

```
1 # frozen_string_literal: true
2
3 require 'redmine'
4 require_relative 'lib/issue_details_hook_listener'
5
6 yaml_data = YAML.safe_load(ERB.new(Rails.root.join('plugins
7 /gnosis/config/application.yml')).read).result)
8 ENV = ActiveSupport::HashWithIndifferentAccess.new(
9   yaml_data)
10
11 if ENV['GITHUB_ACCESS_TOKEN'].blank?
12   raise 'GITHUB_ACCESS_TOKEN is not set'
13 elsif ENV['GITHUB_ACCESS_TOKEN'] == 'your_token'
14   Rails.logger.warn 'GITHUB_ACCESS_TOKEN is default value'
15 end
16
17 Redmine::Plugin.register :gnosis do
18   name 'Gnosis plugin'
19   author 'Anes Hodza'
20   description 'This Plugin allows you to see the status of
21   issues in a project'
22   version '1.0.0'
23   url 'https://github.com/aneshodza/gnosis/'
24   author_url 'https://www.aneshodza.ch/'
```

```

1 # gnosis
2
3 Welcome to gnosis!
4 This is a Redmine Plugin with the goal of making your
development process easier to keep an overview of.
5
6 ## What does this plugin do?
7 Gnosis is able to show you GitHub Pull Requests and
Deployments that belong to a Redmine issue, all in its "details"
page.
8 To make this magic work there are only a few steps you have
to follow and certain conventions to keep in mind.
9
10 #### GitHub Webhooks configuration
11 To configure the GitHub Webhooks, go to github.com/your\_org/your\_repo/settings/hooks. There you click on
12 "Add Webhook". The "Payload URL" is your Redmine URL + /
github_webhook. The secret is up for you to decide. It's
13 important that this string is complex and secure. Then, you
get into your `config/application.yml`, where you add the
14 key as your `GITHUB_WEBHOOK_SECRET`. Click "Let me select
individual events." and choose "Pull Requests".
15 You should now be good to go!
16
17 #### SemaphoreCI Webhooks configuration
18 **Important:** The GitHub notifications work without the
SemaphoreCI setup. If you don't use a CI or just a
different
19 you can skip this step.
20 To set up the SemaphoreCI Webhook go to your\_org.semaphoreci.com/notifications and click on "New
Notification".
21 Attributes like "Name of the notification" can be chosen
freely. What you need to setup is: Listing your project
under
22 "in projects", having `.*-deploy\.yml/` under Pipelines (this
just tells the notification to send data every time a
23 deploy script is done running), adding your Redmine URL + /
semaphore_webhook to "Endpoint" and typing `WEBHOOK_SECRET`
24 into the "Secret name" field.
25 Then go to your\_org.semaphoreci.com/notifications and click
on "New Secret". The "Name of the Secret" should be
26 `WEBHOOK_SECRET`. Then you create an environment variable
with the "Variable Name" `WEBHOOK_SECRET`. The "Value" is
the
27 secret. This should match the `SEMAPHORE_WEBHOOK_SECRET` in
your `config/application.yml`.
28 This should now also work just fine!
29
30

```

```
31 ### GitHub Access Token
32 This app needs to query your repository every now and then
   , so its able to know which deployment belongs to which
   pull
33 request. For that you need to set the `GITHUB_ACCESS_TOKEN
   ` in your `config/application.yml`. To create it, go to
34 github.com/settings/tokens and click on "Generate new token"
   ". There click the "(classic)" option. It only needs "repo"
35 access to work.
36 If you make a deploy, it should correctly show now.
37
38 ### Conventions you need to keep
39 The plugin needs information on which Pull Requests belong
   to which issues, which is why there are set conventions for
40 branch naming. When you create a branch that is connected
   to a certain Issue it should be named as follows:
41 something/issue_number-description. The only important part
   of this is the /issue_number as that's how
42 the plugin is able to connect PRs to issues.
43
44
```

```
1 coverage  
2 application.yml  
3
```

```
1 = gnosis
2
3 Welcome to gnosis!
4 This is a Redmine Plugin with the goal of making your
   development process easier to keep an overview of.
5
6 == What does this plugin do?
7 Gnosis is able to show you GitHub Pull Requests and
   Deployments that belong to a Redmine issue, all in its "details" page.
8 To make this magic work there are only a few steps you have
   to follow and certain conventions to keep in mind. For
   additional
9 info, refer to the official README.md.
