



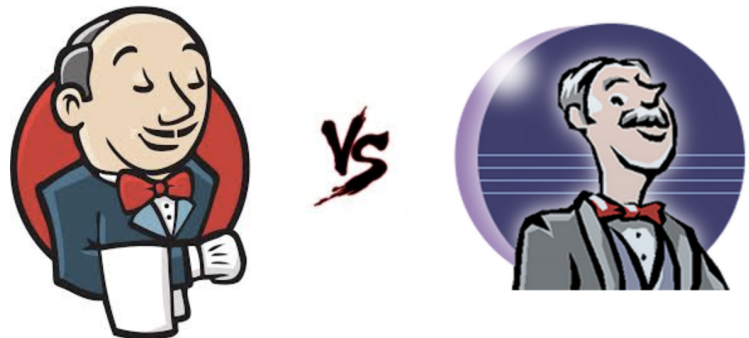
Vea a los mejores desarrolladores hablar sobre todo lo relac...

Regístrese hoy ▶

Jenkins: ¿X o 2.0?

por Oleksii Dzhulai · 13 y 18 de abril · Zona de
nubes · Presentación

Site24x7 - Monitoreo completo Monitoreo de
infraestructura desde la nube. Regístrese para una
prueba gratuita.



Jenkins X se ha convertido en uno de los anuncios
recientes de código abierto más notorios.

En este artículo, analizo las características atractivas
del nuevo producto que no se han mencionado
específicamente en la documentación.

Recomiendo leer sobre la motivación y las
características si necesita una guía de producto o
especificaciones de cualquier tipo.

¿Es solo una nueva herramienta publicitaria? ¿Qué tiene de especial la ejecución de Jenkins en contenedores?

Kubernetes es el estándar de facto para administrar
contenedores, aplicaciones distribuidas e
infraestructura virtual. Se ofrece como un servicio
administrado por los principales proveedores de

nubes públicas y se puede instalar en las instalaciones. Si construyes una aplicación en la nube hoy y quieres que sea portátil en cualquier lugar, ¡Kubernetes es tu elección!

Si bien el ecosistema es enorme, todavía es muy joven y está más centrado en lidiar con los desafíos del "Día 1", como el despliegue de K8. Es natural resolver primero los problemas más urgentes. Se prevé que 2018 traerá más aspectos de estabilización, integración y experiencia del usuario. Hoy, pensamos en cómo operar el Día 2, cómo construir algo en la cima y hacer que los desarrolladores sean más productivos.

Jenkins X está llenando el vacío con la gestión holística de CI / CD aquí.

¿Es solo un cambio de marca de Jenkins 2.0?

Jenkins es una herramienta que se desarrolló mucho antes de que la nube se convirtiera en la norma y, definitivamente, no es una herramienta nativa de la nube, lo que significa que no tiene la capacidad OOTB para sobrevivir interrupciones, escalamiento continuo, etc.

Afortunadamente para nosotros, es extensible. Jenkins X hace su magia con el complemento Kubernetes, lo que le permite olvidarse del aprovisionamiento de máquinas virtuales o servidores físicos para esclavos; cada trabajo utiliza agentes desechables, que se ejecutan en contenedores separados, que pueden escalar con su clúster.

Con Jenkins X, no solo obtiene las canalizaciones de K8S para sus aplicaciones, sino también una solución CI / CD muy escalable.





¿Puedes instalar a Jenkins para nuestro equipo?

Esta es la pregunta más frecuente de los equipos de desarrollo cuando Kubernetes se presenta como una plataforma de proyecto de elección. La migración de herramientas principales significa muchos cambios para cualquier organización; literalmente, afecta a todos los procesos importantes de DevOps y requiere un gran esfuerzo de aprendizaje.

Jenkins X se envía con tuberías, agentes e integraciones de K8, lo que hace que las migraciones a Kubernetes y microservicios sean más simples.

¿Qué pasa con el control de versiones?

Jenkins X está construido con el concepto de GitOps en mente.

Mantener todas las configuraciones en control de versiones significa una mejor seguridad, recuperación ante desastres y la posibilidad de aplicar todas las prácticas de SDLC a las tareas de operaciones.

Al instante, obtienes la configuración de infraestructura de CI / CD guardada en Git y tu juego de herramientas sigue ese principio de manera continua.

¡Nunca tienes la oportunidad de romper esta regla debido a alguna urgencia y esto es por diseño!

Bloques de construcción

Jenkins2.0 - servidor de CI "bueno viejo"

Helm - gestor de paquetes

Draft - herramienta de construcción de entorno dev

Monocular - UI para repositorios de timón

Chartmuseum - repositorio de timón con compatibilidad con back - end de nube

Nexus - repositorio de artefactos

Registro de Docker - registro de imagen de contenedor

Todo lo anterior está bajo el control de una herramienta **jx** y la configuración se envía como una "plataforma", conjunto de gráficos de timón.



Una nueva mirada a la gestión de entornos

¡Uno de los más interesantes para mí!

El entorno como entidad está controlado por la herramienta, puede administrar el ciclo de vida, promover y configurar. Puede personalizar, modelar y reutilizar fácilmente.

Se usa el patrón GitOps y tiene un repositorio para cada entorno. Esta es la mejor práctica y es impulsado por OOTB **JX**.

