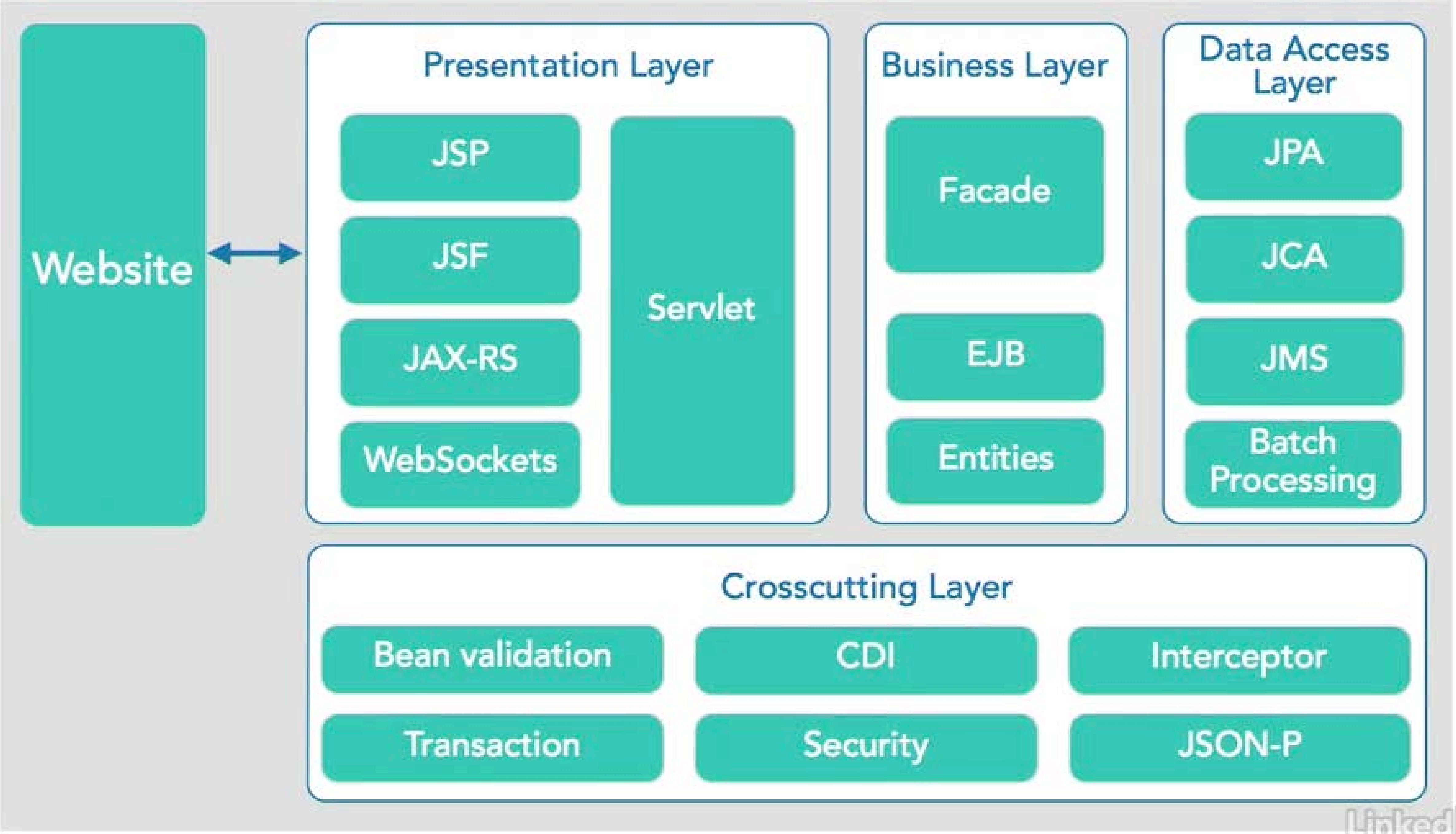
EJB3 Introduction

New programming model

- Convention over configuration
- Dependency injection
- Simple POJO no class hierarchy



Website

Presentation Layer

JSP

JSF

JAX-RS

WebSockets

Servlet

Business Layer

Facade

EJB

Entities

Data Access Layer

JPA

JCA

JMS

Batch Processing

Crosscutting Layer

Bean validation

CDI

Interceptor

Transaction

Security

JSON-P

Container Services

- Thread safety
- Bandwidth throttled
- Monitoring
- Transactional and pooling

Enterprise Java Beans

Business logic

- @Stateless, @Stateful, @Singleton

Workflow logic

- @MessageDriven

Three Access Modes

- Local (default) – from within the container only
- Remote – by an external application
- Web service – by the client using web service as the protocol
- Session: @Local, @Remote, @LocalBean, @WebService
- Message Driven: @WebService

Management Mode

- Container – container uses convention over configuration
- Bean managed – developer manages behaviour of beans

Service Annotations

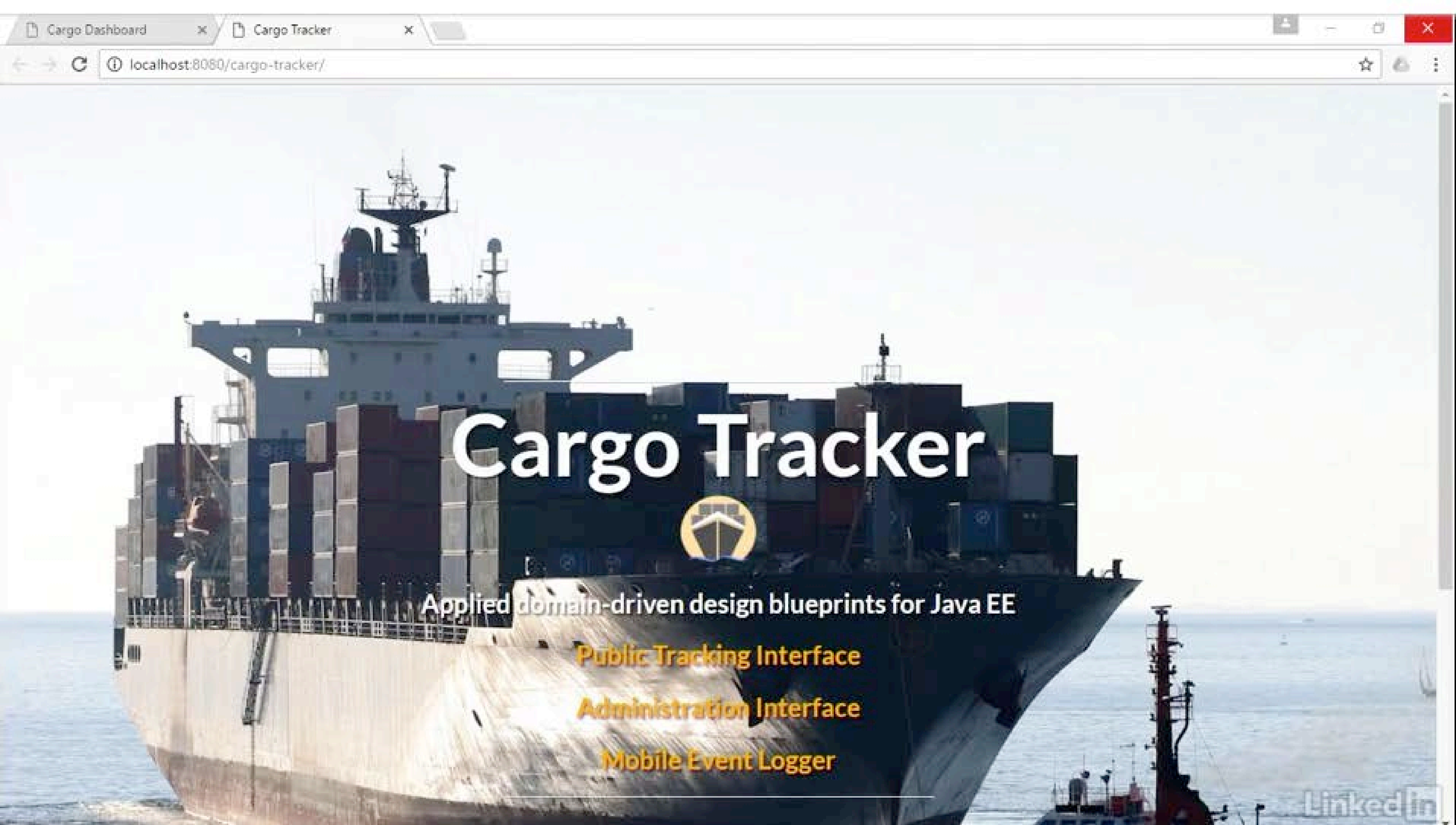
- Concurrency: @Lock, @AccessTimeout

Service Annotations

- Concurrency: @Lock, @AccessTimeout
- Transactions: @Transactional, @TransactionAttribute
- Security: @PermitAll, @RolesAllowed
- Access: @Remote, @Local
- Behaviour: @Schedule, @Asynchronous

EJB3 Introduction

	@Stateless	@Stateful	@Singleton
Remote Client Access	✓	✓	✓
Local Client Access	✓	✓	✓
Concurrent Client Access	X	X	X
Unique Per Client	✓	✓	X
Client-bean Instances	Pooled	1:1	Many:1



