

Commit 67039cfc authored Sep 21, 2023 by  **Mayra Cabrera** 0

parents [eea873dc](#) [ac0f8d8e](#)



No related merge requests found

Files 7

Search (e.g. *.vue) (F)

↑		@@ -169,7 +169,6 @@ included_attributes:
169	169	- :cron_timezone
170	170	- :description
171	171	- :next_run_at
172	-	- :owner_id
173	172	- :project_id
174	173	- :ref
175	174	- :updated_at
↑	↓	@@ -373,7 +372,6 @@ included_attributes:
373	372	- :updated_by_id
374	373	- :merge_error
375	374	- :merge_params
376	-	- :merge_when_pipeline_succeeds
377	375	- :merge_user_id
378	376	- :merge_commit_sha
379	377	- :squash_commit_sha

↑		@@ -98,6 +98,7 @@ def setup_models
98	98	when :ProtectedBranch::PushAccessLevel then
99	99	when :ApprovalProjectRulesProtectedBranch then
100	100	when :releases then setup_release
101	+	when :merge_requests, :MergeRequest, :merge_request then
102	102	end
103	103	update_project_references
↑	↓	@@ -148,22 +149,30 @@ def setup_diff
148	149	end
149	150	
150	151	def setup_pipeline
152	+	@relation_hash['status'] =
153	+	transform_status(@relation_hash['status'])
154	154	@relation_hash.fetch('stages', []).each do stage
155	+	stage.status = transform_status(stage.status)
156	+	
157	157	# old export files have statuses
158	-	stage.statuses.each do status
159	+	stage.statuses.each do job
160	+	job.status = transform_status(job.status)
161	+	job.pipeline = imported_object
162	-	end
163	+	stage.builds.each do status
164	+	status.pipeline = imported_object
165	+	stage.builds.each do job
166	+	job.status = transform_status(job.status)
167	+	job.pipeline = imported_object
168	-	end
169	+	stage.bridges.each do status
170	+	status.pipeline = imported_object
171	+	stage.bridges.each do job
172	+	job.status = transform_status(job.status)
173	+	job.pipeline = imported_object

<u>163</u>	<u>171</u>	end
<u>164</u>	<u>172</u>	
<u>165</u>	-	stage. generic_commit_statuses .each do status
<u>166</u>	-	status.pipeline = imported_object
	<u>173</u>	+ stage. generic_commit_statuses .each do job
	<u>174</u>	+ job.status = transform_status(job.status)
	<u>175</u>	+ job.pipeline = imported_object
<u>167</u>	<u>176</u>	end
<u>168</u>	<u>177</u>	end
<u>169</u>	<u>178</u>	end
	↕	@@ -187,6 +196,10 @@ def setup_pipeline_schedule
<u>187</u>	<u>196</u>	@relation_hash['active'] = false
<u>188</u>	<u>197</u>	end
<u>189</u>	<u>198</u>	
	<u>199</u>	+ def setup_merge_request
	<u>200</u>	+ @relation_hash['merge_when_pipeline_succeeds'] = false
	<u>201</u>	+ end
	<u>202</u>	+
<u>190</u>	<u>203</u>	def setup_protected_branch_access_level
<u>191</u>	<u>204</u>	return if root_group_owner?
<u>192</u>	<u>205</u>	return if @relation_hash['access_level'] ==
		Gitlab::Access::NO_ACCESS
	↑ ↓	@@ -226,6 +239,12 @@ def legacy_trigger?
<u>226</u>	<u>239</u>	@relation_name == ':Ci::Trigger' &&
		@relation_hash['owner_id'].nil?
<u>227</u>	<u>240</u>	end
<u>228</u>	<u>241</u>	
	<u>242</u>	+ def transform_status (status)
	<u>243</u>	+ return 'canceled' if
		::Ci::HasStatus::COMPLETED_STATUSES.exclude?(status)
	<u>244</u>	+
	<u>245</u>	+ status
	<u>246</u>	+ end
	<u>247</u>	+
<u>229</u>	<u>248</u>	def preload_keys (object, references, value)
<u>230</u>	<u>249</u>	return object unless value
<u>231</u>	<u>250</u>	
	↓	

File changed.

Preview size limit exceeded, changes collapsed.