10
```

```

1 inherit_from: .rubocop_todo.yml
2
3 AllCops:
4   TargetRubyVersion: 2.4
5   TargetRailsVersion: 5.2
6
7   NewCops: enable
8
9   Exclude:
10  - '**/vendor/**/*'
11  - '**/tmp/**/*'
12  - '**/bin/**/*'
13  - '**/extra/**/*'
14  - '**/lib/generators/**/templates/*'
15  - '**/lib/tasks/**/*'
16  - '**/files/**/*'
17  - 'db/schema.rb'
18
19 # Enable extensions
20
21 require:
22   - rubocop-performance
23   - rubocop-rails
24
25 # Rules for Redmine
26
27 Rails/ActionControllerTestCase:
28   Enabled: false
29
30 Bundler/OrderedGems:
31   Enabled: false
32
33 Layout/CaseIndentation:
34   Exclude:
35     - 'config/initializers/10-patches.rb'
36
37 Layout/ClosingParenthesisIndentation:
38   Enabled: true
39   Exclude:
40     - 'lib/redmine/wiki_formatting/textile/redcloth3.rb'
41
42 Layout/EmptyLinesAroundAccessModifier:
43   Enabled: true
44   Exclude:
45     - 'config/initializers/10-patches.rb'
46
47 Layout/EmptyLineBetweenDefs:
48   AllowAdjacentOneLineDefs: true
49
50 Layout/FirstHashElementIndentation:

```

```

51  Enabled: true
52  Exclude:
53    - 'lib/redmine/wiki_formatting/textile/redcloth3.rb'
54
55 Layout/LineLength:
56  Enabled: true
57  Exclude:
58    - 'db/migrate/0*.rb'
59    - 'db/migrate/1*.rb'
60    - 'db/migrate/201*.rb'
61    - 'db/migrate/20200*.rb'
62    - 'lib/redmine/wiki_formatting/textile/redcloth3.rb'
63
64 Layout/RescueEnsureAlignment:
65  Enabled: true
66  Exclude:
67    - 'db/migrate/
085_add_role_tracker_old_status_index_to_workflows.rb'
68
69 Layout/SpaceAroundEqualsInParameterDefault:
70  Enabled: false
71
72 Layout/SpaceBeforeSemicolon:
73  Enabled: true
74  Exclude:
75    - 'config/initializers/10-patches.rb'
76
77 Layout/SpaceBeforeBlockBraces:
78  # "space" is used more than "no_space".
79  # But "no_space" is more natural in one liner.
80  #   str = path.to_s.split(%r{[/\|]}).select{|p| !p.blank?
?}.join("/")
81  Enabled: false
82
83 Layout/SpaceInsideBlockBraces:
84  Enabled: false
85
86 Layout/SpaceInsideHashLiteralBraces:
87  Enabled: false
88
89 Lint/RaiseException:
90  Enabled: true
91
92 Lint/StructNewOverride:
93  Enabled: true
94
95 Lint/SuppressedException:
96  AllowComments: true
97
98 Layout/TrailingWhitespace:

```

```
99  AllowInHeredoc: true
100
101 Metrics:
102   Enabled: false
103
104 Naming/AccessorMethodName:
105   Enabled: false
106
107 Naming/VariableNumber:
108   CheckMethodNames: false
109   CheckSymbols: false
110
111 Naming/BinaryOperatorParameterName:
112   Enabled: false
113
114 Naming/PredicateName:
115   Enabled: false
116
117 Rails/BulkChangeTable:
118   Exclude:
119     - 'db/migrate/20120714122200_add_workflows_rule_fields
. rb'
120     - 'db/migrate/
20131214094309_remove_custom_fields_min_max_length_default
_values.rb'
121
122 Rails/HelperInstanceVariable:
123   Enabled: false
124
125 Style/AsciiComments:
126   Exclude:
127     # Copyright credit has non ascii character.
128     # We can not change nor remove it.