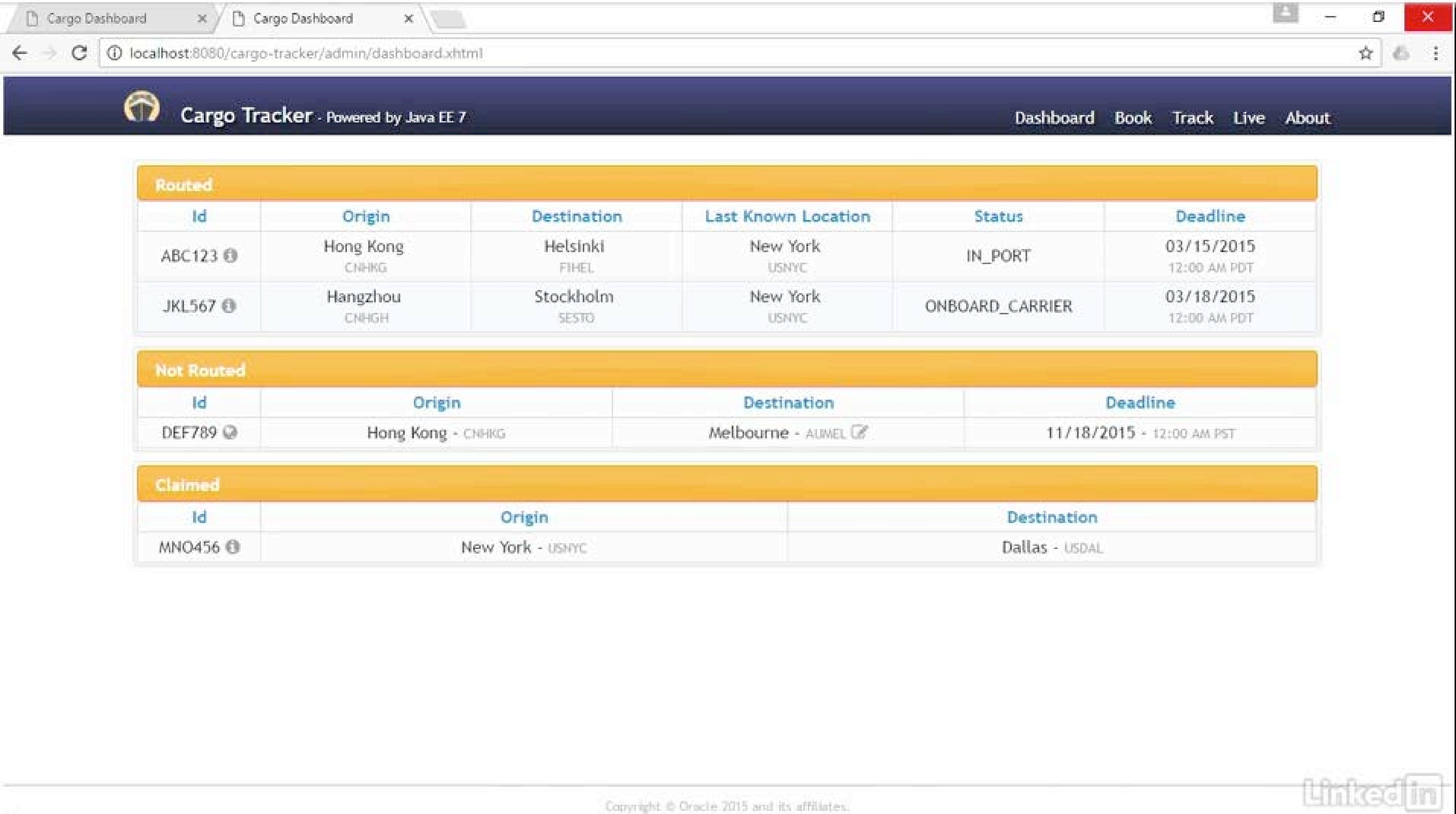
Cargo Tracker

Applied domain-driven design blueprints for Java EE

Public Tracking Interface

Administration Interface

Mobile Event Logger


Cargo Dashboard
Cargo Dashboard
localhost:8080/cargo-tracker/admin/dashboard.xhtml
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

Routed

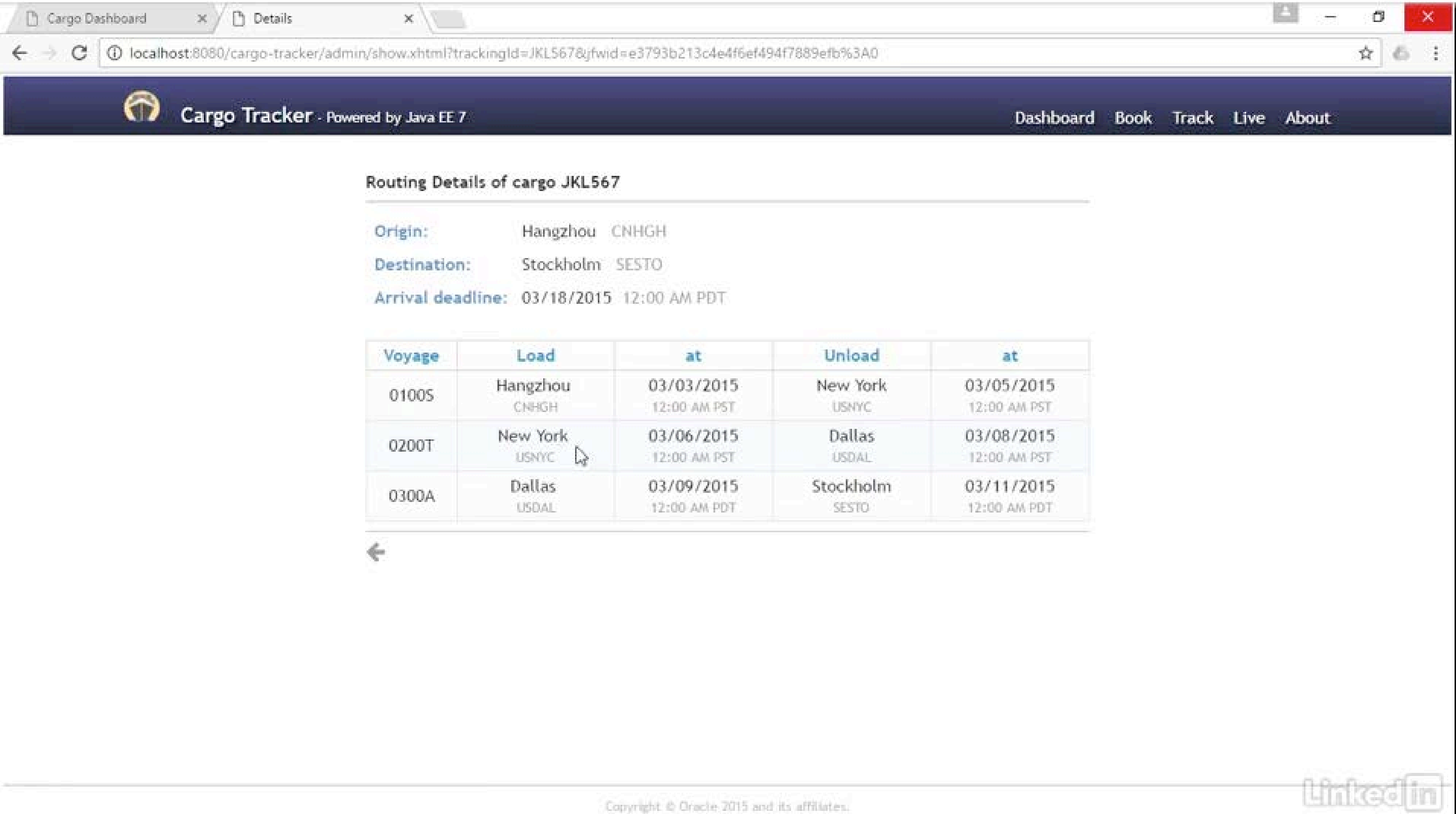
Id	Origin	Destination	Last Known Location	Status	Deadline
ABC123	Hong Kong CNHKG	Helsinki FIHEL	New York USNYC	IN_PORT	03/15/2015 12:00 AM PDT
JKL567	Hangzhou CNHGH	Stockholm SESTO	New York USNYC	ONBOARD_CARRIER	03/18/2015 12:00 AM PDT

Not Routed

Id	Origin	Destination	Deadline
DEF789	Hong Kong - CNHKG	Melbourne - AUMEL	11/18/2015 - 12:00 AM PST

Claimed

Id	Origin	Destination
MNO456	New York - USNYC	Dallas - USDAL

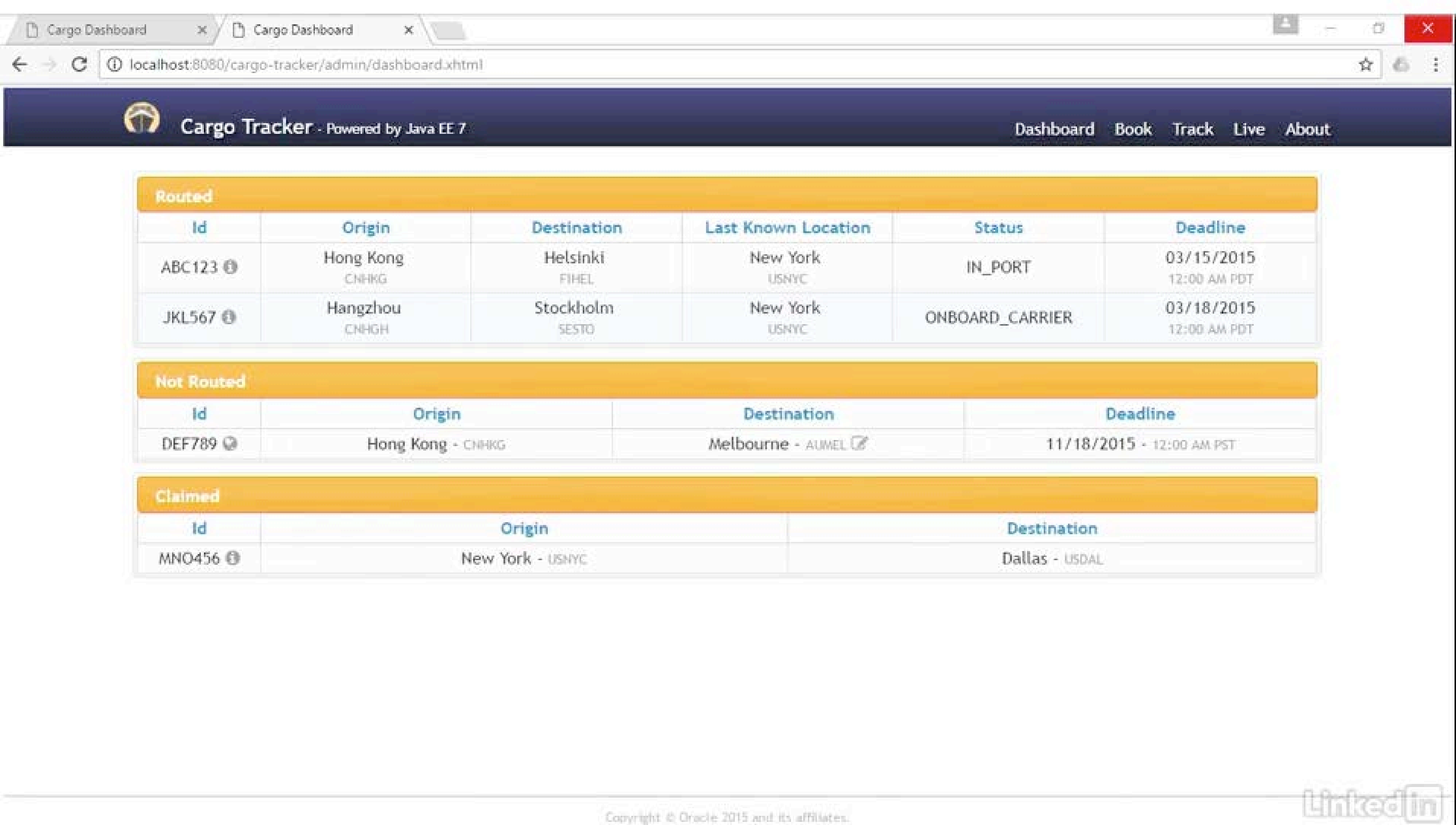


Cargo Dashboard
Details
localhost:8080/cargo-tracker/admin/show.xhtml?trackingId=JKL567&jfwid=e3793b213c4e4f6ef494f7889efb%3A0
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

Routing Details of cargo JKL567

Origin:	Hangzhou CNHGH
Destination:	Stockholm SESTO
Arrival deadline:	03/18/2015 12:00 AM PDT

Voyage	Load	at	Unload	at
0100S	Hangzhou CNHGH	03/03/2015 12:00 AM PST	New York USNYC	03/05/2015 12:00 AM PST
0200T	New York USNYC	03/06/2015 12:00 AM PST	Dallas USDAL	03/08/2015 12:00 AM PST
0300A	Dallas USDAL	03/09/2015 12:00 AM PDT	Stockholm SESTO	03/11/2015 12:00 AM PDT



