

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

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



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

↑		@@ -98,6 +98,7 @@ def setup_models
<u>98</u>	<u>98</u>	when :ProtectedBranch::PushAccessLevel then
		setup_protected_branch_access_level
<u>99</u>	<u>99</u>	when :ApprovalProjectRulesProtectedBranch then
		setup_merge_approval_protected_branch
<u>100</u>	<u>100</u>	when :releases then setup_release
	<u>101</u>	+ when :merge_requests, :MergeRequest, :merge_request then
		setup_merge_request
<u>101</u>	<u>102</u>	end
<u>102</u>	<u>103</u>	
<u>103</u>	<u>104</u>	update_project_references
↑	↓	@@ -148,22 +149,30 @@ def setup_diff
<u>148</u>	<u>149</u>	end
<u>149</u>	<u>150</u>	
<u>150</u>	<u>151</u>	def setup_pipeline
	<u>152</u>	+ @relation_hash['status'] =
		transform_status(@relation_hash['status'])
	<u>153</u>	+
<u>151</u>	<u>154</u>	@relation_hash.fetch('stages', []).each do stage
	<u>155</u>	+ stage.status = transform_status(stage.status)
	<u>156</u>	+
<u>152</u>	<u>157</u>	# old export files have statuses
<u>153</u>	-	stage.statuses.each do status
<u>154</u>	-	status.pipeline = imported_object
	<u>158</u>	+ stage.statuses.each do job
	<u>159</u>	+ job.status = transform_status(job.status)
	<u>160</u>	+ job.pipeline = imported_object
<u>155</u>	<u>161</u>	end
<u>156</u>	<u>162</u>	
<u>157</u>	-	stage.builds.each do status
<u>158</u>	-	status.pipeline = imported_object
	<u>163</u>	+ stage.builds.each do job
	<u>164</u>	+ job.status = transform_status(job.status)
	<u>165</u>	+ job.pipeline = imported_object
<u>159</u>	<u>166</u>	end
<u>160</u>	<u>167</u>	
<u>161</u>	-	stage.bridges.each do status
<u>162</u>	-	status.pipeline = imported_object
	<u>168</u>	+ stage.bridges.each do job
	<u>169</u>	+ job.status = transform_status(job.status)
	<u>170</u>	+ 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
```



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