129     - 'app/models/repository/git.rb'
130
131 Style/BlockComments:
132   Enabled: true
133   Exclude:
134     - 'lib/diff.rb'
135
136 Style/BlockDelimiters:
137   Enabled: true
138   Exclude:
139     - 'db/migrate/007_create_journals.rb'
140     - 'lib/diff.rb'
141
142 Style/EmptyElse:
143   EnforcedStyle: empty
144
145 Style/FormatString:
```

```
146 EnforcedStyle: sprintf
147
148 Style/FormatStringToken:
149   Enabled: false
150
151 Style/FrozenStringLiteralComment:
152   Enabled: true
153   EnforcedStyle: always
154   Exclude:
155     - 'db/**/*.rb'
156     - 'Gemfile'
157     - 'Rakefile'
158     - 'config.ru'
159     - 'config/additional_environment.rb'
160
161 Style/HashEachMethods:
162   Enabled: true
163
164 Style/HashSyntax:
165   Enabled: true
166   EnforcedStyle: no_mixed_keys
167
168 Style/HashTransformKeys:
169   Enabled: true
170
171 Style/HashTransformValues:
172   Enabled: true
173
174 Style/IdenticalConditionalBranches:
175   Exclude:
176     - 'config/initializers/10-patches.rb'
177     - 'lib/redmine/wiki_formatting/textile/redcloth3.rb'
178
179 Style/NegatedIfElseCondition:
180   Enabled: false
181
182 Style/RaiseArgs:
183   Enabled: false
184
185 Style/TrailingCommaInArrayLiteral:
186   Enabled: false
187
188 Style/TrailingCommaInHashLiteral:
189   Enabled: false
190
191 Style/Documentation:
192   Enabled: false
193
194 Rails/CreateTableWithTimestamps:
195   Enabled: false
```

```
1 source 'https://rubygems.org'  
2 git_source(:github) { |repo| "https://github.com/#{repo}.  
  git" }  
3  
4 group :test, :development do  
5   gem 'brakeman'  
6   gem 'factory_bot_rails'  
7   gem 'faker'  
8   gem 'byebug'  
9 end  
10  
11 gem 'octokit'  
12
```

```
1 # This configuration was generated by
2 # `rubocop --auto-gen-config`
3 # on 2023-05-25 14:47:51 UTC using RuboCop version 1.50.2.
4 # The point is for the user to remove these configuration
5 # records
6 # one by one as the offenses are removed from the code base
7 # Note that changes in the inspected code, or installation
8 # of new
# versions of RuboCop, may require this file to be
# generated again.
```

```
1 GEM
2   remote: https://rubygems.org/
3   specs:
4     activesupport (7.0.5)
5       concurrent-ruby (~> 1.0, >= 1.0.2)
6       i18n (>= 1.6, < 2)
7       minitest (>= 5.1)
8       tzinfo (~> 2.0)
9     ast (2.4.2)
10    brakeman (6.0.0)
11    concurrent-ruby (1.2.2)
12    i18n (1.13.0)
13      concurrent-ruby (~> 1.0)
14    json (2.6.3)
15    minitest (5.18.0)
16    parallel (1.23.0)
17    parser (3.2.2.1)
18      ast (~> 2.4.1)
19    rack (3.0.7)
20    rainbow (3.1.1)
21    regexp_parser (2.8.0)
22    rexml (3.2.5)
23    rubocop (1.50.2)
24      json (~> 2.3)
25      parallel (~> 1.10)
26      parser (>= 3.2.0.0)
27      rainbow (>= 2.2.2, < 4.0)
28      regexp_parser (>= 1.8, < 3.0)
29      rexml (>= 3.2.5, < 4.0)
30      rubocop-ast (>= 1.28.0, < 2.0)
31      ruby-progressbar (~> 1.7)
32      unicode-display_width (>= 2.4.0, < 3.0)
33    rubocop-ast (1.28.1)
34      parser (>= 3.2.1.0)
35    rubocop-performance (1.17.1)
36      rubocop (>= 1.7.0, < 2.0)
37      rubocop-ast (>= 0.4.0)
38    rubocop-rails (2.19.1)
39      activesupport (>= 4.2.0)
40      rack (>= 1.1)
41      rubocop (>= 1.33.0, < 2.0)
42    ruby-progressbar (1.13.0)
43    tzinfo (2.0.6)
44      concurrent-ruby (~> 1.0)
45    unicode-display_width (2.4.2)
46
47 PLATFORMS
48   arm64-darwin-21
49
50 DEPENDENCIES
```

```
51 brakeman
52 rubocop (~> 1.50.0)
53 rubocop-performance (~> 1.17.1)
54 rubocop-rails (~> 2.19.1)
55
56 BUNDLED WITH
57   2.4.7
58
```

```
1 class CreateDeployments < ActiveRecord::Migration[6.1]
2   def change
3     create_table :deployments do |t|
4       t.string :deploy_branch
5       t.string :url
