

N-Layered architectuur

LEERDOELEN

1. Je kan het gepaste project type kiezen voor de Data Access Layer.
2. Je kan het gepaste project type kiezen voor de Business Logic Layer.
3. Je kan het gepaste project type kiezen voor de API Layer.
4. Je kan de Data Access Layer structureren.
5. Je kan de Business Logic Layer structureren.
6. Je kan de API Layer structureren.
7. Je kan gepaste interfaces definiëren.
8. Je kan gepaste service classes implementeren.
9. Je kan controllers implementeren.
10. Je kan in functie van datavalidatie custom exceptions definiëren.
11. Je kan in functie van datavalidatie custom exceptions gebruiken.
12. Je kan exceptions opvangen en omzetten naar een gepast ActionResult.
13. Je kan Data Access Layer componenten op een gepaste manier registreren bij de DI container.
14. Je kan Business Logic Layer componenten op een gepaste manier registreren bij de DI container.

OEFENINGEN

1. Refactor de Racing applicatie zodat je op een goede manier de Data Access Layer implementeert. Denk hierbij onder andere aan:
 - a. De DbContext class
 - b. Entity classes
 - c. EntityConfiguration classes
 - d. Extension methods
 - e. Namespaces
2. Refactor de Racing applicatie zodat je op een goede manier de Business Logic Layer implementeert. Denk hierbij onder andere aan:
 - a. Gepaste interfaces
 - b. Gepaste service classes
 - c. Custom exceptions
 - d. Datavalidatie
 - e. Extension methods
 - f. Namespaces
 - g. Project references

3. Refactor de Racing applicatie zodat je op een goede manier de API Layer implementeert.
Denk hierbij onder andere aan:
 - a. Gepaste Controller classes
 - b. Opvangen van exceptions en omzetten in juiste ActionResult
 - c. Project references