File changed.

Preview size limit exceeded, changes collapsed.

	↑	@@ -172,6 +172,10 @@ def values
<u>172</u>	<u>172</u>	it 'has preloaded target project' do
<u>173</u>	<u>173</u>	expect(created_object.target_project).to equal(project)
<u>174</u>	<u>174</u>	end
	<u>175</u>	+
	<u>176</u>	+ it 'has MWPS set to false' do
	<u>177</u>	+ expect(created_object.merge_when_pipeline_succeeds).to eq(false)
	<u>178</u>	+ end
<u>175</u>	<u>179</u>	end
<u>176</u>	<u>180</u>	
<u>177</u>	<u>181</u>	context 'issue object' do
	↑ ↓	@@ -297,6 +301,38 @@ def values
<u>297</u>	<u>301</u>	end
<u>298</u>	<u>302</u>	end
<u>299</u>	<u>303</u>	
	<u>304</u>	+ context 'pipeline setup' do
	<u>305</u>	+ let(:relation_sym) { :ci_pipelines }
	<u>306</u>	+ let(:relation_hash) do
	<u>307</u>	+ {
	<u>308</u>	+ "id" => 1,
	<u>309</u>	+ "status" => status
	<u>310</u>	+ }
	<u>311</u>	+ end
	<u>312</u>	+
	<u>313</u>	+ subject { created_object }

```

314 +
315 +   ::Ci::HasStatus::COMPLETED_STATUSES.each do |status|
316 +     context "when relation_hash has a completed status of #
      {status}" do
317 +       let(:status) { status }
318 +
319 +       it "does not change the created object status" do
320 +         expect(created_object.status).to eq(status)
321 +       end
322 +     end
323 +   end
324 +
325 +   ::Ci::HasStatus::CANCELABLE_STATUSES.each do |status|
326 +     context "when relation_hash has cancelable status of #{status}"
      do
327 +       let(:status) { status }
328 +
329 +       it "sets the created object status to canceled" do
330 +         expect(created_object.status).to eq('canceled')
331 +       end
332 +     end
333 +   end
334 + end
335 +
300 336 context 'pipeline_schedule' do
301 337   let(:relation_sym) { :pipeline_schedules }
302 338   let(:relation_hash) do

```



```

↑ @@ -135,6 +135,7 @@ def match_mr1_note(content_regex)
135 135   expect(pipeline.merge_request.id).to be > 0
136 136   expect(pipeline.merge_request.target_branch).to
      eq('feature')
137 137   expect(pipeline.merge_request.source_branch).to
      eq('feature_conflict')
138 +
      expect(pipeline.merge_request.merge_when_pipeline_succeeds).to
      eq(false)
138 139   end
139 140
140 141   it 'restores pipelines based on ascending id order' do
      ↓ @@ -146,6 +147,9 @@ def match_mr1_note(content_regex)
146 147       5f923865dde3436854e9ceb9cdb7815618d4e849
147 148       d2d430676773caa88cdf7c55944073b2fd5561a
148 149       2ea1f3dec713d940208fb5ce4a38765ecb5d3f73
149 150 +       1b6c4f044c63217d1ed06e514c84d22871bed912
150 151 +       ded178474ef2ba1f80a9964ba15da3ddb3cf664b
151 152 +       fd459e5c514d70dc525c5e70990ca5e0debb3105
149 153   ]
150 154
151 155   project = Project.find_by_path('project')
      ↑ ↓ @@ -297,6 +301,12 @@ def match_mr1_note(content_regex)
297 301   end
298 302   end
299 303
304 +   it 'sets MWPS to false for all merge requests' do
305 +     MergeRequest.find_each do |merge_request|
306 +       expect(merge_request.merge_when_pipeline_succeeds).to
      eq(false)
307 +     end
308 +   end
309 +
300 310   it 'has multiple merge request assignees' do
301 311     expect(MergeRequest.find_by(title: 'MR1').assignees).to
      contain_exactly(@user, *@existing_members)
302 312     expect(MergeRequest.find_by(title: 'MR2').assignees).to
      be_empty
      ↑ ↓ @@ -545,9 +555,9 @@ def match_mr1_note(content_regex)
545 555   end
546 556
547 557   it 'has the correct number of pipelines and statuses' do
548 -     expect(@project.ci_pipelines.size).to eq(7)
549 +     expect(@project.ci_pipelines.size).to eq(10)
549 559

