

avigate Search Project Run Window Help

Desktop.java Trainee.java DesktopAllocationServiceImpl.java

```
1 package com.infy.entity;
2
3 import javax.persistence.CascadeType;
4
5
6 @Entity
7 public class Trainee {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private Integer traineeId;
11    private String traineeName;
12    @OneToOne(cascade=CascadeType.ALL)
13    @JoinColumn(name="desktop_id",unique=true)
14    private Desktop desktop;
15    public Integer getTraineeId() {
16        return traineeId;
17    }
18    public void setTraineeId(Integer traineeId) {
19        this.traineeId = traineeId;
20    }
21    public String getTraineeName() {
22        return traineeName;
23    }
24    public void setTraineeName(String traineeName) {
25        this.traineeName = traineeName;
26    }
27    public Desktop getDesktop() {
28    }
29}
```

Navigate Search Project Run Window Help

Desktop.java Trainee.java DesktopAllocationServiceImpl.java

```
1 package com.infy.entity;
2
3 import javax.persistence.Entity;
4 @Entity
5 public class Desktop {
6     @Id
7         private String desktopId;
8         private String desktopMake;
9         private String desktopModel;
10        @Enumerated(EnumType.STRING)
11        private DesktopStatus desktopStatus;
12        public String getDesktopId() {
13            return desktopId;
14        }
15        public void setDesktopId(String desktopId) {
16            this.desktopId = desktopId;
17        }
18        public String getDesktopMake() {
19            return desktopMake;
20        }
21        public void setDesktopMake(String desktopMake) {
22            this.desktopMake = desktopMake;
23        }
24        public String getDesktopModel() {
25            return desktopModel;
26        }
27        public void setDesktopModel(String desktopModel) {
28    }
```

```
1 package com.infy.service;
2
3
4 import java.util.Optional;
5 @Service(value="desktopAllocationService")
6 @Transactional
7 public class DesktopAllocationServiceImpl implements DesktopAllocationService {
8
9     @Autowired
10    private TraineeRepository traineeRepository;
11
12    @Autowired
13    private DesktopRepository desktopRepository;
14
15    /**
16     This method calls findById method of traineeRepository sending traineeId received
17     @param Integer traineeId
18     @return TraineeDTO object object populated from entity object returned by findById
19     @throws Service.TRAINEE_NOT_FOUND exception if object returned by findById method
20     */
21    @Override
22    public TraineeDTO getTraineeDetails(Integer traineeId) throws InfyTrainingException {
23        Optional<Trainee> optional=traineeRepository.findById(traineeId);
24        Trainee trainee=optional.orElseThrow(()-> new InfyTrainingException("Service.
25        TraineeDTO traineeDTO=new TraineeDTO();
26        traineeDTO.setTraineeId(trainee.getTraineeId());
27        traineeDTO.setTraineeName(trainee.getTraineeName());
```

nfy/service/DesktopAllocationServiceImpl.java - Eclipse IDE

File Run Window Help

Trainee.java DesktopAllocationServiceImpl.java

```
@throws Service.TRAINEE_NOT_FOUND exception if object returned by findById method of traineeRepository
*/
@Override
public TraineeDTO getTraineeDetails(Integer traineeId) throws InfyTrainingException {
    Optional<Trainee> optional=traineeRepository.findById(traineeId);
    Trainee trainee=optional.orElseThrow(()-> new InfyTrainingException("Service.TRAINEE_NOT_FOUND"));
    TraineeDTO traineeDTO=new TraineeDTO();
    traineeDTO.setTraineeId(trainee.getTraineeId());
    traineeDTO.setTraineeName(trainee.getTraineeName());
    DesktopDTO dDTO=new DesktopDTO();
    dDTO.setDesktopId(trainee.getDesktop().getDesktopId());
    dDTO.setDesktopMake(trainee.getDesktop().getDesktopMake());
    dDTO.setDesktopModel(trainee.getDesktop().getDesktopModel());
    dDTO.setDesktopStatus(trainee.getDesktop().getDesktopStatus());
    traineeDTO.setDesktop(dDTO);
    return traineeDTO;
}

/**
 * This method calls findById method of desktopRepository sending desktopId received in parameter
 * @param String desptktopId
 * @return DesktopDTO object populated from entity object returned by findById method of desktopRepository
 * @throws Service.DESKTOP_NOT_FOUND exception if object returned by findById method of desktopRepository
 */
@Override
public DesktopDTO getDesktopDetails(String desktopId) throws InfyTrainingException {
    Optional<Desktop> optional=desktopRepository.findById(desktopId);
    Desktop desktop=optional.orElseThrow(()-> new InfyTrainingException("Service.DESKTOP_NOT_FOUND"));
    DesktopDTO dDTO=new DesktopDTO();
    dDTO.setDesktopId(desktop.getDesktopId());
```

inee/src/main/java/com/infy/service/DesktopAllocationServiceImpl.java - Eclipse IDE