Cargo Dashboard
Cargo Dashboard
localhost:8080/cargo-tracker/admin/dashboard.xhtml
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

Routed

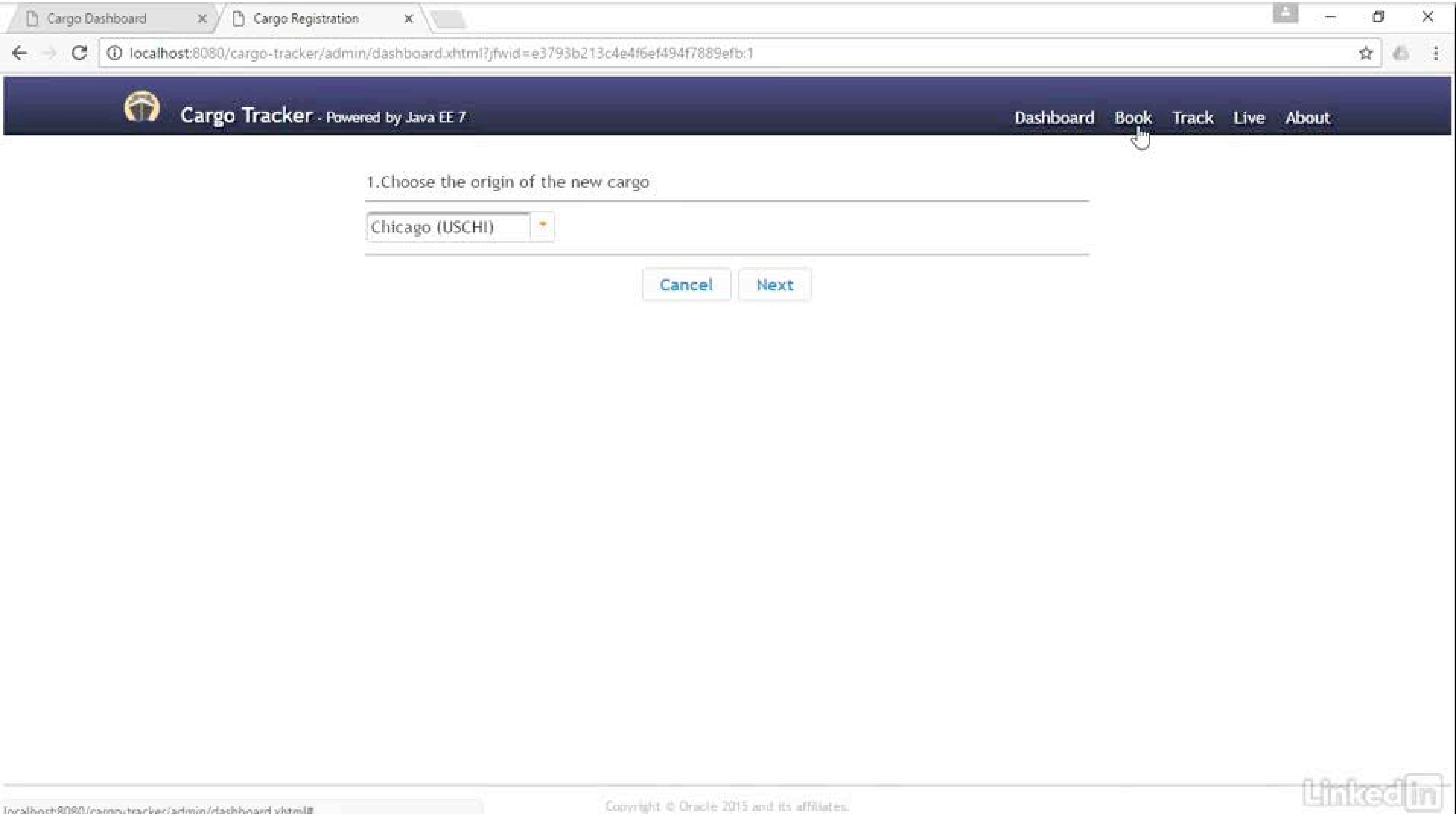
Id	Origin	Destination	Last Known Location	Status	Deadline
ABC123 ⓘ	Hong Kong CNHKG	Helsinki FIHEL	New York USNYC	IN_PORT	03/15/2015 12:00 AM PDT
JKL567 ⓘ	Hangzhou CNHGH	Stockholm SESTO	New York USNYC	ONBOARD_CARRIER	03/18/2015 12:00 AM PDT

Not Routed

Id	Origin	Destination	Deadline
DEF789	Hong Kong - CNHKG	Melbourne - AUMEL	11/18/2015 - 12:00 AM PST

Claimed

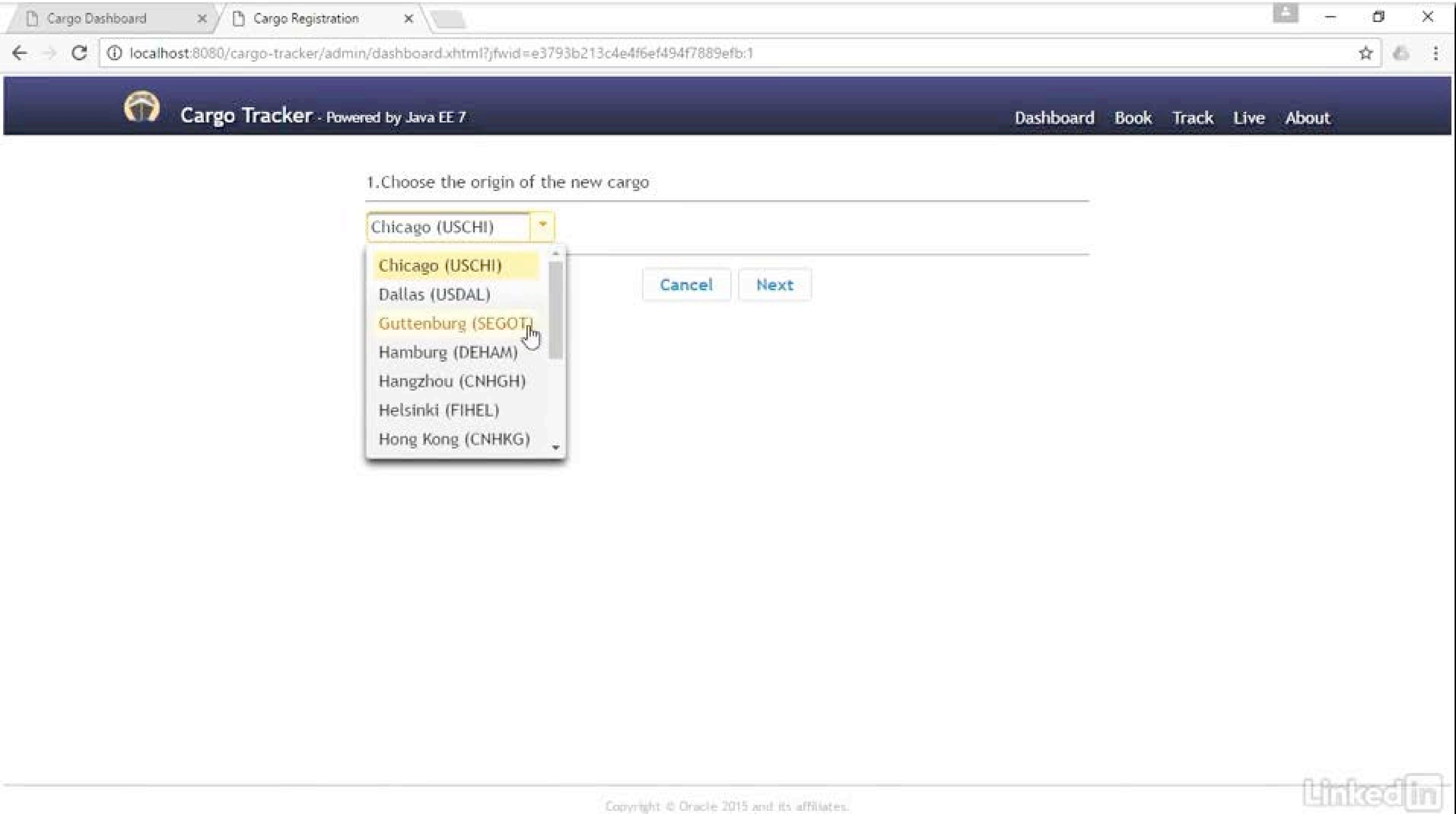
Id	Origin	Destination
MNO456 ⓘ	New York - USNYC	Dallas - USDAL


Cargo Dashboard
Cargo Registration
localhost:8080/cargo-tracker/admin/dashboard.xhtml?jfwid=e3793b213c4e4f6ef494f7889efb:1
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

1.Choose the origin of the new cargo

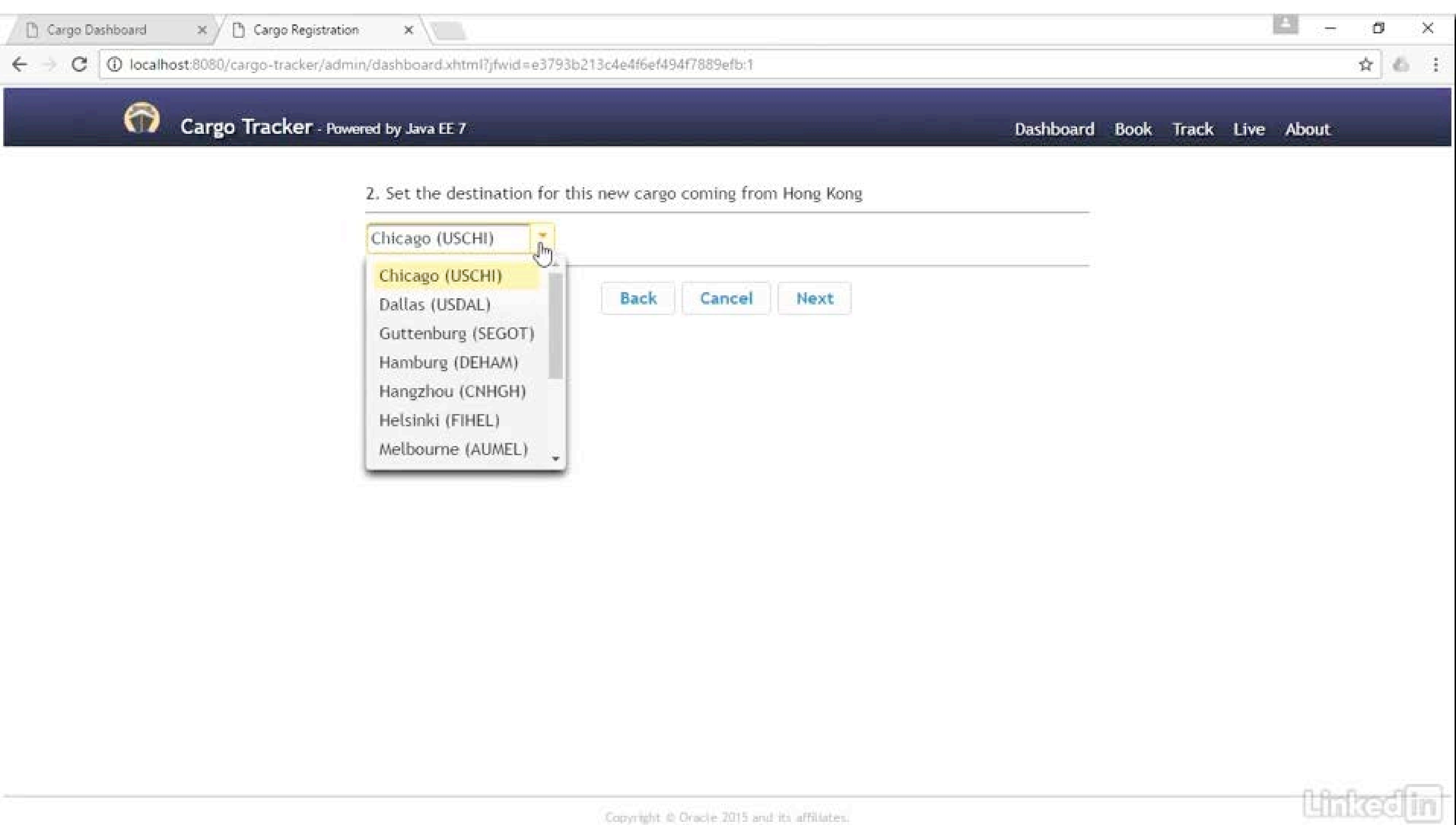
Chicago (USCHI)