6       t.boolean :has_passed, default: false, null: false
7       t.references :pull_request, null: false, foreign_key
8       : true
9       t.timestamps
10      end
11    end
12  end
13 end
14
```

```
1 class CreatePullRequests < ActiveRecord::Migration[6.1]
2   def change
3     create_table :pull_requests do |t|
4       t.string :state
5       t.string :url
6       t.string :title
7       t.string :source_branch
8       t.string :target_branch
9       t.string :merge_commit_sha
10      t.boolean :was_merged, default: false, null: false
11      t.references :issue, null: false, foreign_key: true
12
13      t.timestamps
14    end
15  end
16 end
17
```

```
1 # frozen_string_literal: true
2
3 class Deployment < GnosisApplicationRecord
4   belongs_to :pull_request
5   has_one :issue, through: :pull_request, source: :issue
6
7   def self.auto_create_or_update(branch, pull_request_id,
8     url, has_passed, ci_date_string)
9     ci_date = Time.zone.parse(ci_date_string)
10    deploy = Deployment.find_or_initialize_by(deploy_branch
11      : branch, pull_request_id: pull_request_id)
12    deploy.update!(url: url, has_passed: has_passed,
13    ci_date: ci_date)
14  end
15 end
16
17
```

```
1 # frozen_string_literal: true
2
3 class PullRequest < GnosisApplicationRecord
4   belongs_to :issue
5   has_many :deployments, dependent: :destroy
6
7   def self.auto_create_or_update(params)
8     state = params[:pull_request]][:merged] ? 'merged' :
9       params[:pull_request][:state]
10    state = 'draft' if params[:pull_request][:draft]
11    pr = PullRequest.find_or_initialize_by(url: params[:pull_request][:html_url])
12    pr.update!(state: state, url: params[:pull_request][:html_url],
13               title: params[:pull_request][:title],
14               source_branch: params[:pull_request][:head][:ref],
15               target_branch: params[:pull_request][:base][:ref],
16               was_merged: params[:pull_request][:merged],
17               merge_commit_sha: params[:pull_request][:merge_commit_sha],
18               issue_id: params[:issue_id])
19   end
20 end
21
22
```

```
1 # frozen_string_literal: true
2
3 # rubocop:disable Rails/ApplicationRecord
4 class GnosisApplicationRecord < ActiveRecord::Base
5   # rubocop:enable Rails/ApplicationRecord
6   self.abstract_class = true
7 end
8
```

```
1 # frozen_string_literal: true
2
3 module WebhooksHelper
4 end
5
```

```

1 # frozen_string_literal: true
2
3 class WebhooksController < ApplicationController
4   protect_from_forgery except: %i[github_webhook_catcher
5   semaphore_webhook_catcher]
6
7   Octokit.configure do |config|
8     config.access_token = ENV.fetch('GITHUB_ACCESS_TOKEN',
9     nil)
10    end
11    CLIENT = Octokit::Client.new
12
13    def github_webhook_catcher
14      unless verify_signature(request.body.read, request.env[
15        'HTTP_X_HUB_SIGNATURE_256'],
16        ENV.fetch(
17          'GITHUB_WEBHOOK_SECRET', nil))
18        p 'Signature mismatch'
19        return render json: {status: 403}, status: :forbidden
20      end
21
22      github_webhook_handler(params)
23
24      render json: {status: :ok}
25    end
26
27    def semaphore_webhook_catcher
28      unless verify_signature(request.body.read, "sha256=#{request.headers['X-Semaphore-Signature-256']}",
29        ENV.fetch(
30          'SEMAPHORE_WEBHOOK_SECRET', nil))
31        return render json: {status: 403}, status: :forbidden
32      end
33
34      semaphore_webhook_handler(params)
35
36      render json: {status: :ok}
37    end
38
39    private
40
41    def github_webhook_handler(params)
42      numbers = params[:pull_request][:head][:ref].match(%r{
43        /(\d+)} || []
44
45      return unless numbers.length.positive? && Issue.exists?
