

Oppsummering av klasser og metoder i API HarbNet v.1

Obs obs: Dette er kun ment som en rask oversikt. For dypere forklaring og fullstendig oversikt, se «Dokumentasjon av fullstendig API».

Tar forbehold om småfeil og unøyaktigheter som ikke nødvendigvis reflekterer innholdet i den fullstendige oversikten, «Dokumentasjon av fullstendig API».

IHarbor

Hvert havn-objekt representerer en enkelt havn og holder informasjon om skipene som brukes i simuleringen, deres posisjoner, og tilgjengelige kaiplasser og lagerplasser for containere. Havnen er ansvarlig for det meste av det som skjer i simuleringen, inkludert håndtering av skipenes ankomst og avreise, og laste av og på containere.

Her er en kort oversikt over konstruktøren og attributtene i IHarbor:

- `Harbor(IList<Ship> listOfShips, int numberOfSmallLoadingDocks, int numberOfMediumLoadingDocks, int numberOfLargeLoadingDocks, int numberOfSmallShipDocks, int numberOfMediumShipDocks, int numberOfLargeShipDocks, int numberOfSmallContainerSpaces, int numberOfMediumContainerSpaces, int numberOfLargeContainerSpaces)`
Konstruktøren tar imot ti attributter som definerer listen over skip, antall lastekaiplasser, kaiplasser for skip, og lagerplasser for containere basert på deres størrelser.
- `ID` Returnerer den unike IDen (Guid) til havnen.
- `LoadingDockIsFreeForAllDocks()` Returnerer en liste med key-value par der key er IDen (Guid) til en lastekai, og value er en boolsk verdi som indikerer om lastekaien er ledig.
- `LoadingDockIsFree(Guid dockID)` Tar imot en ID (Guid) til en lastekai og returnerer en boolsk verdi som indikerer om lastekaien er ledig.
- `GetShipStatus(Guid ShipID)` Tar imot en ID (Guid) til et skip og returnerer en Status enum som representerer skipets status.
- `ShipDockIsFreeForAllDocks()` Returnerer en liste med key-value par der key er IDen (Guid) til en kaiplass for skip, og value er en boolsk verdi som indikerer om kaiplassen er ledig.
- `GetStatusAllShips()` Returnerer en liste med key-value par der key er et Ship-objekt og value er en Status enum som beskriver skipets status.
- `AnchorageID` Returnerer IDen (Guid) til ankerplassen.
- `TransitLocationID` Returnerer IDen (Guid) til transittlokasjonen.

IShip

Hvert skip-objekt representerer et lasteskip av en viss størrelse (definert av ShipSize enum), og kan enten være for en enkeltseilas eller for kontinuerlige seilaser.

Her er en kort oversikt over konstruktøren og metodene i IShip:

- `Ship (ShipSize shipSize, DateTime StartDate, bool IsForASingleTrip, int roundTripInDays, int numberOfcontainersOnBoard)` Konstruktøren tar imot fem attributter som definerer skipets størrelse, startdato, om det er for en enkeltseilas, rundtur i dager, og antall containere ombord ved første ankomst.
- `ID` Returnerer den unike IDen (Guid) til skipet.
- `ShipSize` Returnerer en ShipSize enum som definerer størrelsen på skipet.
- `StartDate` Returnerer tidspunktet når skipet først ankommer havnen.
- `RoundTripInDays` Returnerer antall dager det tar fra skipet legger fra kai med ny last til det er tilbake igjen ved havnen.
- `CurrentLocation` Returnerer IDen (Guid) til skipets nåværende posisjon. Posisjonene kan være Transit (på havet), Anchorage (til anker), LoadingDock (ved lastekai), eller ShipDock (ved kai etter enkeltseilas).
- `History` Returnerer en liste med Event-objekter som representerer hver statusendring for skipet.
- `ContainersOnBoard` Returnerer en liste med Container-objekter som representerer containerene som skipet har i lasterommet i nåværende øyeblikk.
- `ContainerCapacity` Returnerer antall containere skipet har plass til i lasterommet sitt.
- `MaxWeightInTonn` Returnerer maksimal vekt i tonn skipet kan ha før det synker. Dette inkluderer vekten av skipet pluss vekten av lasten ombord.
- `BaseWeightInTonn` Returnerer vekten i tonn av skipet når det ikke har noen last ombord.
- `CurrentWeightInTonn` Returnerer vekten i tonn av skipet på det tidspunktet metoden kalles. Dette inkluderer vekten av skipet når det er tomt pluss summen av vekten til alle containerene i lasterommet. Denne verdien kan aldri overskride MaxWeightInTonn.

IContainer

Representerer en container og inneholder relevant informasjon som ID, historikk, størrelse, vekt og nåværende posisjon. Containere opprettes typisk sammen med nye skip.

Her er en kort oversikt over metodene i IContainer:

- `ID` Returnerer den unike IDen (Guid) til containeren.
- `History` Returnerer en liste med Event-objekter som representerer hver statusendring for containeren.
- `Size` Returnerer en ContainerSize enum som definerer størrelsen på containeren.
- `WeightInTonn` Returnerer vekten til containeren i tonn.
- `CurrentPosition` Returnerer IDen (Guid) til objektet som for øyeblikket lagrer containeren.
- `GetStatus()` Returnerer en status enum som indikerer containerens status på det tidspunktet metoden blir kalt.

