

Spigot - ItemBuilder

Moin,

als ich den Eintrag "Spigot Inventare erstellen..." gesehen habe, mit dem ItemBuilder fiel mir auf dass dieser relativ klein geschrieben wurde. Deshalb veröffentliche ich mal meinen, der einen relativ großen Funktionsumfang bietet, und welchen ich ggf noch weiter ausbaue.

Java: ItemBuilder

```
1. import org.bukkit.ChatColor;
2. import org.bukkit.Color;
3. import org.bukkit.Material;
4. import org.bukkit.enchantments.Enchantment;
5. import org.bukkit.inventory.ItemFlag;
6. import org.bukkit.inventory.ItemStack;
7. import org.bukkit.inventory.meta.ItemMeta;
8. import org.bukkit.inventory.meta.LeatherArmorMeta;
9. import org.bukkit.inventory.meta.SkullMeta;
10. import java.util.Arrays;
11. import java.util.List;
12. /**
13.  * ItemBuilder for
14.  * @see ItemStack
15.  * by SpigotException
16.  *
17.  * version 1.4
18.  */
19. public class ItemBuilder implements Cloneable{
20. //<editor-fold desc="variables">
21. private ItemStack itemStack;
22. private ItemMeta itemMeta;
23. private SkullMeta skullMeta;
24. private LeatherArmorMeta leatherArmorMeta;
25. //</editor-fold>
26. //<editor-fold desc="constructor with ItemStack">
27. /**
28.  * Constructor with params:
29.  * @param itemStack -> ItemStack for item
30.  */
31. public ItemBuilder(ItemStack itemStack) {
32. this.itemStack = itemStack.clone();
33. this.itemMeta = itemStack.getItemMeta();
34. }
35. //</editor-fold>
36. //<editor-fold desc="constructor with material">
37. /**
38.  * Constructor with params:
```

```

46. * @param material -> Material for item
47. */
48. public ItemBuilder(Material material) {
49. this.itemStack = new ItemStack(material);
50. this.itemMeta = itemStack.getItemMeta();
51. }
52. }
53. //</editor-fold>
54. //<editor-fold desc="setMaterial with Material">
55. /**
56. * set material with params:
57. * @param material -> material
58. * @return this class
59. */
60. public ItemBuilder setMaterial(Material material) {
61. this.itemStack = new ItemStack(material);
62. return this;
63. }
64. //</editor-fold>
65. //<editor-fold desc="setMaterial with ItemStack">
66. /**
67. * set itemstack of item with params:
68. * @param itemStack -> itemstack
69. * @return this class
70. */
71. public ItemBuilder setMaterial(ItemStack itemStack) {
72. this.itemStack = itemStack.clone();
73. return this;
74. }
75. //</editor-fold>
76. //<editor-fold desc="setDisplayname">
77. /**
78. * set the DisplayName for the item with params:
79. * @param name -> name which will be displayed
80. * @return this class
81. */
82. public ItemBuilder setDisplayName(String name) {
83. itemMeta.setDisplayName(ChatColor.translateAlternateColorCodes('&', name));
84. return this;
85. }
86. //</editor-fold>
87. //<editor-fold desc="setMetalID">
88. /**
89. * set the subID/metalID for the item (if needed) with params:
90. * @param metalID -> subID/metalID for item
91. * @return this class
92. */
93. public ItemBuilder setMetalID(byte metalID) {
94. itemStack.getData().setData(metalID);
95. return this;
96. }
97. //</editor-fold>
98. //<editor-fold desc="setFlag">
99. /**

```

```

109. * add itemflags to item meta
110. * @param itemFlags -> itemflag
111. * @return this class
112. */
113. public ItemBuilder setFlag(ItemFlag... itemFlags) {
115. this.itemMeta.addItemFlags(itemFlags);
116. return this;
117. }
118. //</editor-fold>
119. //<editor-fold desc="setAmount">
121. /**
122. * set the amount of items with params:
123. * @param amount -> amount of items
124. * @return this class
125. */
126. public ItemBuilder setAmount(int amount) {
128. this.itemStack.setAmount(amount);
129. return this;
130. }
131. //</editor-fold>
132. //<editor-fold desc="setDurability">
134. /**
135. * set the durability of the item with params:
136. * @param durability -> durability to set
137. * @return this class
138. */
139. public ItemBuilder setDurability(short durability) {
141. this.itemStack.setDurability(durability);
142. return this;
143. }
144. //</editor-fold>
145. //<editor-fold desc="addEnchantment">
147. /**
148. * add enchantment to item with params:
149. * @param enchantment -> enchantment for the item
150. * @param lvl -> level of enchantment
151. * @return this class
152. */
153. public ItemBuilder addEnchantment(Enchantment enchantment, int lvl) {
155. this.itemMeta.addEnchant(enchantment, lvl, false);
156. return this;
157. }
158. //</editor-fold>
159. //<editor-fold desc="addUnsafeEnchantment">
161. /**
162. * add unsafe enchantment to item with params:
163. * @param enchantment -> enchantment for the item
164. * @param lvl -> level of enchantment
165. * @return this class
166. */
167. public ItemBuilder addUnsafeEnchantment(Enchantment enchantment, int lvl) {
169. this.itemStack.addUnsafeEnchantment(enchantment, lvl);
170. return this;