46      (id: numbers[1])
47
48      PullRequest.auto_create_or_update(params.merge(issue_id
49      : numbers[1]))

```

```

42   end
43
44   def semaphore_webhook_handler(params)
45     range = params[:revision][:branch][:commit_range]
46     branch = params[:revision][:branch][:name]
47     repo = params[:repository][:slug]
48     passed = params[:pipeline][:result] == 'passed'
49     time = params[:pipeline][:done_at]
50
51     first_sha = range.split('...').first
52     last_sha = range.split('...').last
53
54     sha_between = fetch_commit_history(repo, branch,
55                                         first_sha, last_sha)
55     create_deploys_for_pull_requests(sha_between, branch,
56                                     passed, time)
56   end
57
58   def verify_signature(payload_body, received_signature,
59                         secret)
59     signature = "sha256=#{OpenSSL::HMAC.hexdigest(OpenSSL::
60                               Digest.new('sha256'), secret, payload_body)}"
60     p "Signature: #{signature}"
61     p "Received Signature: #{received_signature}"
62     Rack::Utils.secure_compare(signature,
63                               received_signature)
63   end
64
65   def fetch_commit_history(repo, branch, first_commit,
66                           last_commit)
66     commit_sha_list = CLIENT.commits(repo, branch).pluck(:sha)
67
68     first_commit_index = commit_sha_list.index(first_commit)
69     last_commit_index = commit_sha_list.index(last_commit)
70
71     commit_sha_list[last_commit_index..first_commit_index]
72   end
73
74   def create_deploys_for_pull_requests(sha_between, branch,
75                                         passed, time)
75     sha_between.each do |sha|
76       pr = PullRequest.find_by(merge_commit_sha: sha)
77       next unless pr
78
79       Deployment.auto_create_or_update(branch, pr.id, url,
80                                       passed, time)
80     end
81   end

```

```
82
83   def url
84     "https://#{params\[:organization\]}\[:name\].semaphoreci.com/workflows/#{{params\[:workflow\]}\[:id\]}/"
85   end
86 end
87
```

```
1 #!/bin/bash
2
3 white='\033[1;37m'
4 red='\033[0;31m'
5 green='\033[0;32m'
6
7 rake redmine:plugins:test
8
9 if [[ $? -eq 0 ]]; then
10   echo -e "${green}"
11   echo -e "-----"
12   echo -e "Tests passed"
13 else
14   echo -e "${red}"
15   echo -e "-----"
16   echo -e "Tests failed"
17
18 fi
19
```

```
1 #!/bin/bash
2
3 cp config/application.example.yml config/application.yml
4
5 yarn install
6 bundle exec rake redmine:plugins:migrate
7
```

```

1 #!/bin/bash
2
3 white='\033[1;37m'
4 red='\033[0;31m'
5 green='\033[0;32m'
6
7 function return_code {
8   if [ $? -ne 0 ]; then
9     echo -e "${red} Error: $1 found errors! ${white}"
10    failing_checks+=("$1")
11    passing=false
12  fi
13 }
14
15 passing=true
16 failing_checks=()
17
18 echo "Executing fastchecks..."
19
20 if [[ $* == *--fix* ]]; then
21   echo "Fixing..."
22   bundle exec rubocop --config .rubocop.yml
23   return_code "Rubocop"
24 else
25   echo "Note: You can use --fix to fix the errors
26 automatically"
27   bundle exec rubocop --config .rubocop.yml
28   return_code "Rubocop"
29 fi
30
31 bundle exec brakeman -q -z --no-summary --no-pager --force
32 return_code "Brakeman"
33
34 if [ "$passing" = true ]; then
35   echo -e "${green}
36   echo -e "-----${white}"
37   exit 0
38 else
39   echo -e "${red}-----"
40   echo -e "Some checks failed"
41   echo -e "Failed checks: ${failing_checks[@]}"
42   echo -e "-----${white}"
43   exit 1
44 fi
45

```

File - /Users/anes/c/redmine-deploy-scheduler/plugins/gnosis/lib/issue_details_hook_listener.rb

```
1 # frozen_string_literal: true
2
3 class NewSectionHookListener < Redmine::Hook::ViewListener
4   def view_issues_show_description_bottom(context={})
5     @context = context
6     setup
7     <<-HTML
8       <hr/>
9       <p><strong>Pull Requests</strong></p>
10      #{@pr_string.length.positive? ? "<ul>#{@pr_string}</ul>" : 'There are currently no PRs open for this issue'}
11     HTML
12   end
13
14   private
15
16   def setup
17     get_prs
18     get_deployments
19     set_deployment_strings
20     set_pr_string
21   end
22
23   def get_prs