Helm se utiliza para la gestión de la configuración. El entorno se representa como un paquete de timón y cada aplicación se agrega como una dependencia, mientras se promociona. Este enfoque tiene mucho sentido ya que la configuración del entorno es el área menos estandarizada. Muchos equipos pasaron mucho tiempo hasta que llegué a algo similar. ¡Lo obtendrás gratis!

Swiss Army Knife CLI

Jx CLI es un punto de entrada a todas las funcionalidades del framework.

Lo más importante: es una construcción clave, ampliamente invocada en las tuberías de CD. Pueden reproducirse y depurarse fácilmente directamente, cualquier paso puede reproducirse o repetirse desde CLI.

Site24x7 - Monitoreo completo Monitoreo de infraestructura desde la nube. Regístrese para una prueba gratuita.

Temas: KUBERNETES, JENKINS CI, ARQUITECTURA DE MICROSERVICIO, DOCKER, DEVOPS

Las opiniones expresadas por los contribuidores de DZone son suyas.

Recursos de Cloud Partner

Informe técnico: Site24x7: Gestión de TI potente, ágil y rentable desde la nube

Zoho



Métricas clave de AWS Elastic Load Balancer que necesita para supervisar

Zoho



Tres indicadores clave para monitorear el rendimiento y el uso de AWS SNS

Zoho



Desarrolle, implemente y escale aplicaciones en la nube de manera más rápida y eficiente

Digital Ocean



Spice Up Your Java Using Lombok Annotations!

by Sudip Bhandari · Apr 11, 18 · Java Zone · Tutorial

Get the Edge with a Professional Java IDE. 30-day free trial.

Java is heavily criticized and hated for its verbosity. Even for things such as reading/writing files we have to ceremoniously write a lot of generic stuff. The

increasingly popular Java library Lombok has been

increasingly popular Java library Lombok has been hugely successful in addressing this issue. It's an annotation based Java library that plugs itself into editors, IDE, and build tools at compile time, facilitating the generation of boilerplate code and making the code clean and more readable. There are a lot of features and annotations in this library, but to illustrate how useful I have found it in my day to day work I am going to talk about `@Data` annotation.

Let's say we have a POJO (Plain old Java object) to represent some sort of data in our program. Using vanilla Java this is what we get:

```
1  public class Person {
2      private String id,name,nationality;
3
4      public Person(String id, String name, String nationality) {
5          this.id = id;
6          this.name = name;
7          this.nationality = nationality;
8      }
9
10     public String getId() {
11         return id;
12     }
13
14     public void setId(String id) {
15         this.id = id;
16     }
17
18     public String getName() {
19         return name;
20     }
21
22     public void setName(String name) {
23         this.name = name;
24     }
25
26     public String getNationality() {
27         return nationality;
28     }
29
30     public void setNationality(String nationality) {
31         this.nationality = nationality;
32     }
33 }
```

```

33
34 //override equals, hashCode, toString
35 }

```

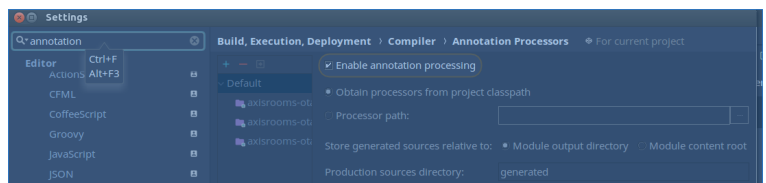
There is nothing wrong with this code. Much of these boilerplate codes can be generated by using internal IDE tools too. But going through this code is not a pleasant experience. Is it just a Data class or are there some special methods embedded too? Now let's write the same using `@Data`.

```

1  import lombok.Data;
2
3  @Data
4  public class Person {
5      private String id,name,nationality;
6  }

```

Yup, that's it. `@Data` annotation creates getters, setters, constructors, overrides equals, hashCode and toString all for us. The code is now very clean and readable, not confusing. For this to work, we need to import Lombok library (jar, Gradle, and Maven) and based on IDE, annotation processing should be turned on. For example, in IntelliJ:



After annotating the Java class, we can check what's been inserted by using the `javap` disassembler or IDE tool called `DeLombok`.

For our above example:

```

//generated using delombok tool of IntelliJ IC
1
2
3  public class Person {
4      private String id,name,nationality;
5
6      public Person() {
7      }
8
9      public String getId() {
10         return this.id;
11     }