```

```

171. }
172. //</editor-fold>
173. //<editor-fold desc="clearEnchantments">
174. /**
175. * remove all enchantments from the item
176. * @return this class
177. */
178. */
179. public ItemBuilder clearEnchantments() {
180. this.itemMeta.getEnchants().forEach((enchantment, integer) -> this
181. .itemMeta.removeEnchant(enchantment));
182. return this;
183. }
184. //</editor-fold>
185. //<editor-fold desc="removeEnchantment">
186. /**
187. * remove an enchantment from the item with params:
188. * @param enchantment -> enchantment which will be removed
189. * @return this class
190. */
191. */
192. public ItemBuilder removeEnchantment(Enchantment enchantment) {
193. if(this.itemMeta.getEnchants().containsKey(enchantment))
194. this.itemMeta.removeEnchant(enchantment);
195. return this;
196. }
197. }
198. //</editor-fold>
199. //<editor-fold desc="setLore with List">
200. /**
201. * set the lore with an list and params:
202. * @param lines -> lines for lore
203. * @return this class
204. */
205. */
206. public ItemBuilder setLore(List<String> lines) {
207. this.itemMeta.setLore(lines);
208. return this;
209. }
210. }
211. //</editor-fold>
212. //<editor-fold desc="setLore with Array">
213. /**
214. * set the lore with an array and params:
215. * @param lines -> lines for lore
216. * @return this class
217. */
218. */
219. public ItemBuilder setLore(String... lines) {
220. this.itemMeta.setLore(Arrays.asList(lines));
221. return this;
222. }
223. }
224. //</editor-fold>
225. //<editor-fold desc="addLore">
226. /**
227. * add lines to lore
228. * @param lines -> lines to add
229. * @return this class
230. */
231. */

```

```

232. public ItemBuilder addLore(String... lines) {
233.     this.itemMeta.getLore().addAll(Arrays.asList(lines));
234.     return this;
235. }
236. //</editor-fold>
237. //<editor-fold desc="resetLore">
238. /**
239.  * clears the lore of the item
240.  * @return this class
241.  */
242. public ItemBuilder resetLore() {
243.     this.itemMeta.getLore().clear();
244.     return this;
245. }
246. //</editor-fold>
247. //<editor-fold desc="setSkullOwner">
248. /**
249.  * set the skull texture by player name with params:
250.  * @param name -> name of player
251.  * @return this class
252.  */
253. public ItemBuilder setSkullOwner(String name) {
254.     itemStack.setItemMeta(itemMeta);
255.     skullMeta = (SkullMeta) itemStack.getItemMeta();
256.     skullMeta.setOwner(name);
257.     itemStack.setItemMeta(skullMeta);
258.     itemMeta = itemStack.getItemMeta();
259.     return this;
260. }
261. //</editor-fold>
262. //<editor-fold desc="setColor">
263. /**
264.  * set the color of leather armor with params:
265.  * @param color -> color for the armor
266.  * @return this class
267.  */
268. public ItemBuilder setColor(Color color) {
269.     itemStack.setItemMeta(itemMeta);
270.     leatherArmorMeta = (LeatherArmorMeta) itemStack.getItemMeta();
271.     leatherArmorMeta.setColor(color);
272.     itemStack.setItemMeta(leatherArmorMeta);
273.     itemMeta = itemStack.getItemMeta();
274.     return this;
275. }
276. //</editor-fold>
277. //<editor-fold desc="build as normal itemstack">
278. /**
279.  *
280.  * @return the final ItemStack
281.  */
282. public ItemStack build() {
283.     itemStack.setItemMeta(itemMeta);
284.     return itemStack;

```

```
293. }
294. //</editor-fold>
296. //<editor-fold desc="build as skull">
297. /**
298.  * @return the final ItemStack as head
299.  */
300. public ItemStack buildSkull() {
301.     this.setDurability((short)3);
302.     itemStack.setItemMeta(itemMeta);
303.     return itemStack;
304. }
305. //</editor-fold>
306. }
```

Alles anzeigen

Verwendung ist relativ simpel, ein kleines Beispiel welches ohne IDE geschrieben wurde:

```
player.getInventory().setItem(1,
ItemBuilder(Material.CLOCK).setDisplayNames("%5Teleporter in eine neue
Dimension").build())
```

Im Prinzip kann man mal alle Methode anschauen und ausprobieren 😊

MfG