24     @prs = PullRequest.where(issue_id: @context[:issue].id)
25   end
26
27   def get_deployments
28     @deployments = @prs.map do |pr|
29       Deployment.where(pull_request_id: pr['id']).to_a
30     end
31   end
32
33   def set_deployment_strings
34     @deployments_strings = []
35     @deployments.each do |deployment_list|
36       formatted_deployment_list = []
37       deployment_list.each do |deployment|
38         formatted_deployment_list << <<-LISTOBJECT
39           <li>
40             <a href="#{deployment['url']}' target='_blank'
41               id='deployment-#{deployment['id']}'>
42               on "#{deployment['deploy_branch']}@"
43               at "#{deployment['ci_date'].strftime('%d.%m.%Y
44               at %I:%M%p UTC')}@"
45             #{deployment['has_passed']} ? ' ' : ' '
46           </a>
47           </li>
48         LISTOBJECT
```

```
47      end
48      @deployments_strings << formatted_deployment_list
49    end
50  end
51
52  def set_pr_string
53    @pr_string = @prs.each_with_index.map do |pr, index|
54      formatted_deployments_list = @deployments_strings[
55        index].join
56      "LISTOBJECT
57      <li>
58        <a href='#{pr['url']}' target='_blank' id='pr-#{pr['id']}'>#{pr['title']} (#{'#{pr['state']}'})</a> <br/>
59        #{formatted_deployments_list.length.positive? ? '<
60          strong>Deployments:</strong>' : ''}
61        <ul>
62          #{formatted_deployments_list}
63        </ul>
64      </li>
65    LISTOBJECT
66  end.join
67 end
```

File - /Users/anes/c/redmine-deploy-scheduler/plugins/gnosis/test/test_helper.rb

```
1 # frozen_string_literal: true
2
3 require 'simplecov'
4
5 ENV['GOOGLE_CHROME_OPTS_ARGS'] = 'headless,disable-gpu,no-
6 sandbox,disable-dev-shm-usage'
7 ENV['GITHUB_WEBHOOK_SECRET'] = 'test'
8 ENV['SEMAPHORE_WEBHOOK_SECRET'] = 'test'
9
10 SimpleCov.coverage_dir('plugins/gnosis/coverage')
11 SimpleCov.start do
12   add_filter do |source_file|
13     source_file.lines.count < 7
14   end
15
16   add_filter do |source_file|
17     source_file.filename.exclude?('gnosis/app') &&
18     source_file.filename.exclude?('gnosis/lib')
19   end
20 end
21
22 SimpleCov.minimum_coverage 100
23
24 FactoryBot.definition_file_paths = [File.expand_path(
25   'factories', __dir__)]
26 FactoryBot.find_definitions
27
28 # Load the lib folder
29 Dir[Rails.root.join('plugins/gnosis/lib/**/*.rb')].each { |f|
30   f| load f }
31
32 # Load the Redmine helper
33 require File.expand_path("#{File.dirname(__FILE__)}"
34   '/../..../test/test_helper")
35
36 # Create seeds
37 IssueStatus.create!(name: 'To start', default_done_ratio: 0)
38 Tracker.create!(name: 'Feature', default_status: IssueStatus.first)
39 IssuePriority.create!(name: 'Normal', is_default: true)
40
41 User.create!(
42   login: 'admin',
43   mail: 'admin@example.com',
44   password: '12345678',
45   password_confirmation: '12345678',
46   admin: true,
47   firstname: Faker::Name.first_name,
48   lastname: Faker::Name.last_name,
```

```
44   status: Principal::STATUS_ACTIVE
45 )
46
47 name = Faker::Lorem.word
48 Project.create!(name: name, identifier: name.downcase,
  is_public: false, description: '..', issues: [
49   Issue.new(subject: Faker::Lorem.word,
    description: '..', tracker: Tracker.first, author: User.
    first,
      status: IssueStatus.first,
    priority: IssuePriority.first),
50   Issue.new(subject: Faker::Lorem.word,
    description: '..', tracker: Tracker.first, author: User.
    first,
      status: IssueStatus.first,
    priority: IssuePriority.first)
51   ])
52
53
54
```

```
1 # frozen_string_literal: true
2
3 require 'test_helper'
4 require File.expand_path("#{File.dirname(__FILE__)}\n    ../../../../test/application_system_test_case")
5
6 class GnosisSystemTest < ApplicationSystemTestCase
7   def login
8     visit '/login'
9     fill_in 'username', with: 'admin'
10    fill_in 'password', with: '12345678'
11    click_button 'Login'
12  end
13 end
14
```

```
1 # frozen_string_literal: true
2
3 require_relative '../test_helper'
4
5 class DeploymentTest < ActiveSupport::TestCase
6   def test_belongs_to_pull_request
7     FactoryBot.create(:deployment)
8     assert_equal 1, PullRequest.where(id: Deployment.first.
