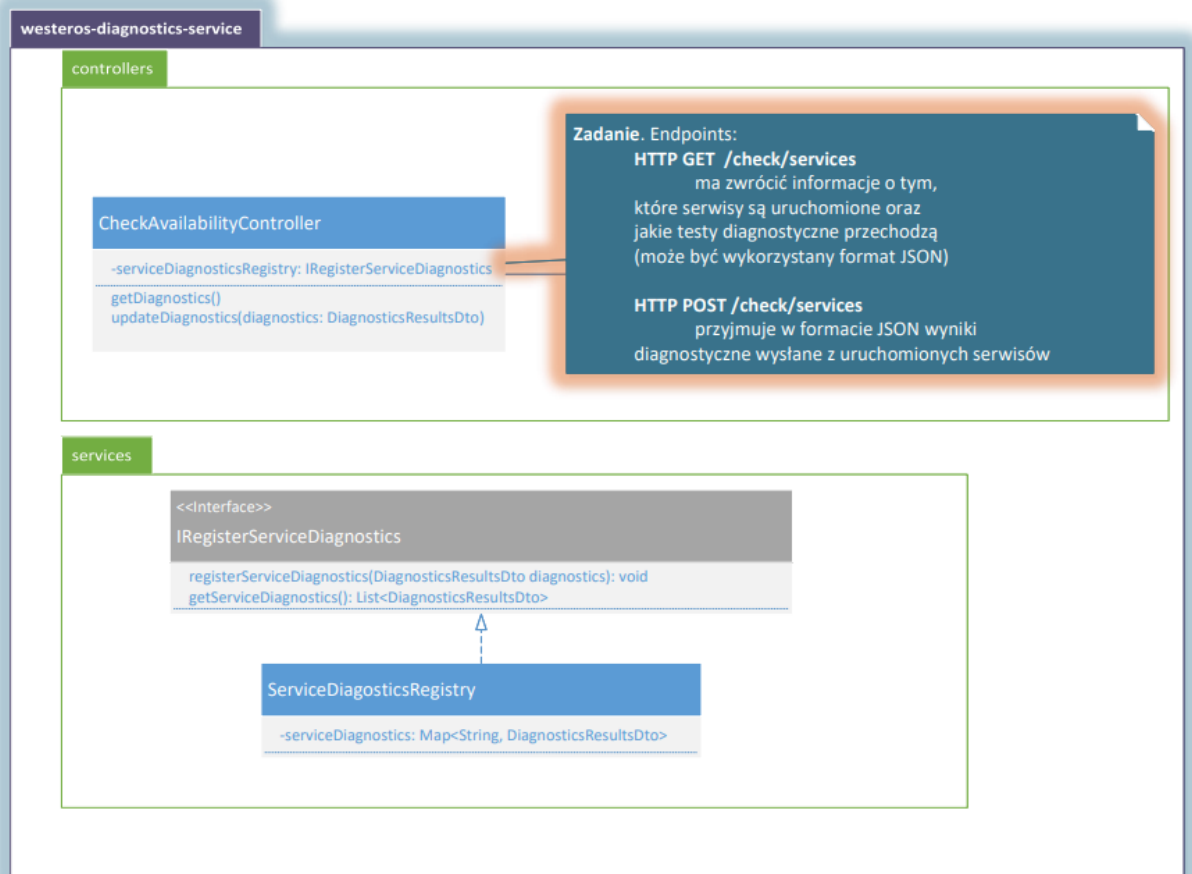


## Lab 11 – instrukcje

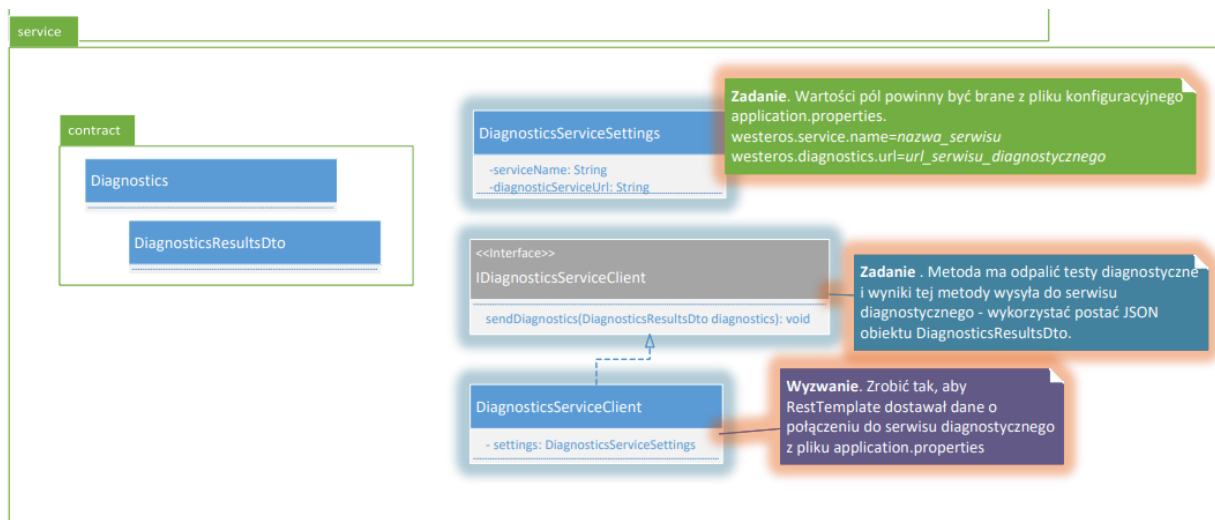
1. Zaimplementuj klasy testów diagnostycznych (implementując interfejs IDiagnose)
  - a. W module westeros-diagnostics-tools test sprawdzający ilość miejsca na dysku twardym. Ten test będzie współdzielony między modułami
  - b. W module westeros-updater:
    - Skonfiguruj zależność od modułu westeros-diagnostics-tools
    - Zaimplementuj test sprawdzający połączenie z bazą danych DatabaseConnectivityDiagnostics
    - Zaimplementuj test sprawdzający połączenie z serwisem themoviedb.org TheMovieDbApiConnectivityDiagnostics
  - c. W module westeros-web-api:
    - Skonfiguruj zależność od modułu westeros-diagnostics-tools
    - Zaimplementuj test sprawdzający połączenie z bazą danych DatabaseConnectivityDiagnostics
2. Zaimplementuj klasę DiagnosticsRunner interfejsu IRunDiagnoses. Wszystkie poprzednio napisane testy powinny być komponentami w Springu – wtedy bez większego problemu lista z testami diagnostycznymi będzie wypełniona przez kontener IOC
3. Wykorzystaj DiagnosticsRunner w kontrolerze DiagnosticsController i zaimplementuj akcję kontrolera która odpali testy diagnostyczne danego serwisu, po wejściu na określony w diagramie klas endpoint.
  - a. Uruchom moduł westeros-updater i sprawdź czy działają endpointy:
    - <http://localhost:8081/diagnostics>
      - Tutaj powinien wyświetlić się napis „ALIVE”
    - <http://localhost:8081/diagnostics/check>