```

```

12
13     public String getName() {
14         return this.name;
15     }
16
17     public String getNationality() {
18         return this.nationality;
19     }
20
21     public void setId(String id) {
22         this.id = id;
23     }
24
25     public void setName(String name) {
26         this.name = name;
27     }
28
29     public void setNationality(String nationality) {
30         this.nationality = nationality;
31     }
32
33     public boolean equals(Object o) {
34         if (o == this)
35             return true;
36         if (!(o instanceof Person))
37             return false;
38         final Person other = (Person) o;
39         if (!other.canEqual((Object) this))
40             return false;
41         final Object this$id = this.getId();
42         final Object other$id = other.getId();
43         if (this$id == null ? other$id != null
44             : !this$id.equals(other$id))
45             return false;
46         final Object this$name = this.getName();
47         final Object other$name = other.getName();
48         if (this$name == null ? other$name != null
49             : !this$name.equals(other$name))
50             return false;
51         final Object this$nationality = this.getNationality();
52         final Object other$nationality = other.getNationality();
53         if (this$nationality == null ? other$nationality != null
54             : !this$nationality.equals(other$nationality))
55             return false;
56         return true;
57     }
58
59     public int hashCode() {
60         int result = 1;
61         result = result * 31 + getId().hashCode();
62         result = result * 31 + getName().hashCode();
63         result = result * 31 + getNationality().hashCode();
64         return result;
65     }
66
67     public String toString() {
68         return "Person{id='" + getId() + "', name='" + getName() + "', nationality='" + getNationality() + "'}";
69     }
70
71     public static Person create(String id, String name, String nationality) {
72         return new Person(id, name, nationality);
73     }
74
75     public static Person[] createArray(int size) {
76         Person[] persons = new Person[size];
77         for (int i = 0; i < persons.length; i++) {
78             persons[i] = create(i + 1, "Person " + i, "Nationality " + i);
79         }
80         return persons;
81     }
82
83     public static void main(String[] args) {
84         Person p1 = create(1, "John", "American");
85         Person p2 = create(2, "Jane", "British");
86         Person p3 = create(3, "Bob", "Canadian");
87         Person p4 = create(4, "Alice", "Australian");
88         Person p5 = create(5, "Charlie", "New Zealander");
89         Person p6 = create(6, "Diana", "South African");
90         Person p7 = create(7, "Eve", "Indian");
91         Person p8 = create(8, "Frank", "Chinese");
92         Person p9 = create(9, "Grace", "Japanese");
93         Person p10 = create(10, "Henry", "Korean");
94         Person p11 = create(11, "Ivy", "Vietnamese");
95         Person p12 = create(12, "Jack", "Thai");
96         Person p13 = create(13, "Karen", "Filipino");
97         Person p14 = create(14, "Leo", "Indonesian");
98         Person p15 = create(15, "Mia", "Malay");
99         Person p16 = create(16, "Noah", "Singaporean");
100        Person p17 = create(17, "Olivia", "Bruneian");
101        Person p18 = create(18, "Peter", "Mongolian");
102        Person p19 = create(19, "Quinn", "Tibetan");
103        Person p20 = create(20, "Sam", "Hmong");
104        Person p21 = create(21, "Tina", "Khmer");
105        Person p22 = create(22, "Uma", "Lao");
106        Person p23 = create(23, "Victor", "Myanmar");
107        Person p24 = create(24, "Wendy", "Cambodian");
108        Person p25 = create(25, "Xavier", "Laotian");
109        Person p26 = create(26, "Yara", "Vietnamese");
110        Person p27 = create(27, "Zoe", "Thai");
111        Person p28 = create(28, "Adam", "Filipino");
112        Person p29 = create(29, "Bella", "Indonesian");
113        Person p30 = create(30, "Charlie", "Malay");
114        Person p31 = create(31, "Diana", "Singaporean");
115        Person p32 = create(32, "Eve", "Bruneian");
116        Person p33 = create(33, "Frank", "Mongolian");
117        Person p34 = create(34, "Grace", "Tibetan");
118        Person p35 = create(35, "Henry", "Hmong");
119        Person p36 = create(36, "Ivy", "Khmer");
120        Person p37 = create(37, "Jack", "Lao");
121        Person p38 = create(38, "Karen", "Myanmar");
122        Person p39 = create(39, "Leo", "Cambodian");
123        Person p40 = create(40, "Mia", "Laotian");
124        Person p41 = create(41, "Noah", "Vietnamese");
125        Person p42 = create(42, "Olivia", "Thai");
126        Person p43 = create(43, "Peter", "Filipino");
127        Person p44 = create(44, "Quinn", "Indonesian");
128        Person p45 = create(45, "Sam", "Malay");
129        Person p46 = create(46, "Tina", "Singaporean");
130        Person p47 = create(47, "Uma", "Bruneian");
131        Person p48 = create(48, "Victor", "Mongolian");
132        Person p49 = create(49, "Wendy", "Tibetan");