Cancel
Next
localhost:8080/cargo-tracker/admin/dashboard.xhtml#


Cargo Dashboard
Cargo Registration
localhost:8080/cargo-tracker/admin/dashboard.xhtml?jfwid=e3793b213c4e4f6ef494f7889efb:1
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

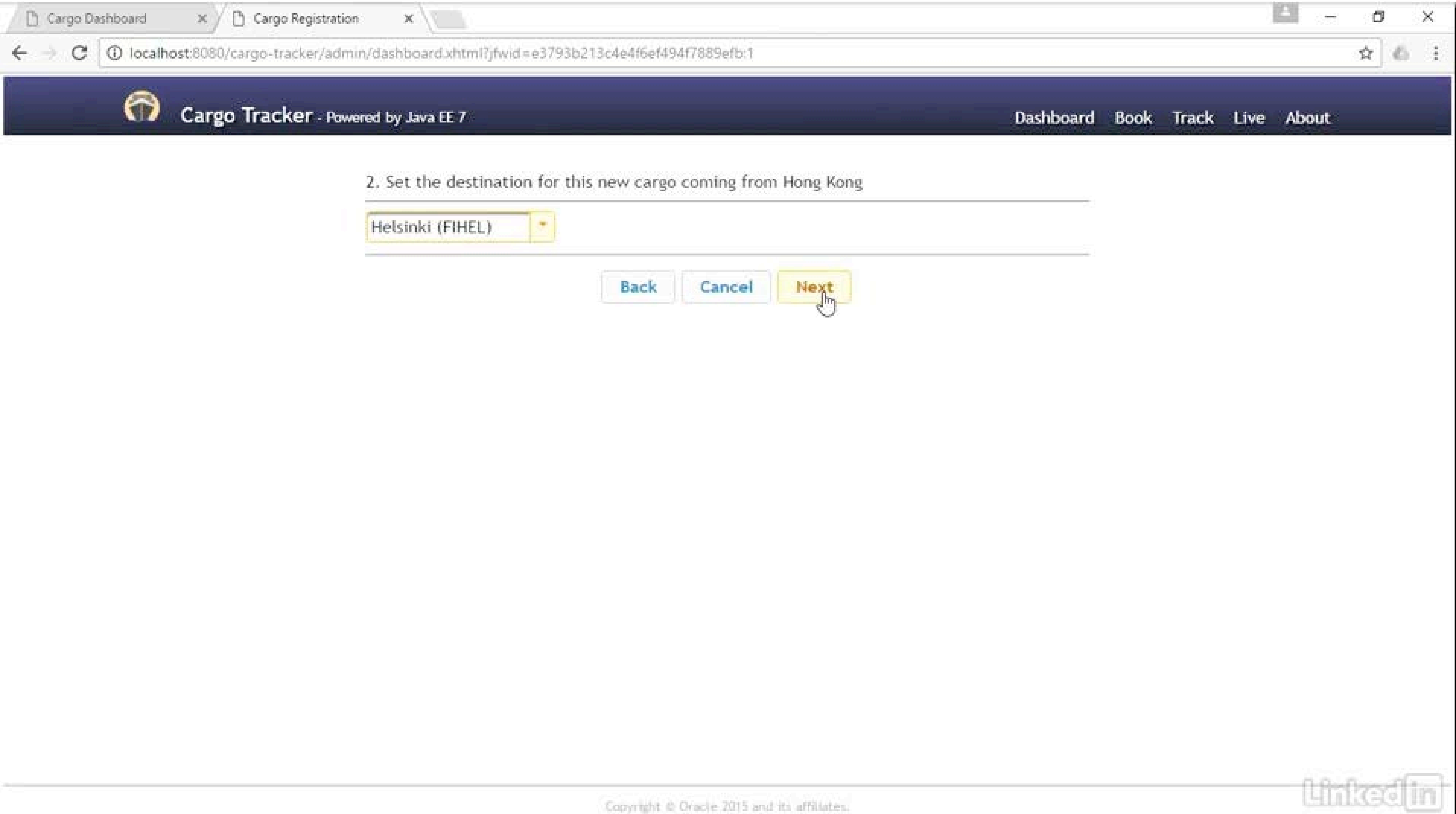
1.Choose the origin of the new cargo

Chicago (USCHI)



2. Set the destination for this new cargo coming from Hong Kong

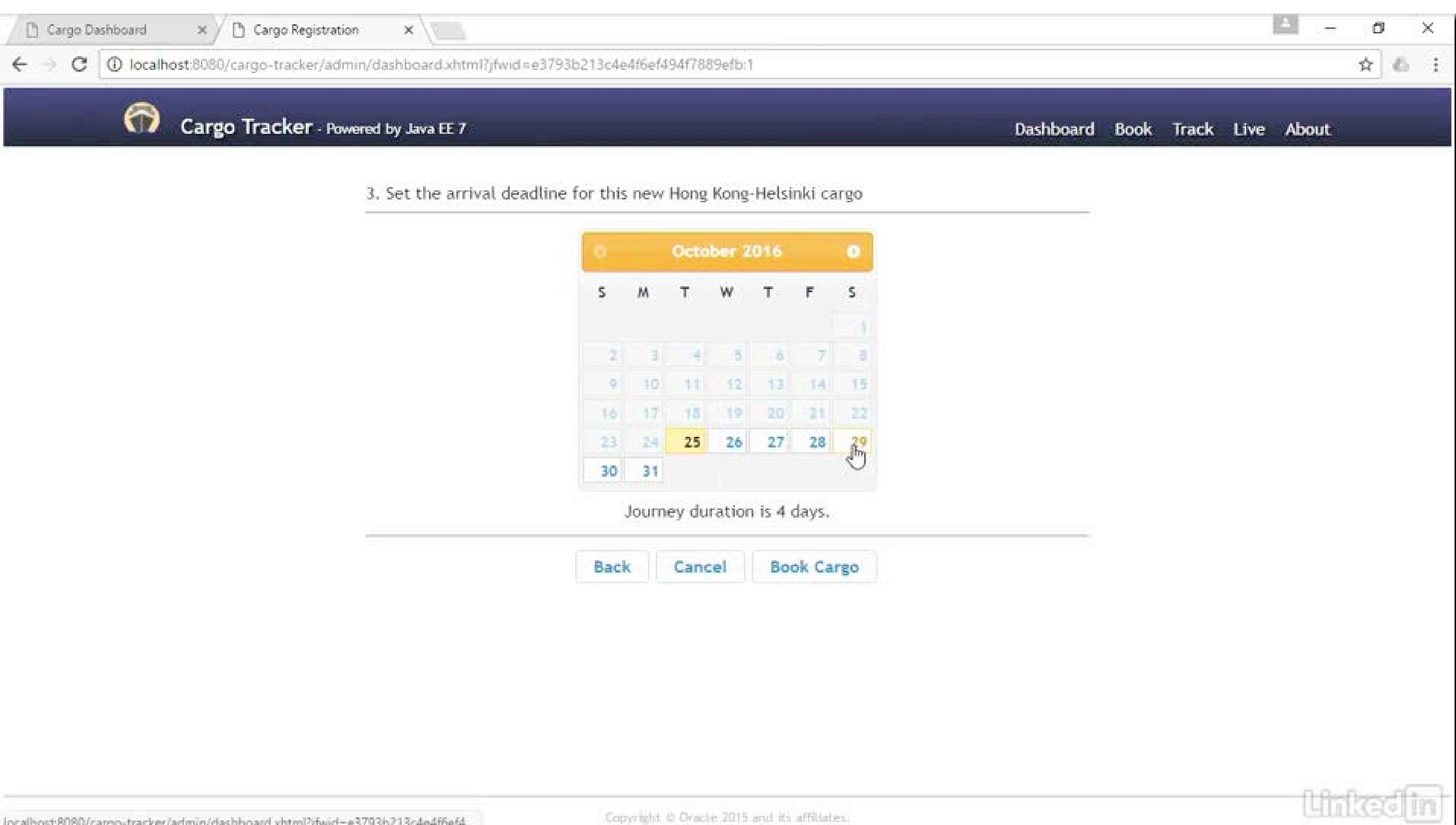
Chicago (USCHI)



Cargo Dashboard
Cargo Registration
localhost:8080/cargo-tracker/admin/dashboard.xhtml?jfwid=e3793b213c4e4f6ef494f7889efb:1
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

2. Set the destination for this new cargo coming from Hong Kong

Helsinki (FIHEL)



Cargo Dashboard
Cargo Registration
localhost:8080/cargo-tracker/admin/dashboard.xhtml?jfwid=e3793b213c4e4f6ef494f7889efb:1
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