```

```

550 - @project.ci_pipelines.order(:id).zip([2, 0, 2, 3, 2, 2,
0])
560 + @project.ci_pipelines.order(:id).zip([2, 0, 2, 3, 2, 8, 0,
0, 0, 0])
551 561 .each do |(pipeline, expected_status_size)|
552 562 expect(pipeline.statuses.size).to
eq(expected_status_size)
553 563 end
↕ @@ -555,28 +565,58 @@ def match_mr1_note(content_regex)
555 565 end
556 566
557 567 context 'when restoring hierarchy of pipeline, stages and
jobs' do
568 + context 'pipelines' do
558 569 it 'restores pipelines' do
559 - expect(Ci::Pipeline.all.count).to be 7
570 + expect(Ci::Pipeline.all.count).to be 10
571 + end
572 +
573 + it 'marks cancelable pipelines as canceled' do
574 + expect(Ci::Pipeline.where(status: 'canceled').count).to
eq 7
575 + end
560 576 end
561 577
578 + context 'stages' do
562 579 it 'restores pipeline stages' do
563 - expect(Ci::Stage.all.count).to be 6
580 + expect(Ci::Stage.all.count).to be 7
581 + end
582 +
583 + it 'marks cancelable stages as canceled' do
584 + expect(Ci::Stage.where(status: 'canceled').count).to eq
6
564 585 end
565 586
566 587 it 'correctly restores association between stage and a
pipeline' do
567 588 expect(Ci::Stage.all).to
all(have_attributes(pipeline_id: a_value > 0))
568 589 end
590 + end
569 591
592 + context 'builds' do
570 593 it 'restores builds' do
571 594 expect(Ci::Build.all.count).to be 7
572 595 end
573 596
597 + it 'marks cancelable builds as canceled' do
598 + expect(Ci::Build.where(status: 'canceled').count).to eq
3
599 + end
600 + end
601 +
602 + context 'bridges' do
574 603 it 'restores bridges' do
575 - expect(Ci::Bridge.all.count).to be 1
604 + expect(Ci::Bridge.all.count).to be 5
576 605 end
577 606
607 + it 'marks cancelable bridges as canceled' do
608 + expect(Ci::Bridge.where(status: 'canceled').count).to eq
4
609 + end
610 + end
611 +
612 + context 'generic commit statuses' do
578 613 it 'restores generic commit statuses' do
579 - expect(GenericCommitStatus.all.count).to be 1
614 + expect(GenericCommitStatus.all.count).to be 3
615 + end
616 +
617 + it 'marks cancelable generic commit statuses as canceled'
do

```

```

618 + expect(GenericCommitStatus.where(status:
619 +   'canceled').count).to eq 2
580 620 end
581 621
582 622 it 'correctly restores association between a stage and a
job' do
↓

```

```

↑ @@ -127,6 +127,10 @@
127 127 expect(system_notes.size).to eq(1)
128 128 expect(system_notes.first['note']).to eq('merged')
129 129 end
130 +
131 + it 'has no merge_when_pipeline_succeeds' do
132 +   expect(subject.first['merge_when_pipeline_succeeds']).to
be_nil
133 + end
130 134 end
131 135
132 136 context 'with snippets' do
↑ ↓ @@ -301,6 +305,14 @@
301 305
302 306 it { is_expected.not_to be_empty }
303 307 end
308 +
309 + context 'with pipeline schedules' do
310 +   let(:relation_name) { :pipeline_schedules }
311 +
312 +   it 'has no owner_id' do
313 +     expect(subject.first['owner_id']).to be_nil
314 +   end
315 + end
304 316 end
305 317
306 318 describe '#saves project tree' do
↑ ↓ @@ -535,6 +547,7 @@ def setup_project
535 547
536 548 design = create(:design, :with_file, versions_count: 2, issue:
issue)
537 549 create(:diff_note_on_design, noteable: design, project: project,
author: user)
550 + create(:ci_pipeline_schedule, project: project, owner: user)
538 551
539 552 project
540 553 end
↓

```

Please [register](#) or [sign in](#) to reply