133        Person p50 = create(50, "Xavier", "Hmong");
134        Person p51 = create(51, "Yara", "Khmer");
135        Person p52 = create(52, "Zoe", "Lao");
136        Person p53 = create(53, "Adam", "Myanmar");
137        Person p54 = create(54, "Bella", "Cambodian");
138        Person p55 = create(55, "Charlie", "Laotian");
139        Person p56 = create(56, "Diana", "Vietnamese");
140        Person p57 = create(57, "Eve", "Thai");
141        Person p58 = create(58, "Frank", "Filipino");
142        Person p59 = create(59, "Grace", "Indonesian");
143        Person p60 = create(60, "Henry", "Malay");
144        Person p61 = create(61, "Ivy", "Singaporean");
145        Person p62 = create(62, "Jack", "Bruneian");
146        Person p63 = create(63, "Karen", "Mongolian");
147        Person p64 = create(64, "Leo", "Tibetan");
148        Person p65 = create(65, "Mia", "Hmong");
149        Person p66 = create(66, "Noah", "Khmer");
150        Person p67 = create(67, "Olivia", "Lao");
151        Person p68 = create(68, "Peter", "Myanmar");
152        Person p69 = create(69, "Quinn", "Cambodian");
153        Person p70 = create(70, "Sam", "Laotian");
154        Person p71 = create(71, "Tina", "Vietnamese");
155        Person p72 = create(72, "Uma", "Thai");
156        Person p73 = create(73, "Victor", "Filipino");
157        Person p74 = create(74, "Wendy", "Indonesian");
158        Person p75 = create(75, "Xavier", "Malay");
159        Person p76 = create(76, "Yara", "Singaporean");
160        Person p77 = create(77, "Zoe", "Bruneian");
161        Person p78 = create(78, "Adam", "Mongolian");
162        Person p79 = create(79, "Bella", "Tibetan");
163        Person p80 = create(80, "Charlie", "Hmong");
164        Person p81 = create(81, "Diana", "Khmer");
165        Person p82 = create(82, "Eve", "Lao");
166        Person p83 = create(83, "Frank", "Myanmar");
167        Person p84 = create(84, "Grace", "Cambodian");
168        Person p85 = create(85, "Henry", "Laotian");
169        Person p86 = create(86, "Ivy", "Vietnamese");
170        Person p87 = create(87, "Jack", "Thai");
171        Person p88 = create(88, "Karen", "Filipino");
172        Person p89 = create(89, "Leo", "Indonesian");
173        Person p90 = create(90, "Mia", "Malay");
174        Person p91 = create(91, "Noah", "Singaporean");
175        Person p92 = create(92, "Olivia", "Bruneian");
176        Person p93 = create(93, "Peter", "Mongolian");
177        Person p94 = create(94, "Quinn", "Tibetan");
178        Person p95 = create(95, "Sam", "Hmong");
179        Person p96 = create(96, "Tina", "Khmer");
180        Person p97 = create(97, "Uma", "Lao");
181        Person p98 = create(98, "Victor", "Myanmar");
182        Person p99 = create(99, "Wendy", "Cambodian");
183        Person p100 = create(100, "Xavier", "Laotian");
184        Person p101 = create(101, "Yara", "Vietnamese");
185        Person p102 = create(102, "Zoe", "Thai");
186        Person p103 = create(103, "Adam", "Filipino");
187        Person p104 = create(104, "Bella", "Indonesian");
188        Person p105 = create(105, "Charlie", "Malay");
189        Person p106 = create(106, "Diana", "Singaporean");
190        Person p107 = create(107, "Eve", "Bruneian");
191        Person p108 = create(108, "Frank", "Mongolian");
192        Person p109 = create(109, "Grace", "Tibetan");
193        Person p110 = create(110, "Henry", "Hmong");
194        Person p111 = create(111, "Ivy", "Khmer");
195        Person p112 = create(112, "Jack", "Lao");
196        Person p113 = create(113, "Karen", "Myanmar");
197        Person p114 = create(114, "Leo", "Cambodian");
198        Person p115 = create(115, "Mia", "Laotian");
199        Person p116 = create(116, "Noah", "Vietnamese");
200        Person p117 = create(117, "Olivia", "Thai");
201        Person p118 = create(118, "Peter", "Filipino");
202        Person p119 = create(119, "Quinn", "Indonesian");
203        Person p120 = create(120, "Sam", "Malay");
204        Person p121 = create(121, "Tina", "Singaporean");
205        Person p122 = create(122, "Uma", "Bruneian");
206        Person p123 = create(123, "Victor", "Mongolian");
207        Person p124 = create(124, "Wendy", "Tibetan");
208        Person p125 = create(125, "Xavier", "Hmong");
209        Person p126 = create(126, "Yara", "Khmer");
210        Person p127 = create(127, "Zoe", "Lao");
211        Person p128 = create(128, "Adam", "Myanmar");
212        Person p129 = create(129, "Bella", "Cambodian");
213        Person p130 = create(130, "Charlie", "Laotian");
214        Person p131 = create(131, "Diana", "Vietnamese");
215        Person p132 = create(132, "Eve", "Thai");
216        Person p133 = create(133, "Frank", "Filipino");
217        Person p134 = create(134, "Grace", "Indonesian");
218        Person p135 = create(135, "Henry", "Malay");
219        Person p136 = create(136, "Ivy", "Singaporean");
220        Person p137 = create(137, "Jack", "Bruneian");
221        Person p138 = create(138, "Karen", "Mongolian");
222        Person p139 = create(139, "Leo", "Tibetan");
223        Person p140 = create(140, "Mia", "Hmong");
224        Person p141 = create(141, "Noah", "Khmer");