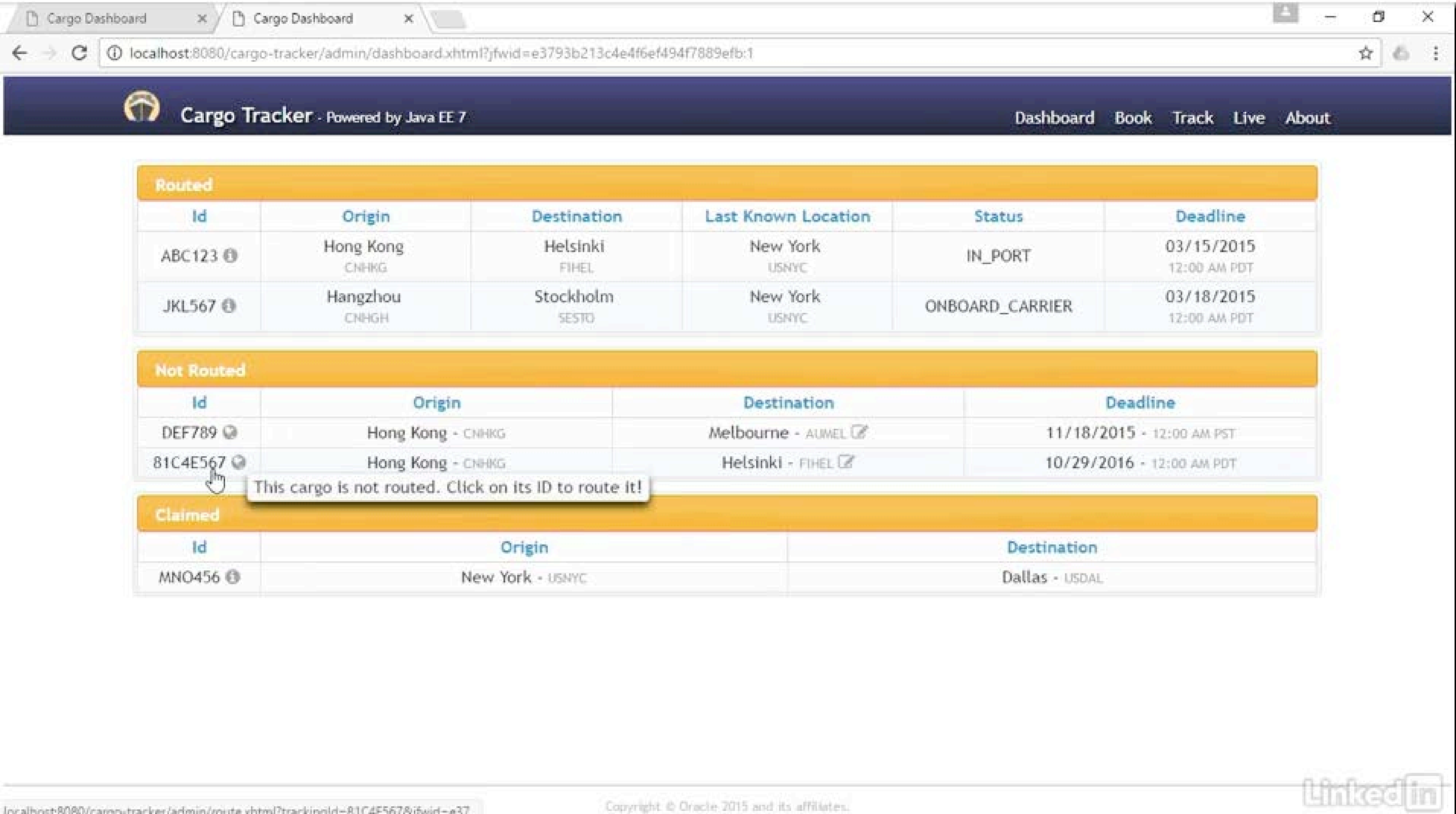
3. Set the arrival deadline for this new Hong Kong-Helsinki cargo

October 2016

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Journey duration is 4 days.

localhost:8080/cargo-tracker/admin/dashboard.xhtml?jfwid=e3793b213c4e4f6ef4



Cargo Dashboard
Cargo Dashboard
localhost:8080/cargo-tracker/admin/dashboard.xhtml?jfwid=e3793b213c4e4f6ef494f7889efb:1
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

Routed

Id	Origin	Destination	Last Known Location	Status	Deadline
ABC123	Hong Kong CNHKG	Helsinki FIHEL	New York USNYC	IN_PORT	03/15/2015 12:00 AM PDT
JKL567	Hangzhou CNHGH	Stockholm SESTO	New York USNYC	ONBOARD_CARRIER	03/18/2015 12:00 AM PDT

Not Routed

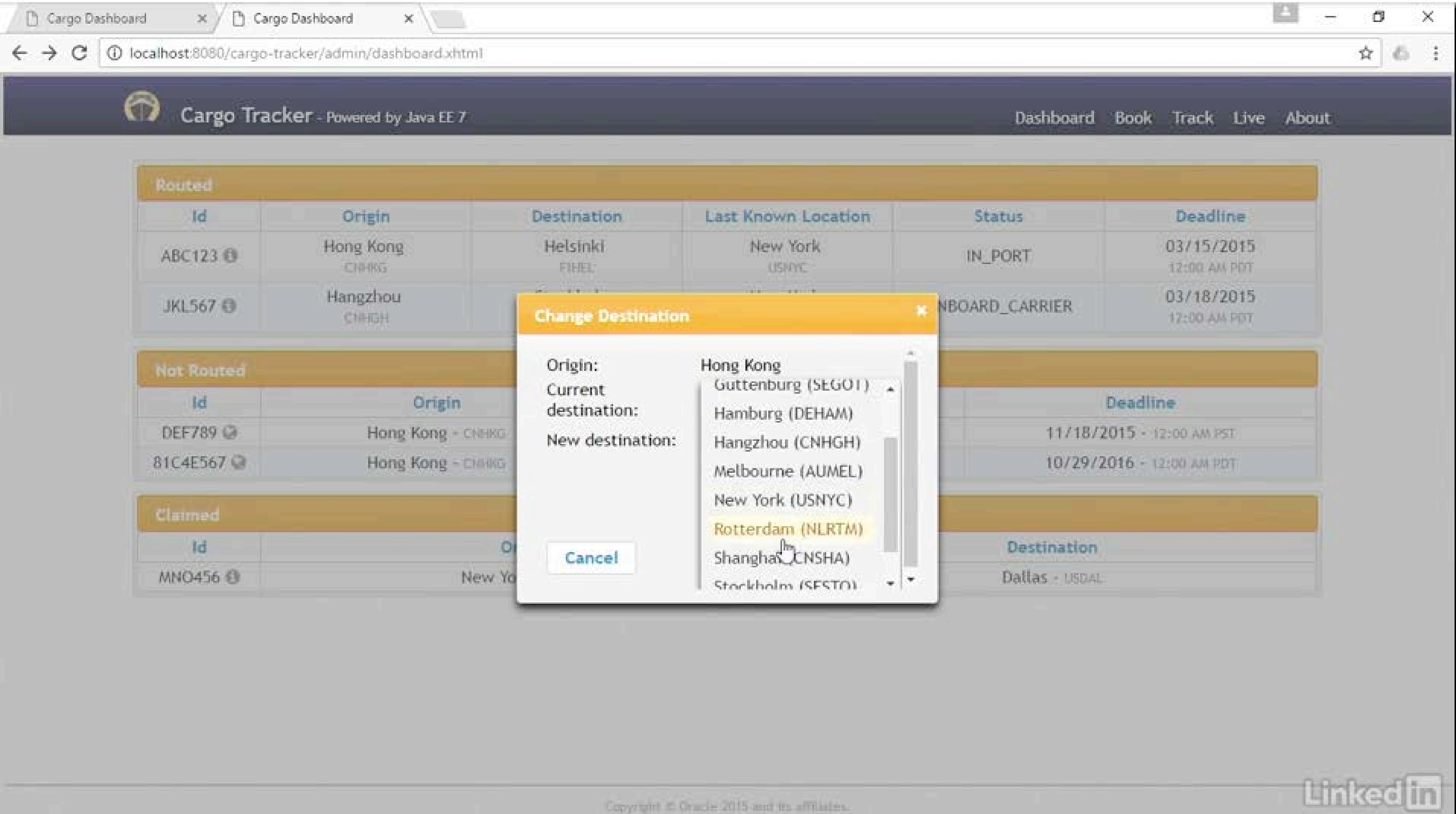
Id	Origin	Destination	Deadline
DEF789	Hong Kong - CNHKG	Melbourne - AUMEL	11/18/2015 - 12:00 AM PST
81C4E567	Hong Kong - CNHKG	Helsinki - FIHEL	10/29/2016 - 12:00 AM PDT

This cargo is not routed. Click on its ID to route it!

Claimed

Id	Origin	Destination
MNO456	New York - USNYC	Dallas - USDAL

localhost:8080/cargo-tracker/admin/route.xhtml?trackingId=81C4E567&jfwid=e37



Cargo Dashboard
Cargo Dashboard
localhost:8080/cargo-tracker/admin/dashboard.xhtml
Cargo Tracker - Powered by Java EE 7
Dashboard
Book
Track
Live
About

Routed

Id	Origin	Destination	Last Known Location	Status	Deadline
ABC123	Hong Kong CNHKG	Helsinki FIHEL	New York USNYC	IN_PORT	03/15/2015 12:00 AM PDT
JKL567	Hangzhou CNHGH			NBOARD_CARRIER	03/18/2015 12:00 AM PDT

Not Routed

Id	Origin	Deadline
DEF789	Hong Kong - CNHKG	11/18/2015 - 12:00 AM PST
81C4E567	Hong Kong - CNHKG	10/29/2016 - 12:00 AM PDT

Claimed

Id	Or	Destination
MNO456	New Yo	Dallas - USDAL

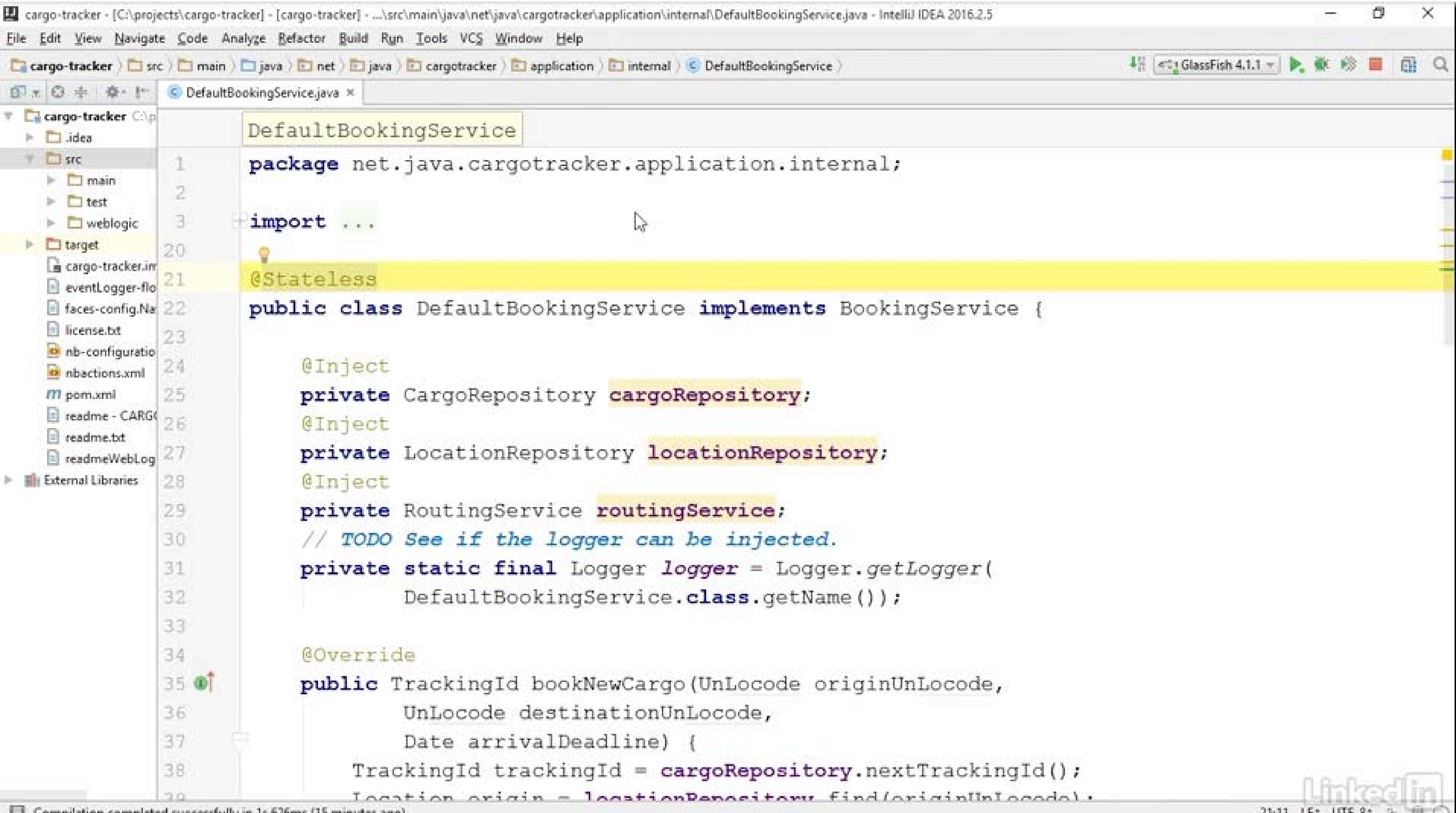
Change Destination

Origin: Hong Kong

Current destination:

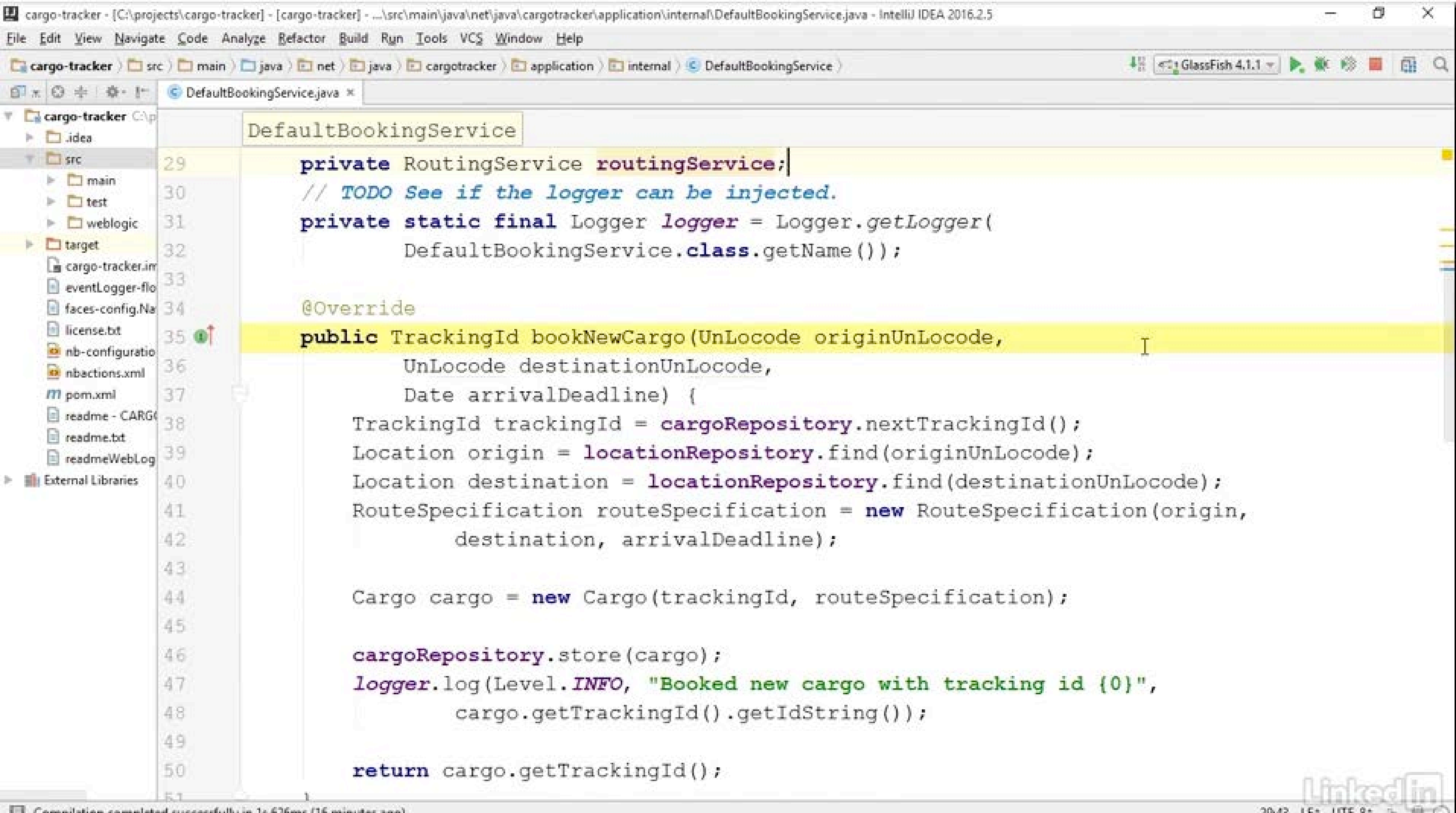
New destination:

Copyright © Oracle 2015 and its affiliates.


cargo-tracker - [C:\projects\cargo-tracker] - [cargo-tracker] - ...\src\main\java\net\java\cargotracker\application\internal\DefaultBookingService.java - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cargo-tracker src main java net java cargotracker application internal DefaultBookingService
GlassFish 4.1.1
DefaultBookingService.java
cargo-tracker C:\p
.idea
src
main
test
weblogic
target
cargo-tracker.im
eventLogger-flo
faces-config.Na
license.txt
nb-configuratio
nbactions.xml
pom.xml
readme - CARG
readme.txt
readmeWebLog
External Libraries
DefaultBookingService

```
package net.java.cargotracker.application.internal;

import ...

@Stateless
public class DefaultBookingService implements BookingService {

    @Inject
    private CargoRepository cargoRepository;
    @Inject
    private LocationRepository locationRepository;
    @Inject
    private RoutingService routingService;
    // TODO See if the logger can be injected.
    private static final Logger logger = Logger.getLogger(
            DefaultBookingService.class.getName());

    @Override
    public TrackingId bookNewCargo(UnLocode originUnLocode,
            UnLocode destinationUnLocode,
            Date arrivalDeadline) {
        TrackingId trackingId = cargoRepository.nextTrackingId();
        Location origin = locationRepository.find(originUnLocode);
```

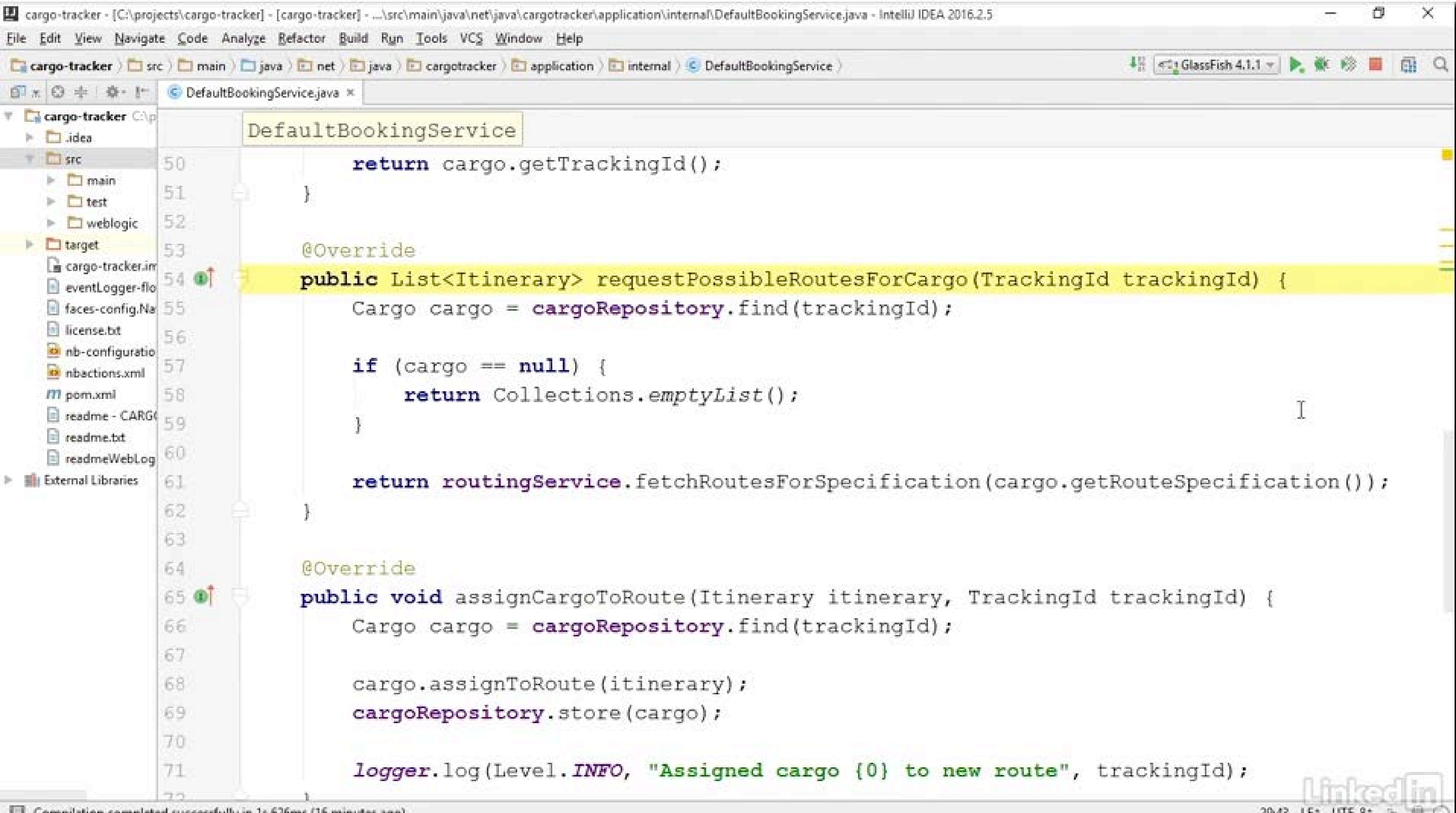
Compilation completed successfully in 1s 626ms (15 minutes ago)



cargo-tracker - [C:\projects\cargo-tracker] - [cargo-tracker] - ...\src\main\java\net\java\cargotracker\application\internal\DefaultBookingService.java - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cargo-tracker src main java net java cargotracker application internal DefaultBookingService
GlassFish 4.1.1
DefaultBookingService.java
cargo-tracker C:\p
.idea
src
main
test
weblogic
target
cargo-tracker.im
eventLogger-flo
faces-config.Na
license.txt
nb-configuratio
nbactions.xml
pom.xml
readme - CARG(
readme.txt
readmeWebLog
External Libraries
DefaultBookingService

```java
    private RoutingService routingService;
    // TODO See if the logger can be injected.
    private static final Logger logger = Logger.getLogger(
            DefaultBookingService.class.getName());

    @Override
    public TrackingId bookNewCargo(UnLocode originUnLocode,
            UnLocode destinationUnLocode,
            Date arrivalDeadline) {
        TrackingId trackingId = cargoRepository.nextTrackingId();
        Location origin = locationRepository.find(originUnLocode);
        Location destination = locationRepository.find(destinationUnLocode);
        RouteSpecification routeSpecification = new RouteSpecification(origin,
                destination, arrivalDeadline);

        Cargo cargo = new Cargo(trackingId, routeSpecification);

        cargoRepository.store(cargo);
        logger.log(Level.INFO, "Booked new cargo with tracking id {0}",
                cargo.getTrackingId().getIdString());

        return cargo.getTrackingId();
    }
```