      - Tutaj powinny pokazać się wyniki testów diagnostycznych
  - b. Uruchom moduł westeros-web-api i sprawdź czy działają endpointy:
    - <http://localhost:8080/diagnostics>
      - Tutaj powinien wyświetlić się napis „ALIVE”
    - <http://localhost:8080/diagnostics/check>
      - Tutaj powinny być wyniki testów diagnostycznych

## Lab 12 – instrukcje

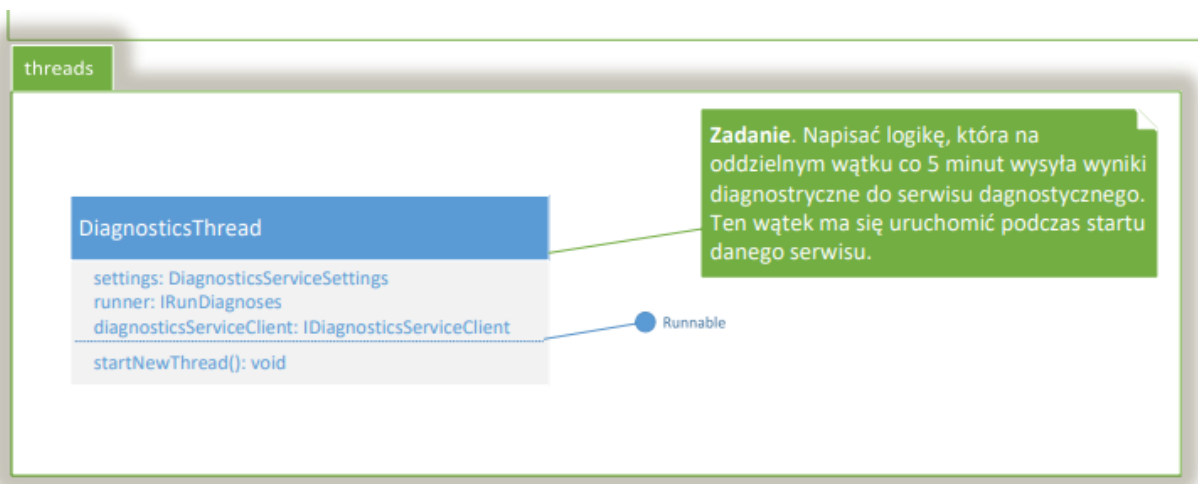
### 1. Zaimplementuj serwis diagnostyczny



- Dodaj zależność od modułu **westeros-diagnostics-tools** aby móc w kodzie wykorzystać klasę **DiagnosticsResultsDto**
  - Zaimplementuj klasę **ServiceDiagnosticsRegistry**, która implementuje interfejs **IRegisterServiceDiagnostics**
    - **Wyzwanie:** zrobić tak, że jeśli dany serwis nie odpowiada przez ostatnie 30 sekund, a był uruchomiony wcześniej, to flaga 'isAlive' dla danego serwisu zmienia swoją wartość na false no i jak serwis znowu będzie odpowiadał to ustawi wartość na true
  - Zaimplementuj kontroler **CheckAvailabilityController**, który korzystając z interfejsu **IRegisterServiceDiagnostics** zwraca dane diagnostyczne uruchomionych serwisów (według specyfikacji diagramów)
2. Zaimplementuj klienta do serwisu diagnostycznego w module **westeros-diagnostics-tools**



3. **Wyzwanie.** Wykorzystując narzędzia do harmonogramów zadań napisz zaqadanie, które co 10 sekund będzie wysyłać dane diagnostyczne do serwisu diagnostycznego



Tutaj zamiast klasy DiagnosticsThread powinna być konfiguracja dla harmonogramów springa.