225        Person p142 = create(142, "Olivia", "Lao");
226        Person p143 = create(143, "Peter", "Myanmar");
227        Person p144 = create(144, "Quinn", "Cambodian");
228        Person p145 = create(145, "Sam", "Laotian");
229        Person p146 = create(146, "Tina", "Vietnamese");
230        Person p147 = create(147, "Uma", "Thai");
231        Person p148 = create(148, "Victor", "Filipino");
232        Person p149 = create(149, "Wendy", "Indonesian");
233        Person p150 = create(150, "Xavier", "Malay");
234        Person p151 = create(151, "Yara", "Singaporean");
235        Person p152 = create(152, "Zoe", "Bruneian");
236        Person p153 = create(153, "Adam", "Mongolian");
237        Person p154 = create(154, "Bella", "Tibetan");
238        Person p155 = create(155, "Charlie", "Hmong");
239        Person p156 = create(156, "Diana", "Khmer");
240        Person p157 = create(157, "Eve", "Lao");
241        Person p158 = create(158, "Frank", "Myanmar");
242        Person p159 = create(159, "Grace", "Cambodian");
243        Person p160 = create(160, "Henry", "Laotian");
244        Person p161 = create(161, "Ivy", "Vietnamese");
245        Person p162 = create(162, "Jack", "Thai");
246        Person p163 = create(163, "Karen", "Filipino");
247        Person p164 = create(164, "Leo", "Indonesian");
248        Person p165 = create(165, "Mia", "Malay");
249        Person p166 = create(166, "Noah", "Singaporean");
250        Person p167 = create(167, "Olivia", "Bruneian");
251        Person p168 = create(168, "Peter", "Mongolian");
252        Person p169 = create(169, "Quinn", "Tibetan");
253        Person p170 = create(170, "Sam", "Hmong");
254        Person p171 = create(171, "Tina", "Khmer");
255        Person p172 = create(172, "Uma", "Lao");
256        Person p173 = create(173, "Victor", "Myanmar");
257        Person p174 = create(174, "Wendy", "Cambodian");
258        Person p175 = create(175, "Xavier", "Laotian");
259        Person p176 = create(176, "Yara", "Vietnamese");
260        Person p177 = create(177, "Zoe", "Thai");
261        Person p178 = create(178, "Adam", "Filipino");
262        Person p179 = create(179, "Bella", "Indonesian");
263        Person p180 = create(180, "Charlie", "Malay");
264        Person p181 = create(181, "Diana", "Singaporean");
265        Person p182 = create(182, "Eve", "Bruneian");
266        Person p183 = create(183, "Frank", "Mongolian");
267        Person p184 = create(184, "Grace", "Tibetan");
268        Person p185 = create(185, "Henry", "Hmong");
269        Person p186 = create(186, "Ivy", "Khmer");
270        Person p187 = create(187, "Jack", "Lao");
271        Person p188 = create(188, "Karen", "Myanmar");
272        Person p189 = create(189, "Leo", "Cambodian");
273        Person p190 = create(190, "Mia", "Laotian");
274        Person p191 = create(191, "Noah", "Vietnamese");
275        Person p192 = create(192, "Olivia", "Thai");
276        Person p193 = create(193, "Peter", "Filipino");
277        Person p194 = create(194, "Quinn", "Indonesian");
278        Person p195 = create(195, "Sam", "Malay");
279        Person p196 = create(196, "Tina", "Singaporean");
280        Person p197 = create(197, "Uma", "Bruneian");
281        Person p198 = create(198, "Victor", "Mongolian");
282        Person p199 = create(199, "Wendy", "Tibetan");
283        Person p200 = create(200, "Xavier", "Hmong");
284        Person p201 = create(201, "Yara", "Khmer");
285        Person p202 = create(202, "Zoe", "Lao");
286        Person p203 = create(203, "Adam", "Myanmar");
287        Person p204 = create(204, "Bella", "Cambodian");
288        Person p205 = create(205, "Charlie", "Laotian");
289        Person p206 = create(206, "Diana", "Vietnamese");
290        Person p207 = create(207, "Eve", "Thai");
291        Person p208 = create(208, "Frank", "Filipino");
292        Person p209 = create(209, "Grace", "Indonesian");
293        Person p210 = create(210, "Henry", "Malay");
294        Person p211 = create(211, "Ivy", "Singaporean");
295        Person p212 = create(212, "Jack", "Bruneian");
296        Person p213 = create(213, "Karen", "Mongolian");
297        Person p214 = create(214, "Leo", "Tibetan");
298        Person p215 = create(215, "Mia", "Hmong");
299        Person p216 = create(216, "Noah", "Khmer");
300        Person p217 = create(217, "Olivia", "Lao");
301        Person p218 = create(218, "Peter", "Myanmar");
302        Person p219 = create(219, "Quinn", "Cambodian");
303        Person p220 = create(220, "Sam", "Laotian");
304        Person p221 = create(221, "Tina", "Vietnamese");
305        Person p222 = create(222, "Uma", "Thai");
306        Person p223 = create(223, "Victor", "Filipino");
307        Person p224 = create(224, "Wendy", "Indonesian");
308        Person p225 = create(225, "Xavier", "Malay");
309        Person p226 = create(226, "Yara", "Singaporean");
310        Person p227 = create(227, "Zoe", "Bruneian");
311        Person p228 = create(228, "Adam", "Mongolian");
312        Person p229 = create(229, "Bella", "Tibetan");
313        Person p230 = create(230, "Charlie", "Hmong");
314        Person p231 = create(231, "Diana", "Khmer");
