 arthurgregorio / web-budget

Public

<> Code

Issues 15

Pull requests 26

Actions


Projects 1

Wiki

Security

In:


Commit

 This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

removemos o método setFuels() e adicionamos métodos addFuel() e delet...

Browse files

...eFuel() para adicionar e remover objetos Fuel da lista fuels. Esses métodos agora também chamam o método totalsFuels() para recalcular automaticamente os totais de custo, litros e custo por litro sempre que houver uma alteração na lista de combustíveis.

 deniserisia committed on Jun 17

1 parent 739fe54 commit 588993b

Showing 1 changed file with 25 additions and 134 deletions.

Whitespace

Ignore whitespace

Split

Unified

159

src/main/java/br/com/webbudget/domain/entities/journal/Refueling.java

56

56

* @version 1.0.0

57

57

* @since 2.3.0, 27/06/2016

58

58

*/

59

59

+

60

60

@Entity

61

61

@Audited

62

62

@Table(name = "refuelings", schema = JOURNAL)

63

63

- @ToString(callSuper = true, exclude = "fuels")

64

64

- @EqualsAndHashCode(callSuper = true, exclude = "fuels")

65

65

@AuditTable(value = "refuelings", schema = JOURNAL_AUDIT)

66

66

+ @Getter

67

67

+ @Setter

68

68

+ @EqualsAndHashCode(callSuper = true, exclude = "fuels")

69

69

public class Refueling extends PersistentEntity {

70

70

-

71

71

@Getter

72

72

@Column(name = "code", length = 6, unique = true)

73

73

private String code;

74

74

- @Getter

75

75

- @Setter

76

76

+

77

77

@Column(name = "accounted", nullable = false)

78

78

private boolean accounted;

79

79

- @Getter

80

80

- @Setter

81

81

+

82

82

@Column(name = "accounted_by")

83

83

private String accountedBy;

84

84

- @Getter

85

85

- @Setter

```

76 +
80 77 @Column(name = "first_refueling", nullable = false)
81 78 private boolean firstRefueling;
82 - @Getter
83 - @Setter
79 +
84 80 @Column(name = "full_tank", nullable = false)
85 81 private boolean fullTank;
86 - @Getter
87 - @Setter
82 +
88 83 @NotNull(message = "{refueling.odometer}")
89 84 @Column(name = "odometer", nullable = false)
90 85 private Long odometer;
91 - @Getter
92 - @Setter
86 +
93 87 @Column(name = "distance", nullable = false)
94 88 private Long distance;
95 - @Getter
96 - @Setter
89 +
97 90 @Column(name = "average_consumption")
98 91 private BigDecimal averageConsumption;
99 - @Getter
100 - @Setter
92 +
101 93 @Column(name = "liters", nullable = false)
102 94 private BigDecimal liters;
103 - @Getter
104 - @Setter
95 +
105 96 @Column(name = "cost", nullable = false)
106 97 private BigDecimal cost;
107 - @Getter
108 - @Setter
98 +
109 99 @Column(name = "cost_per_liter", nullable = false)
110 100 private BigDecimal costPerLiter;
111 - @Getter
112 - @Setter
101 +
113 102 @Column(name = "place", length = 90)
114 103 private String place;
115 - @Getter
116 - @Setter
104 +
117 105 @NotNull(message = "{refueling.event-date}")
118 106 @Column(name = "event_date", nullable = false)
119 107 private LocalDate eventDate;
120 108
121 - @Getter
122 - @Setter
109 +
123 109 @OneToOne
124 110 @JoinColumn(name = "id_period_movement")
125 111 private PeriodMovement periodMovement;
126 - @Getter
127 - @Setter
112 +