avigate Search Project Run Window Help

Desktop.java Trainee.java DesktopAllocationServiceImpl.java

```
55     @return DesktopDTO object populated from entity object returned by findById method of desktopRepository
56     @throws Service.DESKTOP_NOT_FOUND exception if object returned by findById method of desktopRepository
57     */
58     @Override
59     public DesktopDTO getDesktopDetails(String desktopId) throws InfyTrainingException {
60         Optional<Desktop> optional=desktopRepository.findById(desktopId);
61         Desktop desktop=optional.orElseThrow(()-> new InfyTrainingException("Service.DESKTOP_NOT_FOUND"));
62         DesktopDTO dDTO=new DesktopDTO();
63         dDTO.setDesktopId(desktop.getDesktopId());
64         dDTO.setDesktopMake(desktop.getDesktopMake());
65         dDTO.setDesktopModel(desktop.getDesktopModel());
66         dDTO.setDesktopStatus(desktop.getDesktopStatus());
67         return dDTO;
68     }
69
70     /**
71      This method calls save method of traineeRespository sending Trainee entity object populated by trainee
72      @param TraineeDTO object
73      @return Integer traineeId after calling save method of traineeRespository
74      */
75     @Override
76     public Integer addTrainee(TraineeDTO traineeDTO) throws InfyTrainingException {
77         Trainee t=new Trainee();
78         t.setTraineeName(traineeDTO.getTraineeName());
79         traineeRespository.save(t);
80         return t.getTraineeId();
81     }
82
83 }
```

```
Desktop.java Trainee.java DesktopAllocationServiceImpl.java
94     -Service.DESKTOP_NOT_FOUND exception if object returned by traineeRepository method of desktopRepository is null
95     -Service.DESKTOP_ALREADY_ALLOCATED exception if desktop is already allocated to some other trainee
96     */
97     @Override
98     public void allocateDesktop(Integer traineeId, String desktopId) throws InfyTrainingException {
99         Optional<Trainee> optional=traineeRepository.findById(traineeId);
100        Trainee trainee=optional.orElseThrow(()->new InfyTrainingException("Service.TRAINEE_NOT_FOUND"));
101        if(trainee.getDesktop()!=null) {
102            throw new InfyTrainingException("Service.TRAINEE_DESKTOP_FOUND");
103        }
104        Optional<Desktop> optional1=desktopRepository.findById(desktopId);
105        Desktop desktop=optional1.orElseThrow(()->new InfyTrainingException("Service.TRAINEE_NOT_FOUND"));
106
107        if(desktop.getDesktopStatus().equals(DesktopStatus.ALLOCATED)) {
108            throw new InfyTrainingException("Service.DESKTOP_ALREADY_ALLOCATED");
109        }
110
111        desktop.setDesktopStatus(DesktopStatus.ALLOCATED);
112        trainee.setDesktop(desktop);
113
114    }
115    /**
116     * This method calls findById method of traineeRepository sending traineeId received in parameter, then
117     * it deallocates the desktop allocated to the trainee.
118     * @param Integer traineeId
119     * @throws
120     * - Service.TRAINEE_NOT_FOUND exception if object returned by findById method of traineeRepository is null
121     * - Service.DESKTOP_NOT_ALLOCATED exception if no desktop is allocated to the trainee
122     */
123     @Override
```

```
121     - Service.TRAINEE_NOT_FOUND exception if object returned by findById method of traineeRepository
122     - Service.DESKTOP_NOT_ALLOCATED exception if no desktop is allocated to the trainee
123     */
124     @Override
125     public void deallocateDesktop(Integer traineeId) throws InfyTrainingException {
126         Optional<Trainee> optional=traineeRepository.findById(traineeId);
127         Trainee t=optional.orElseThrow(()->new InfyTrainingException("Service.TRAINEE_NOT_FOUND"));
128         if(t.getDesktop()==null) {
129             throw new InfyTrainingException("Service.DESKTOP_NOT_ALLOCATED");
130         }
131         t.getDesktop().setDesktopStatus(DesktopStatus.AVAILABLE);
132         t.setDesktop(null);
133     }
134     /**
135      This method first calls findById method of traineeRepository sending traineeId received in parameter
136      then calls delete method of traineeRepository sending traineeId received in parameter
137      @param Integer traineeId
138      @throws Service.TRAINEE_NOT_FOUND exception if Trainee object returned by findById method of trainee
139      */
140     @Override
141     public void deleteTrainee(Integer traineeId) throws InfyTrainingException {
142         Optional<Trainee> optional=traineeRepository.findById(traineeId);
143         Trainee t=optional.orElseThrow(()->new InfyTrainingException("Service.TRAINEE_NOT_FOUND"));
144         t.getDesktop().setDesktopStatus(DesktopStatus.AVAILABLE);
145         t.setDesktop(null);
146         traineeRepository.delete(t);
147     }
148 }
149 }
150 }
```