315        Person p232 = create(232, "Eve", "Lao");
316        Person p233 = create(233, "Frank", "Myanmar");
317        Person p234 = create(234, "Grace", "Cambodian");
318        Person p235 = create(235, "Henry", "Laotian");
319        Person p236 = create(236, "Ivy", "Vietnamese");
320        Person p237 = create(237, "Jack", "Thai");
321        Person p238 = create(238, "Karen", "Filipino");
322        Person p239 = create(239, "Leo", "Indonesian");
323        Person p240 = create(240, "Mia", "Malay");
324        Person p241 = create(241, "Noah", "Singaporean");
325        Person p242 = create(242, "Olivia", "Bruneian");
326        Person p243 = create(243, "Peter", "Mongolian");
327        Person p244 = create(244, "Quinn", "Tibetan");
328        Person p245 = create(245, "Sam", "Hmong");
329        Person p246 = create(246, "Tina", "Khmer");
330        Person p247 = create(247, "Uma", "Lao");
331        Person p248 = create(248, "Victor", "Myanmar");
332        Person p249 = create(249, "Wendy", "Cambodian");
333        Person p250 = create(250, "Xavier", "Laotian");
334        Person p251 = create(251, "Yara", "Vietnamese");
335        Person p252 = create(252, "Zoe", "Thai");
336        Person p253 = create(253, "Adam", "Filipino");
337        Person p254 = create(254, "Bella", "Indonesian");
338        Person p255 = create(255, "Charlie", "Malay");
339        Person p256 = create(256, "Diana", "Singaporean");
340        Person p257 = create(257, "Eve", "Bruneian");
341        Person p258 = create(258, "Frank", "Mongolian");
342        Person p259 = create(259, "Grace", "Tibetan");
343        Person p260 = create(260, "Henry", "Hmong");
344        Person p261 = create(261, "Ivy", "Khmer");
345        Person p262 = create(262, "Jack", "Lao");
346        Person p263 = create(263, "Karen", "Myanmar");
347        Person p264 = create(264, "Leo", "Cambodian");
348        Person p265 = create(265, "Mia", "Laotian");
349        Person p266 = create(266, "Noah", "Vietnamese");
350        Person p267 = create(267, "Olivia", "Thai");
351        Person p268 = create(268, "Peter", "Filipino");
352        Person p269 = create(269, "Quinn", "Indonesian");
353        Person p270 = create(270, "Sam", "Malay");
354        Person p271 = create(271, "Tina", "Singaporean");
355        Person p272 = create(272, "Uma", "Bruneian");
356        Person p273 = create(273, "Victor", "Mongolian");
357        Person p274 = create(274, "Wendy", "Tibetan");
358        Person p275 = create(275, "Xavier", "Hmong");
359        Person p276 = create(276, "Yara", "Khmer");
360        Person p277 = create(277, "Zoe", "Lao");
361        Person p278 = create(278, "Adam", "Myanmar");
362        Person p279 = create(279, "Bella", "Cambodian");
363        Person p280 = create(280, "Charlie", "Laotian");
364        Person p281 = create(281, "Diana", "Vietnamese");
365        Person p282 = create(282, "Eve", "Thai");
366        Person p283 = create(283, "Frank", "Filipino");
367        Person p284 = create(284, "Grace", "Indonesian");
368        Person p285 = create(285, "Henry", "Malay");
369        Person p286 = create(286, "Ivy", "Singaporean");
370        Person p287 = create(287, "Jack", "Bruneian");
371        Person p288 = create(288, "Karen", "Mongolian");
372        Person p289 = create(289, "Leo", "Tibetan");
373        Person p290 = create(290, "Mia", "Hmong");
374        Person p291 = create(291, "Noah", "Khmer");
375        Person p292 = create(292, "Olivia", "Lao");
376        Person p293 = create(293, "Peter", "Myanmar");
377        Person p294 = create(294, "Quinn", "Cambodian");
378        Person p295 = create(295, "Sam", "Laotian");
379        Person p296 = create(296, "Tina", "Vietnamese");
380        Person p297 = create(297, "Uma", "Thai");
381        Person p298 = create(298, "Victor", "Filipino");
382        Person p299 = create(299, "Wendy", "Indonesian");
383        Person p300 = create(300, "Xavier", "Malay");
384        Person p301 = create(301, "Yara", "Singaporean");
385        Person p302 = create(302, "Zoe", "Bruneian");
386        Person p303 = create(303, "Adam", "Mongolian");
387        Person p304 = create(304, "Bella", "Tibetan");
388        Person p305 = create(305, "Charlie", "Hmong");
389        Person p306 = create(306, "Diana", "Khmer");
390        Person p307 = create(307, "Eve", "Lao");
391        Person p308 = create(308, "Frank", "Myanmar");
392        Person p309 = create(309, "Grace", "Cambodian");
393        Person p310 = create(310, "Henry", "Laotian");
394        Person p311 = create(311, "Ivy", "Vietnamese");
395        Person p312 = create(312, "Jack", "Thai");
396        Person p313 = create(313, "Karen", "Filipino");
397        Person p314 = create(314, "Leo", "Indonesian");
398        Person p315 = create(315, "Mia", "Malay");
399        Person p316 = create(316, "Noah", "Singaporean");
400        Person p317 = create(317, "Olivia", "Bruneian");
401        Person p318 = create(318, "Peter", "Mongolian");
402        Person p319 = create(319, "Quinn", "Tibetan");
403        Person p320 = create(320, "Sam", "Hmong");
404        Person p321 = create(321, "Tina", "Khmer");