9       pull_request_id).count
10    end
11
12  def
13    test_auto_create_or_update_updates_existing_deployment
14      FactoryBot.create(:pull_request)
15      deploy = FactoryBot.create(:deployment)
16
17      assert_difference 'Deployment.count', 0 do
18        Deployment.auto_create_or_update(deploy.deploy_branch,
19          , deploy.pull_request_id, 'aneshodza.ch',
20            !deploy.has_passed,
21            deploy.ci_date.to_s)
22      end
23
24      assert_equal deploy.reload.url, 'aneshodza.ch'
25  end
26
27  def test_auto_create_or_update_new_branch
28    FactoryBot.create(:pull_request)
29    deploy = FactoryBot.create(:deployment)
30
31    assert_difference 'Deployment.count', 1 do
32      Deployment.auto_create_or_update("some branch that
33      doesn't exist", deploy.pull_request_id, deploy.url,
34      deploy.has_passed,
35      deploy.ci_date.to_s)
36    end
37
38    assert_equal 2, Deployment.count
39  end
40 end
41
```

```

1 # frozen_string_literal: true
2
3 require_relative '../test_helper'
4
5 class PullRequestTest < ActiveSupport::TestCase
6   def setup
7     @github_webhook_hash = {
8       pull_request: {
9         state: 'closed',
10        html_url: 'https://github.com/aneshodza/test-repo/
11        pull/17',
12        title: 'Create something',
13        head: {
14          ref: 'feature/1-some-feature'
15        },
16        base: {
17          ref: 'main'
18        },
19        merged: true,
20        merge_commit_sha: '19a89f0050eacf201ccd058d5e28cddf2b035bfc'
21      },
22      issue_id: 1
23    }
24  end
25
26  def test_create_pull_request
27    assert_difference('PullRequest.count', 1) do
28      PullRequest.auto_create_or_update(@
29      github_webhook_hash)
30    end
31
32    pr = PullRequest.last
33    assert_equal 'merged', pr.state
34  end
35
36  def test_update_pull_request
37    PullRequest.auto_create_or_update(@github_webhook_hash)
38    @github_webhook_hash[:pull_request][:state] = 'open'
39    @github_webhook_hash[:pull_request][:merged] = false
40    assert_difference('PullRequest.count', 0) do
41      PullRequest.auto_create_or_update(@
42      github_webhook_hash)
43    end
44
45  end
46

```

```

1 # frozen_string_literal: true
2
3 require File.expand_path("#{File.dirname(__FILE__)}../../../
gnosis_system_test")
4
5 class PullRequestListTest < GnosisSystemTest
6   def setup
7     login
8     @prs = []
9     3.times do |_i|
10       @prs << FactoryBot.create(:pull_request, issue_id: 1)
11     end
12     FactoryBot.create(:deployment, pull_request_id: @prs[0]
13     ].id)
13     FactoryBot.create(:deployment, pull_request_id: @prs[0]
13     ].id, deploy_branch: 'develop')
14     FactoryBot.create(:deployment, pull_request_id: @prs[1]
14     ].id, deploy_branch: 'develop')
15     visit 'issues/1'
16   end
17
18   def test_view_open
19     assert page.has_content?('Pull Requests')
20   end
21
22   def test_no_prs
23     visit 'issues/2'
24     assert page.has_content?('There are currently no PRs
open for this issue')
25   end
26
27   def test_prs_all_info_listed
28     @prs.each do |pr|
29       assert page.has_content?("#{pr.title} (#{}{pr.state})")
30       assert_includes page.find("#pr-#{pr.id}"['href']), pr
.url
31     end
32   end
33
34   def test_deployments_all_info_listed
35     @prs.each do |pr|
36       pr.deployments.each do |deployment|
37         assert page.has_content?(deployment.deploy_branch)
38         assert_includes page.find("#deployment-#{deployment
.id}"['href']), deployment.url
39         assert page.has_content?(deployment.ci_date.
strftime('%d.%m.%Y at %I:%M%p UTC'))
40         assert page.has_content?(deployment.has_passed ?