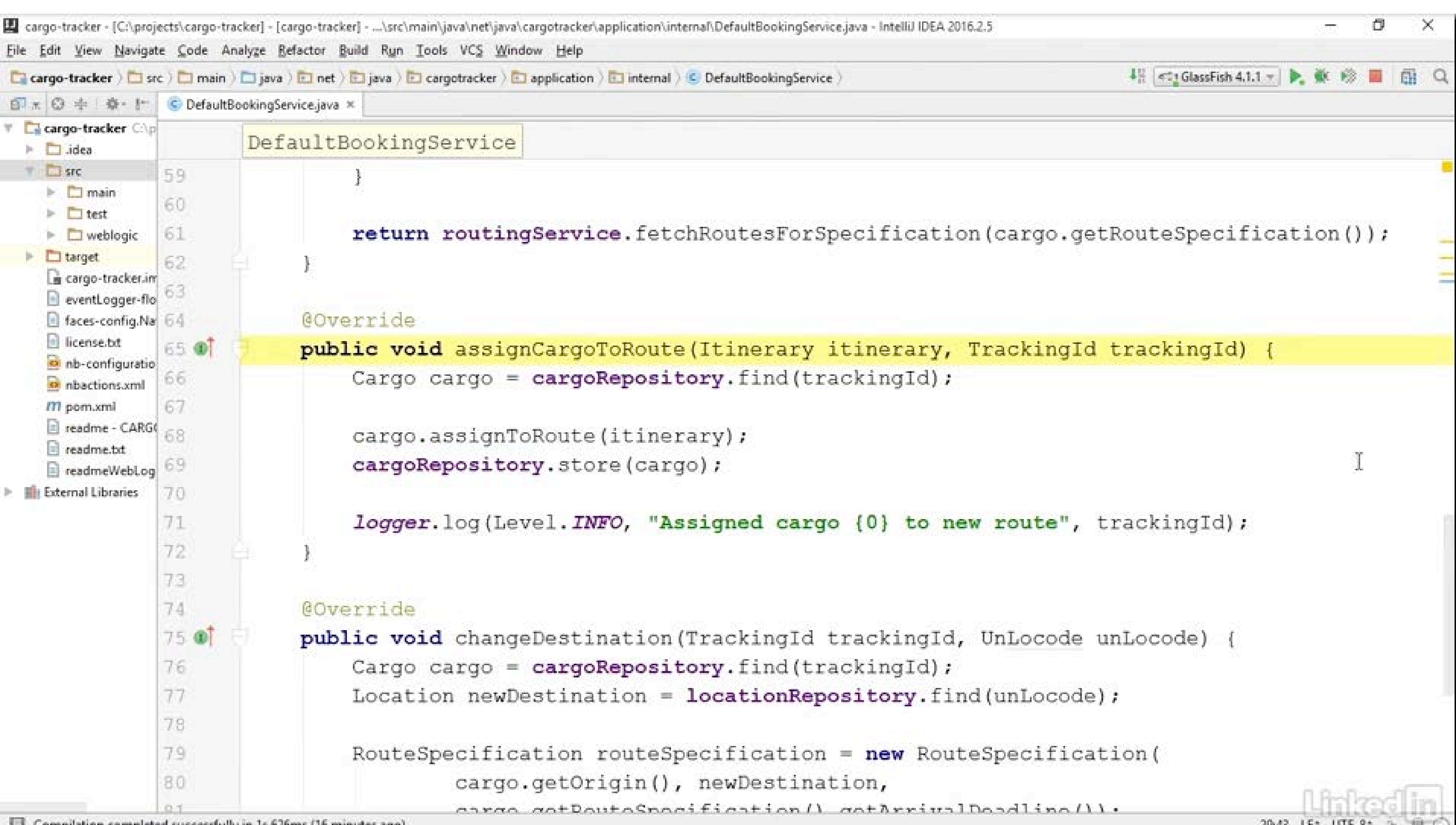
Compilation completed successfully in 1s 626ms (16 minutes ago)



cargo-tracker - [C:\projects\cargo-tracker] - [cargo-tracker] - ...\src\main\java\net\java\cargotracker\application\internal\DefaultBookingService.java - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cargo-tracker src main java net java cargotracker application internal DefaultBookingService
GlassFish 4.1.1
DefaultBookingService.java
cargo-tracker C:\p
.idea
src
main
test
weblogic
target
cargo-tracker.im
eventLogger-flo
faces-config.Na
license.txt
nb-configuratio
nbactions.xml
pom.xml
readme - CARG(
readme.txt
readmeWebLog
External Libraries
DefaultBookingService

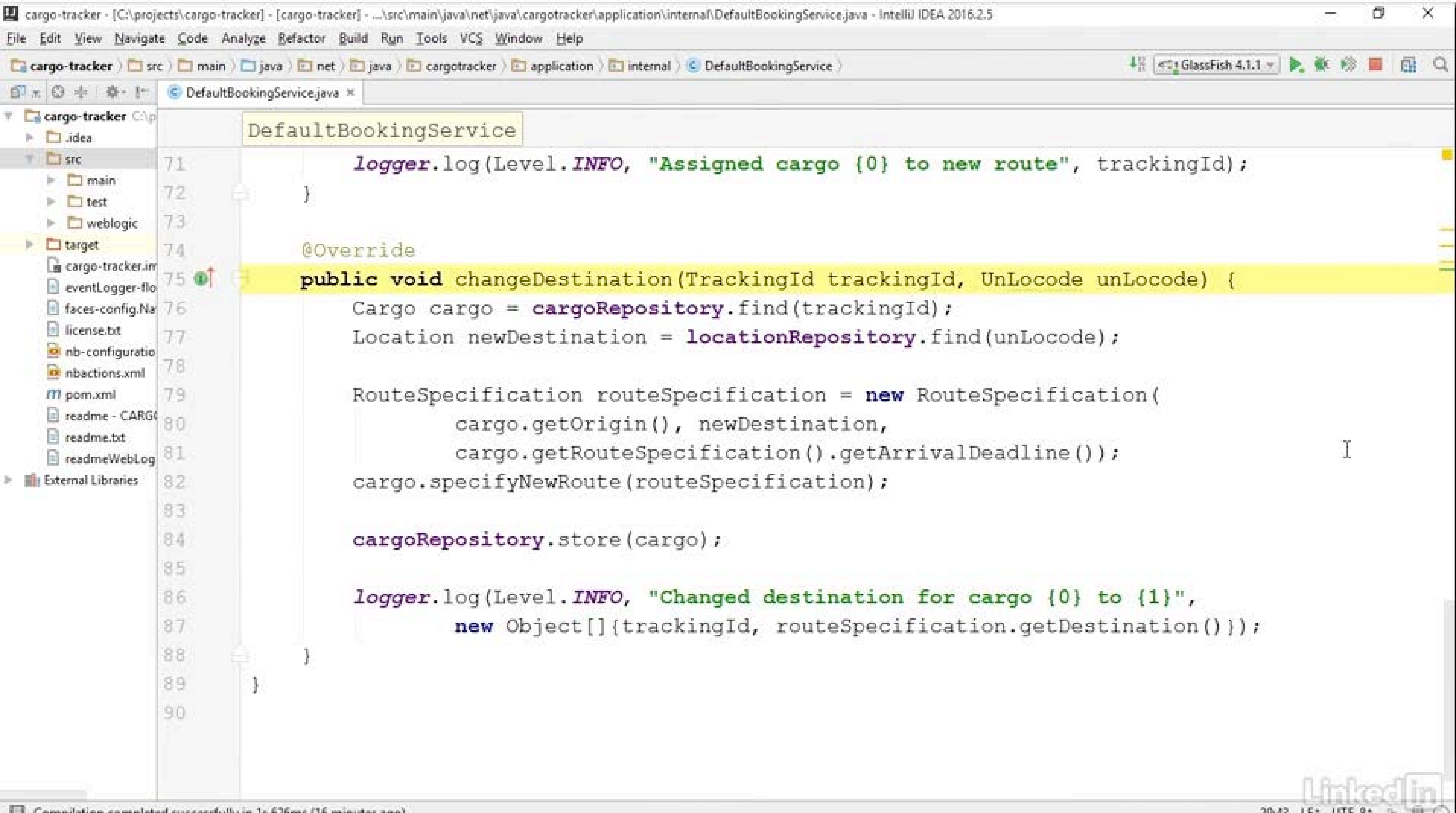
```java
        return cargo.getTrackingId();
    }

    @Override
    public List<Itinerary> requestPossibleRoutesForCargo(TrackingId trackingId) {
        Cargo cargo = cargoRepository.find(trackingId);

        if (cargo == null) {
            return Collections.emptyList();
        }

        return routingService.fetchRoutesForSpecification(cargo.getRouteSpecification());
    }

    @Override
    public void assignCargoToRoute(Itinerary itinerary, TrackingId trackingId) {
        Cargo cargo = cargoRepository.find(trackingId);

        cargo.assignToRoute(itinerary);
        cargoRepository.store(cargo);

        logger.log(Level.INFO, "Assigned cargo {0} to new route", trackingId);
    }
```

Compilation completed successfully in 1s 626ms (16 minutes ago)