405        Person p322 = create(322, "Uma", "Lao");
406        Person p323 = create(323, "Victor", "Myanmar");
407        Person p324 = create(324, "Wendy", "Cambodian");
408        Person p325 = create(325, "Xavier", "Laotian");
409        Person p326 = create(326, "Yara", "Vietnamese");
410        Person p327 = create(327, "Zoe", "Thai");
411        Person p328 = create(328, "Adam", "Filipino");
412        Person p329 = create(329, "Bella", "Indonesian");
413        Person p330 = create(330, "Charlie", "Malay");
414        Person p331 = create(331, "Diana", "Singaporean");
415        Person p332 = create(332, "Eve", "Bruneian");
416        Person p333 = create(333, "Frank", "Mongolian");
417        Person p334 = create(334, "Grace", "Tibetan");
418        Person p335 = create(335, "Henry", "Hmong");
419        Person p336 = create(336, "Ivy", "Khmer");
420        Person p337 = create(337, "Jack", "Lao");
421        Person p338 = create(338, "Karen", "Myanmar");
422        Person p339 = create(339, "Leo", "Cambodian");
423        Person p340 = create(340, "Mia", "Laotian");
424        Person p341 = create(341, "Noah", "Vietnamese");
425        Person p342 = create(342, "Olivia", "Thai");
426        Person p343 = create(343, "Peter", "Filipino");
427        Person p344 = create(344, "Quinn", "Indonesian");
428        Person p345 = create(345, "Sam", "Malay");
429        Person p346 = create(346, "Tina", "Singaporean");
430        Person p347 = create(347, "Uma", "Bruneian");
431        Person p348 = create(348, "Victor", "Mongolian");
432        Person p349 = create(349, "Wendy", "Tibetan");
433        Person p350 = create(350, "Xavier", "Hmong");
434        Person p351 = create(351, "Yara", "Khmer");
435        Person p352 = create(352, "Zoe", "Lao");
436        Person p353 = create(353, "Adam", "Myanmar");
437        Person p354 = create(354, "Bella", "Cambodian");
438        Person p355 = create(355, "Charlie", "Laotian");
439        Person p356 = create(356, "Diana", "Vietnamese");
440        Person p357 = create(357, "Eve", "Thai");
441        Person p358 = create(358, "Frank", "Filipino");
442        Person p359 = create(359, "Grace", "Indonesian");
443        Person p360 = create(360, "Henry", "Malay");
444        Person p361 = create(361, "Ivy", "Singaporean");
445        Person p362 = create(362, "Jack", "Bruneian");
446        Person p363 = create(363, "Karen", "Mongolian");
447        Person p364 = create(364, "Leo", "Tibetan");
448        Person p365 = create(365, "Mia", "Hmong");
449        Person p366 = create(366, "Noah", "Khmer");
450        Person p367 = create(367, "Olivia", "Lao");
451        Person p368 = create(368, "Peter", "Myanmar");
452        Person p369 = create(369, "Quinn", "Cambodian");
453        Person p370 = create(370, "Sam", "Laotian");
454        Person p371 = create(371, "Tina", "Vietnamese");
455        Person p372 = create(372, "Uma", "Thai");
456        Person p373 = create(373, "Victor", "Filipino");
457        Person p374 = create(374, "Wendy", "Indonesian");
458        Person p375 = create(375, "Xavier", "Malay");
459        Person p376 = create(376, "Yara", "Singaporean");
460        Person p377 = create(377, "Zoe", "Bruneian");
461        Person p378 = create(378, "Adam", "Mongolian");
462        Person p379 = create(379, "Bella", "Tibetan");
463        Person p380 = create(380, "Charlie", "Hmong");
464        Person p381 = create(381, "Diana", "Khmer");
465        Person p382 = create(382, "Eve", "Lao");
466        Person p383 = create(383, "Frank", "Myanmar");
467        Person p384 = create(384, "Grace", "Cambodian");
468        Person p385 = create(385, "Henry", "Laotian");
469        Person p386 = create(386, "Ivy", "Vietnamese");
470        Person p387 = create(387, "Jack", "Thai");
471        Person p388 = create(388, "Karen", "Filipino");
472        Person p389 = create(389, "Leo", "Indonesian");
473        Person p390 = create(390, "Mia", "Malay");
474        Person p391 = create(391, "Noah", "Singaporean");
475        Person p392 = create(392, "Olivia", "Bruneian");
476        Person p393 = create(393, "Peter", "Mongolian");
477        Person p394 = create(394, "Quinn", "Tibetan");
478        Person p395 = create(395, "Sam", "Hmong");
479        Person p396 = create(396, "Tina", "Khmer");
480        Person p397 = create(397, "Uma", "Lao");
481        Person p398 = create(398, "Victor", "Myanmar");
482        Person p399 = create(399, "Wendy", "Cambodian");
483        Person p400 = create(400, "Xavier", "Laotian");
484        Person p401 = create(401, "Yara", "Vietnamese");
485        Person p402 = create(402, "Zoe", "Thai");
486        Person p403 = create(403, "Adam", "Filipino");
487        Person p404 = create(404, "Bella", "Indonesian");
488        Person p405 = create(405, "Charlie", "Malay");
489        Person p406 = create(406, "Diana", "Singaporean");
490        Person p407 = create(407, "Eve", "Bruneian");
491        Person p408 = create(408, "Frank", "Mongolian");
492        Person p409 = create(409, "Grace", "Tibetan");
493        Person p410 = create(410, "Henry", "Hmong");
494        Person p411 = create(411, "Ivy", "Khmer");