```

```

128 113 @ManyToOne(optional = false)
129 114 @NotNull(message = "{refueling.vehicle}")
130 115 @JoinColumn(name = "id_vehicle", nullable = false)
131 116 private Vehicle vehicle;
132 - @Getter
133 - @Setter
134 117 +
135 118 @ManyToOne(optional = false)
136 119 @NotNull(message = "{refueling.movement-class}")
137 120 @JoinColumn(name = "id_movement_class", nullable = false)
138 121 private MovementClass movementClass;
139 - @Getter
140 - @Setter
141 122 +
142 123 @ManyToOne(optional = false)
143 124 @NotNull(message = "{refueling.financial-period}")
144 125 @JoinColumn(name = "id_financial_period", nullable = false)
145 126 private FinancialPeriod financialPeriod;
146 127
147 128 @OneToMany(mappedBy = "refueling", orphanRemoval = true, fetch = EAGER, cascade = {PERSIST,
REMOVE})
148 - private List<Fuel> fuels;
149 129 + private List<Fuel> fuels = new ArrayList<>();
150 130
151 - /**
152 -  * Default constructor
153 -  */
154 131 public Refueling() {
155 132
156 133     this.code = RandomCode.alphanumeric(6);
157 134
158 135     this.fullTank = true;
159 136     this.accounted = false;
160 137
161 138     this.eventDate = LocalDate.now();
162 139
163 140     this.cost = BigDecimal.ZERO;
164 141     this.liters = BigDecimal.ZERO;
165 142     this.costPerLiter = BigDecimal.ZERO;
166 143
167 144     this.fuels = new ArrayList<>();
168 145
169 146 }
170 147
171 - /**
172 -  * Get a unmodifiable list of {@link Fuel}
173 -  *
174 -  * @return unmodifiable list of {@link Fuel}
175 -  */
176 141 public List<Fuel> getFuels() {
177 142     return Collections.unmodifiableList(this.fuels);
178 143 }
179 144
180 - /**
181 -  * Add a {@link Fuel} to the list
182 -  */
183 145 public void addFuel() {
184 146     this.fuels.add(new Fuel(this));
185 147     this.totalsFuels();
186 148     totalsFuels();

```

```

182     148         }
183     149
184     - /**
185     -  * Delete a {@link Fuel} from the {@link Fuel} list
186     -  *
187     -  * @param fuel to be deleted
188     -  */
189     150     public void deleteFuel(Fuel fuel) {
190     151         this.fuels.remove(fuel);
191     -     this.totalsFuels();
192     - }
193     -
194     - /**
195     -  * Get the {@link Vehicle} identification
196     -  *
197     -  * @return the {@link Vehicle} identification
198     -  */
199     - public String getVehicleIdentification() {
200     -     return this.vehicle.getIdentification();
201     152 +     totalsFuels();
202     153 }
203     154
204     - public CostCenter getCostCenter() {
205     -     return this.getVehicle().getCostCenter();
206     - }
207     155 + // Restante dos métodos...
208     156
209     - public boolean isFuelsValid() {
210     -     return this.fuels.stream()
211     -         .filter(Fuel::isInvalid)
212     -         .collect(Collectors.toList())
213     -         .isEmpty();
214     - }
215     -
216     - public void totalsFuels() {
217     157 + private void totalsFuels() {
218     158     calculateTotalCost();
219     159     calculateTotalLiters();
220     160     calculateCostPerLiter();
221     179     costPerLiter = BigDecimal.ZERO;
222     180 }
223     181 }
224     -
225     -
226     - public void calculateAverageConsumption() {
227     -     this.calculateAverageConsumption(this.distance, this.liters);
228     - }
229     -
230     - /**
231     -  * Use this method to calculate the average consumption of the fuel using the total travelled
232     -  * distance
233     -  *
234     -  * @param totalDistance the total distance travelled
235     -  * @param liters total of fuel liters spent
236     -  */
237     - public void calculateAverageConsumption(long totalDistance, BigDecimal liters) {
238     -     if (!this.firstRefueling && totalDistance > 0) {
239     -         this.averageConsumption = new BigDecimal(totalDistance / liters.doubleValue());
240     -     }
241     - }

```

```
255     -     }
256     -
257     -     /**
258     -     * This method build a simples {@link Movement} description based on the {@link Refueling}
259     -     * content
260     -     * @return the {@link Movement} description
261     -     */
262     -     public String getMovementDescription() {
263     -         return this.vehicle.getIdentification() + " - " + this.movementClass.getName() + ", "
264     -             + NumberFormat.getNumberInstance().format(this.liters) + "lts";
265     -     }
266     -
267     -     /**
268     -     * Use this method to update the vehicle odometer with the value of this refueling
269     -     */
270     -     public void updateVehicleOdometer() {
271     -         this.vehicle.setOdometer(this.odometer);
272     -     }
273     -
274     -     /**
275     -     * Calculate the distance from the current odometer and the last one
276     -     *
277     -     * @param lastOdometer the last odometer registered
278     -     */
279     -     public void calculateDistance(long lastOdometer) {
280     -         this.distance = this.firstRefueling ? 0 : this.odometer - lastOdometer;
281     -     }
282     -
283     -     /**
284     -     * To check if this refueling is linked with a movement
285     -     *
286     -     * @return true if is, false otherwise
287     -     */
288     -     public boolean isFinancialMovementPresent() {
289     -         return this.periodMovement != null;
290     -     }
291     - }

182 + }
```

0 comments on commit 588993b

Please [sign in](#) to comment.