


Commit

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

Remoção de comentários de direitos autorais duplicados e desnecessários.

Browse files

 deniserisia committed on Jun 17

1 parent 578167a commit fa222de

Showing 1 changed file with 19 additions and 27 deletions.

Whitespace

Ignore whitespace

Split

Unified

46 src/main/java/br/com/webbudget/domain/entities/registration/MovementClass.java

... 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 34 35 36 37 21 22 23 24 25 26 38 39 40

... @@ -1,19 +1,3 @@
- /*
- * Copyright (C) 2014 Arthur Gregorio, AG.Software
- *
- * This program is free software: you can redistribute it and/or modify
- * it under the terms of the GNU General Public License as published by
- * the Free Software Foundation, either version 3 of the License, or
- * (at your option) any later version.
- *
- * This program is distributed in the hope that it will be useful,
- * but WITHOUT ANY WARRANTY; without even the implied warranty of
- * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
- * GNU General Public License for more details.
- *
- * You should have received a copy of the GNU General Public License
- * along with this program. If not, see <http://www.gnu.org/licenses/>.
- */
1 package br.com.webbudget.domain.entities.registration;
2
3 import br.com.webbudget.domain.entities.PersistentEntity;
18 import static br.com.webbudget.infrastructure.utils.DefaultSchemes.REGISTRATION_AUDIT;
19
20 /**
21 * The representation of a movement class in the application
22 * Represents a movement class in the application.
23 * Stores information about the class name, budget, active status, type, cost center, and total
24 * movements.
25 * Handles operations related to the class.
26 *
27 * This class is used to define various types of movements, such as revenue or expenses.
28 * Each movement class is associated with a cost center.
29 *
30 * @author Arthur Gregorio
31 */

```

54 43 @NotBlank(message = "{movement-class.name}")
55 44 @Column(name = "name", nullable = false, length = 45)
56 45 private String name;
57 46 +
58 47 @Getter
59 48 @Setter
60 49 @Column(name = "budget")
61 50 @NotNull(message = "{movement-class.budget}")
62 51 private BigDecimal budget;
63 52 +
64 53 @Getter
65 54 @Setter
66 55 @Column(name = "active", nullable = false)
67 56 private boolean active;
68 57 +
69 58 @Getter
70 59 @Setter
71 60 @Enumerated(EnumType.STRING)
72 75 private BigDecimal totalMovements;
73 76
74 77 /**
75 78 - * Default constructor
76 79 + * Default constructor.
77 80 + * Sets the initial values for active, budget, and totalMovements.
78 81 */
79 82 public MovementClass() {
80 83     this.active = true;
81 84 }
82 85
83 86 /**
84 87 - * To check if is a revenue class
85 88 + * Checks if this is a revenue class.
86 89 *
87 90 - * @return true for revenue class, false otherwise
88 91 + * @return true if it is a revenue class, false otherwise
89 92 */
90 93 public boolean isRevenue() {
91 94     return this.movementClassType == MovementClassType.REVENUE;
92 95 }
93 96
94 97 /**
95 98 - * To check if is a expense class
96 99 + * Checks if this is an expense class.
97 100 *
98 101 - * @return true for expenses class, false otherwise
99 102 + * @return true if it is an expense class, false otherwise
100 103 */
101 104 public boolean isExpense() {
102 105     return this.movementClassType == MovementClassType.EXPENSE;
103 106 }
104 107
105 108 /**
106 109 - * Method to check if the budget of this class is over the maximum value
107 110 + * Checks if the budget of this class is exceeded.
108 111 *
109 112 - * @return true if the budget is over the maximum, false otherwise
110 113 + * @return true if the budget is exceeded, false otherwise
111 114 */
112 115 public boolean isOverBudget() {

```

```
118 111         return this.totalMovements.compareTo(this.budget) >= 0;
119 112     }
120 113
121 114     /**
122  -     * Method used in the UI to draw the graphic to show the user how many percents of the class
      budget is already used
123 115 +     * Calculates the budget completion percentage for this class.
124 116     *
125 117 +     * @return the budget consume percentage
126 118     *
127 119     */
128 120     public int budgetCompletionPercentage() {
129 121         BigDecimal percentage = BigDecimal.ZERO;
130 122         if (this.isOverBudget()) {
```

0 comments on commit fa222de

Please [sign in](#) to comment.