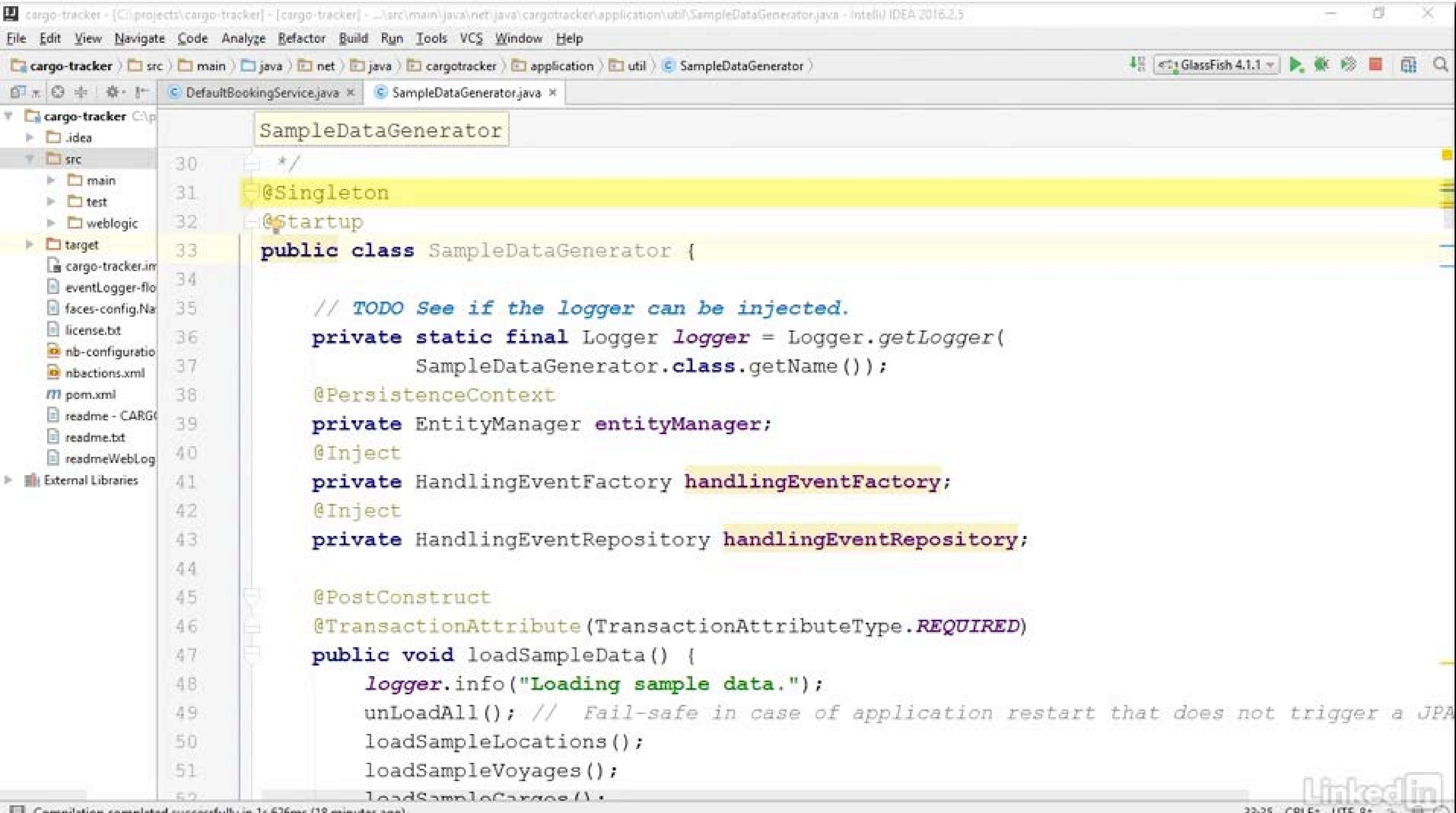

cargo-tracker - [C:\projects\cargo-tracker] - [cargo-tracker] - ...\src\main\java\net\java\cargotracker\application\internal\DefaultBookingService.java - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cargo-tracker src main java net java cargotracker application internal DefaultBookingService
GlassFish 4.1.1
DefaultBookingService.java
cargo-tracker C:\p
.idea
src
main
test
weblogic
target
cargo-tracker.im
eventLogger-flo
faces-config.Na
license.txt
nb-configuratio
nbactions.xml
pom.xml
readme - CARG
readme.txt
readmeWebLog
External Libraries
DefaultBookingService

```java
        }

        return routingService.fetchRoutesForSpecification(cargo.getRouteSpecification());
    }

    @Override
    public void assignCargoToRoute(Itinerary itinerary, TrackingId trackingId) {
        Cargo cargo = cargoRepository.find(trackingId);

        cargo.assignToRoute(itinerary);
        cargoRepository.store(cargo);

        logger.log(Level.INFO, "Assigned cargo {0} to new route", trackingId);
    }

    @Override
    public void changeDestination(TrackingId trackingId, UnLocode unLocode) {
        Cargo cargo = cargoRepository.find(trackingId);
        Location newDestination = locationRepository.find(unLocode);

        RouteSpecification routeSpecification = new RouteSpecification(
                cargo.getOrigin(), newDestination,
                cargo.getRouteSpecification().getArrivalDeadline());
```

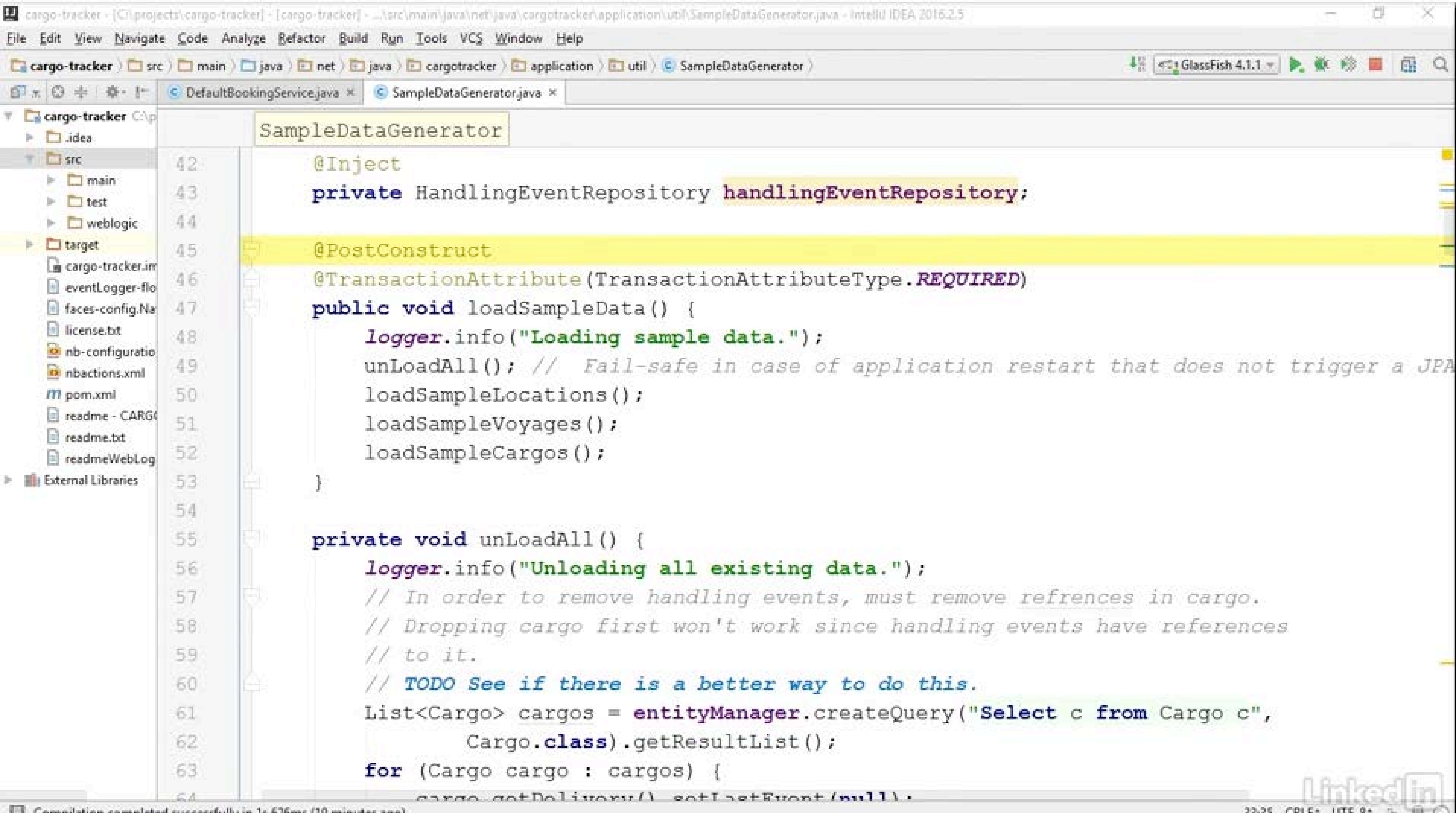
Compilation completed successfully in 1s 626ms (16 minutes ago)


cargo-tracker - [C:\projects\cargo-tracker] - [cargo-tracker] - ...\src\main\java\net\java\cargotracker\application\internal\DefaultBookingService.java - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cargo-tracker src main java net java cargotracker application internal DefaultBookingService
GlassFish 4.1.1
DefaultBookingService.java
cargo-tracker C:\p
.idea
src
main
test
weblogic
target
cargo-tracker.im
eventLogger-flo
faces-config.Na
license.txt
nb-configuratio
nbactions.xml
pom.xml
readme - CARG(
readme.txt
readmeWebLog
External Libraries
DefaultBookingService
Compilation completed successfully in 1s 626ms (16 minutes ago)

```java
        logger.log(Level.INFO, "Assigned cargo {0} to new route", trackingId);
    }

    @Override
    public void changeDestination(TrackingId trackingId, UnLocode unLocode) {
        Cargo cargo = cargoRepository.find(trackingId);
        Location newDestination = locationRepository.find(unLocode);

        RouteSpecification routeSpecification = new RouteSpecification(
                cargo.getOrigin(), newDestination,
                cargo.getRouteSpecification().getArrivalDeadline());
        cargo.specifyNewRoute(routeSpecification);

        cargoRepository.store(cargo);

        logger.log(Level.INFO, "Changed destination for cargo {0} to {1}",
                new Object[]{trackingId, routeSpecification.getDestination()});
    }
}
```



```java
 */
@Singleton
@Startup
public class SampleDataGenerator {

    // TODO See if the logger can be injected.
    private static final Logger logger = Logger.getLogger(
            SampleDataGenerator.class.getName());
    @PersistenceContext
    private EntityManager entityManager;
    @Inject
    private HandlingEventFactory handlingEventFactory;
    @Inject
    private HandlingEventRepository handlingEventRepository;

    @PostConstruct
    @TransactionAttribute(TransactionAttributeType.REQUIRED)
    public void loadSampleData() {
        logger.info("Loading sample data.");
        unLoadAll(); //  Fail-safe in case of application restart that does not trigger a JPA
        loadSampleLocations();
        loadSampleVoyages();
        loadSampleCargos();
```



cargo-tracker - [C:\projects\cargo-tracker] - [cargo-tracker] - ...\src\main\java\net\java\cargotracker\application\util\SampleDataGenerator.java - IntelliJ IDEA 2016.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cargo-tracker src main java net java cargotracker application util SampleDataGenerator
GlassFish 4.1.1
DefaultBookingService.java
SampleDataGenerator.java
cargo-tracker C:\p
.idea
src
main
test
weblogic
target
cargo-tracker.im
eventLogger-flo
faces-config.Na
license.txt
nb-configuratio
nbactions.xml
pom.xml
readme - CARG(
readme.txt
readmeWebLog
External Libraries
SampleDataGenerator

```java
    @Inject
    private HandlingEventRepository handlingEventRepository;

    @PostConstruct
    @TransactionAttribute(TransactionAttributeType.REQUIRED)
    public void loadSampleData() {
        logger.info("Loading sample data.");
        unLoadAll(); //  Fail-safe in case of application restart that does not trigger a JPA
        loadSampleLocations();
        loadSampleVoyages();
        loadSampleCargos();
    }

    private void unLoadAll() {
        logger.info("Unloading all existing data.");
        // In order to remove handling events, must remove refrences in cargo.
        // Dropping cargo first won't work since handling events have references
        // to it.
        // TODO See if there is a better way to do this.
        List<Cargo> cargos = entityManager.createQuery("Select c from Cargo c",
                Cargo.class).getResultList();
        for (Cargo cargo : cargos) {
            cargo.getDelivery().setLastEvent(null);
```

Compilation completed successfully in 1s 626ms (10 minutes ago)