Container-objekter opprettes automatisk ved opprettelse av skip for å minimere brukerens arbeid ved oppstart av en ny simulering. Konstruktøren for Container-klassen er derfor ikke tilgjengelig for brukere av APllet.

ShipSize

Brukes til å angi størrelsen på et skip. Hver skip har en ShipSize enum som definerer dens størrelse.

Her er en kort oversikt over størrelsene:

- `None = 0` C# konvensjon, udefinert størrelse.
- `Small` Skipet er av liten størrelse.
- `Medium` Skipet er av middels størrelse.
- `Large` Skipet er av stor størrelse.

ContainerSize

Alle containere har en ContainerSize enum som representerer deres størrelse.

Her er en kort oversikt over størrelsene og deres tilsvarende vekt i tonn:

- `None = 0` C# konvensjon, 0 tonn.
- `Small = 10` Liten container, 10 tonn.
- `Medium = 15` Medium container, 15 tonn.
- `Large = 20` Stor container, 20 tonn.

For eksempel vil en stor (Large) container veie 20 tonn.

IEvent

Holder informasjon om skip og containere i en simulering. Hvert event representerer en statusendring for et skip eller en container på et gitt tidspunkt.

Her er en kort oversikt over metodene i IEvent:

- **Subject** Returnerer ID (Guid) til skipet eller containeren eventet er knyttet til.
- **SubjectLocation** Returnerer ID (Guid) til lokasjonen der subjectet befinner seg når eventet opprettes. Lokasjonene kan være forskjellige for skip (Anchorage, Transit, LoadingDock, ShipDock) og containere (Ship, ContainerSpace).
- **PointInTime** Returnerer tidspunktet når eventet ble opprettet, som er når subjectet endrer status.
- **Status** Returnerer en Status enum som beskriver subjectets status når eventet opprettes.
Statusene varierer for skip (*DockingToLoadingDock*, *DockingToShipDock*, *Undocking*, *Loading*, *Unloading*, *Transit*, *Anchoring*) og containere (*Loading*, *Unloading*, *Transit*, *InStorage*).

Status

Hver status representerer en handling et skip eller en container kan utføre på et gitt tidspunkt.

Statusene som er definert er som følger:

- **None = 0** C# konvensjon
- **DockingToLoadingDock** Skipet legger seg til en lastekai.
- **DockingToShipDock** Skipet legger seg til en kai hvor de ikke kan laste.
- **Undocking** Skipet legger fra kai.
- **Loading** Skipet laster ombord containere, eller en container lastes ombord på et skip.
- **Unloading** Skipet laster av containere, eller en container lastes av et skip.
- **Transit** Skipet er på havet.
- **InStorage** Containeren er lagret på land på havna.
- **Anchoring** Skipet ligger til anker og venter på en ledig kaiplass.
- **UnloadingDone** Skipet er ferdig med å laste av lasten sin.
- **LoadingDone** Skipet er ferdig med å laste på ny last.

ILog

Log-objekter brukes til å registrere statusen til alle skip og containere i simuleringen på et gitt tidspunkt. Hver dag opprettes et nytt Log-objekt som gir status og plassering for alle skip og containere på det tidspunktet.

Her er en kort oversikt over metodene i ILog:

- `Time` Returnerer et `DateTime`-objekt som angir tidspunktet i simuleringen da Log-objektet ble opprettet.
- `ShipsInAnchorage` Returnerer en liste med Ship-objekter som inneholder alle skip som lå til anker da Log-objektet ble opprettet.
- `ShipsInTransit` Returnerer en liste med Ship-objekter som inneholder alle skip som var på havet da Log-objektet ble opprettet.
- `ShipsDockedInShippingDocks` Returnerer en liste med Ship-objekter som inneholder alle skip som lå ved en lastekai og lastet av og på containere da Log-objektet ble opprettet.
- `ShipsDockedInShippingDocks` Returnerer en liste med Ship-objekter som inneholder alle skip som lå ved en kaiplass for skip som ikke laster av eller på containere da Log-objektet ble opprettet.
- `ContainersInHarbour` Returnerer en liste med Container-objekter som inneholder alle containere som var lagret på land i havnen da Log-objektet ble opprettet.
- `PrintInfoForAllShips()` Printer informasjon om alle skip og deres posisjoner til konsollen.
- `PrintInfoForAllContainers()` Skriver ut informasjon om alle containere og deres posisjoner til konsollen.

ISimulation

Simulation-objekter brukes til å kjøre simuleringen av en havn.

Her er en kort oversikt over konstruktøren og metodene i ISimulation:

- `Simulation (Harbor harbor, DateTime simulationStartTime, DateTime simulationEndTime)` Konstruktøren tar imot tre attributter som definerer havnen som skal brukes i simuleringen, starttidspunktet for simuleringen, og sluttidspunktet for simuleringen.
- `History` Returnerer en liste med Log-objekter som inneholder et Log-objekt for hver dag simuleringen har pågått.
- `Run()` Starter simuleringen og returnerer en liste med Log-objekter som forteller om historien til hvordan simuleringen har utspilt seg.
- `PrintShipHistory()` Printer hele historien til alle skip for den forrige simuleringen som har blitt utført.
- `PrintContainerHistory()` Printer hele historien til alle containere for den forrige simuleringen som har blitt utført.