495        Person p412 = create(412, "Jack", "Lao");
496        Person p413 = create(413, "Karen", "Myanmar");
497        Person p414 = create(414, "Leo", "Cambodian");
498        Person p415 = create(415, "Mia", "Laotian");
499        Person p416 = create(416, "Noah", "Vietnamese");
500        Person p417 = create(417, "Olivia", "Thai");
501        Person p418 = create(418, "Peter", "Filipino");
502        Person p419 = create(419, "Quinn", "Indonesian");
503        Person p420 = create(420, "Sam", "Malay");
504        Person p421 = create(421, "Tina", "Singaporean");
505        Person p422 = create(422, "Uma", "Bruneian");
506        Person p423 = create(423, "Victor", "Mongolian");
507        Person p424 = create(424, "Wendy", "Tibetan");
508        Person p425 = create(425, "Xavier", "Hmong");
509        Person p426 = create(426, "Yara", "Khmer");
510        Person p427 = create(427, "Zoe", "Lao");
511        Person p428 = create(428, "Adam", "Myanmar");
512        Person p429 = create(429, "Bella", "Cambodian");
513        Person p430 = create(430, "Charlie", "Laotian");
514        Person p431 = create(431, "Diana", "Vietnamese");
515        Person p432 = create(432, "Eve", "Thai");
516        Person p433 = create(433, "Frank", "Filipino");
517        Person p434 = create(434, "Grace", "Indonesian");
518        Person p435 = create(435, "Henry", "Malay");
519        Person p436 = create(436, "Ivy", "Singaporean");
520        Person p437 = create(437, "Jack", "Bruneian");
521        Person p438 = create(438, "Karen", "Mongolian");
522        Person p439 = create(439, "Leo", "Tibetan");
523        Person p440 = create(440, "Mia", "Hmong");
524        Person p441 = create(441, "Noah", "Khmer");
525        Person p442 = create(442, "Olivia", "Lao");
526        Person p443 = create(443, "Peter", "Myanmar");
527        Person p444 = create(444, "Quinn", "Cambodian");
528        Person p445 = create(445, "Sam", "Laotian");
529        Person p446 = create(446, "Tina", "Vietnamese");
530        Person p447 = create(447, "Uma", "Thai");
531        Person p448 = create(448, "Victor", "Filipino");
532        Person p449 = create(449, "Wendy", "Indonesian");
533        Person p450 = create(450, "Xavier", "Malay");
534        Person p451 = create(451, "Yara", "Singaporean");
535        Person p452 = create(452, "Zoe", "Bruneian");
536        Person p453 = create(453, "Adam", "Mongolian");
537        Person p454 = create(454, "Bella", "Tibetan");
538        Person p455 = create(455, "Charlie", "Hmong");
539        Person p456 = create(456, "Diana", "Khmer");
540        Person p457 = create(457, "Eve", "Lao");
541        Person p458 = create(458, "Frank", "Myanmar");
542        Person p459 = create(459, "Grace", "Cambodian");
543        Person p460 = create(460, "Henry", "Laotian");
544        Person p461 = create(461, "Ivy", "Vietnamese");
545        Person p462 = create(462, "Jack", "Thai");
546        Person p463 = create(463, "Karen", "Filipino");
547        Person p464 = create(464, "Leo", "Indonesian");
548        Person p465 = create(465, "Mia", "Malay");
549        Person p466 = create(466, "Noah", "Singaporean");
550        Person p467 = create(467, "Olivia", "Bruneian");
551        Person p468 = create(468, "Peter", "Mongolian");
552        Person p469 = create(469, "Quinn", "Tibetan");
553        Person p470 = create(470, "Sam", "Hmong");
554        Person p471 = create(471, "Tina", "Khmer");
555        Person p472 = create(472, "Uma", "Lao");
556        Person p473 = create(473, "Victor", "Myanmar");
557        Person p474 = create(474
```



```

        if (this$nationality == null ? other$na
51  ◀────────────────────────────────────────▶
52      return false;
53      return true;
54  }
55
56  public int hashCode() {
57      final int PRIME = 59;
58      int result = 1;
59      final Object $id = this.getId();
60      result = result * PRIME + ($id == null
61  ◀────────────────────────────────────────▶
62      final Object $name = this.getName();
63  ◀────────────────────────────────────────▶
64      result = result * PRIME + ($name == nul
65  ◀────────────────────────────────────────▶
66      final Object $nationality = this.getNat
67  ◀────────────────────────────────────────▶
68      result = result * PRIME + ($nationality
69  ◀────────────────────────────────────────▶
70      return result;
71  }
72
73  protected boolean canEqual(Object other) {
74  ◀────────────────────────────────────────▶
75      return other instanceof Person;
76  }
77
78  public String toString() {
79      return "Person(id=" + this.getId() + ",
80  ◀────────────────────────────────────────▶
81      ")";
82  }
83  }

```

This is just one example of how we can use Lombok to spice up our java coding. There are numerous other annotations for purposes like exception handling, resource cleaning and so on.

Get the Java IDE that understands code & makes developing enjoyable. Level up your code with IntelliJ IDEA. Download the free trial.

Topics: JAVA, LOMBOK, SOFTWARE ENGINEERING, PROGRAMMING, JVM, COMPILATION, SPRINGBOOT, DATA ANNOTATION

Opinions expressed by DZone contributors are their own.