'': ''))
41     end

```

```
42      end
43    end
44 end
45
```

```
1 # frozen_string_literal: true
2
3 FactoryBot.define do
4   factory :deployment do
5     deploy_branch { 'main' }
6     url { 'example.com' }
7     has_passed { true }
8     ci_date { Time.zone.now }
9
10    pull_request { create(:pull_request) }
11  end
12 end
13
```

```
1 # frozen_string_literal: true
2
3 FactoryBot.define do
4   factory :pull_request do
5     state { %w[open close merged].sample }
6     url { 'example.com' }
7     title { Faker::Lorem.sentence }
8     source_branch { Faker::Lorem.word }
9     target_branch { 'main' }
10    was_merged { false }
11    merge_commit_sha { Faker::Lorem.characters(number: 40) }
12    issue { Issue.first || null }
13  end
14 end
15
```

File - /Users/anes/c/redmine-deploy-scheduler/plugins/gnosis/test/functional/webhooks_controller_test.rb

```
1 # frozen_string_literal: true
2
3 require_relative '../test_helper'
4
5 class WebhookCatchControllerControllerTest <
6   ActionController::TestCase
7   def setup
8     @controller = WebhooksController.new
9
10    Octokit::Client.any_instance.stubs(:commits).returns([
11      { sha: 'another_hash' },
12      { sha: 'in_between_hash' },
13      { sha: 'one_hash' }
14    ])
15
16    @github_webhook_hash = {
17      pull_request: {
18        state: 'closed',
19        html_url: 'https://github.com/aneshodza/test-repo/
pull/17',
20        title: 'Create something',
21        head: {
22          ref: 'feature/1-some-feature'
23        },
24        base: {
25          ref: 'main'
26        },
27        merged: true,
28        merge_commit_sha: '19a89f0050eacf201ccd058d5e28cddf2b035bfc',
29        draft: false
30      }
31    }
32
33    @semaphore_webhook_hash = {
34      workflow: {
35        id: '5432cce0-196d-4898-9385-c1d670e4a9e9',
36      },
37      revision: {
38        branch: {
39          name: 'main',
40          commit_range: 'one_hash...another_hash'
41        }
42      },
43      pipeline: {
44        result: 'passed',
45        done_at: '2021-03-03T12:00:00Z'
```

```

45      },
46      organization: {
47        name: 'aneshodza',
48      },
49      repository: {
50        slug: 'aneshodza/test-repo',
51      }
52    }
53
54    FactoryBot.build_list(:pull_request, 3).each(&:save!)
55    PullRequest.first.update!(merge_commit_sha: 'one_hash')
56    PullRequest.first(2).last.update!(merge_commit_sha: 'in_between_hash')
57    PullRequest.first(3).last.update!(merge_commit_sha: 'another_hash')
58  end
59
60  def test_create_pull_request
61    @request.headers['X-Hub-Signature-256'] =
62      'sha256='
63      09a2bb0a3c451c2dd5c2c227e05b0f2da426211b1b67c09bc724b1ea851
64      538ce'
65      assert_difference('PullRequest.count', 1) do
66        post :github_webhook_catcher, params: @
67        github_webhook_hash, as: :json
68        end
69
70        new_pr = PullRequest.last
71        assert_equal 'merged', new_pr.state
72        assert_equal 1, new_pr.issue_id
73      end
74
75      def test_update_pull_request_existing_url
76        PullRequest.auto_create_or_update(@github_webhook_hash,
77        merge(issue_id: Issue.first.id))
78        @request.headers['X-Hub-Signature-256'] =
79        'sha256='
80        4256f0ad4e317084d15e7fffeecb6e66667f45d12fcbd5ad0fecf4c0e318
81        02ce2'
82        @github_webhook_hash[:pull_request][:state] = 'open'
83        assert_difference('PullRequest.count', 0) do
84          post :github_webhook_catcher, params: @
85          github_webhook_hash, as: :json
86          end
87          assert @response.status == 200
88          assert 'closed', PullRequest.last.state
89        end
90
91        def test_create_pull_request_no_issue
92          @request.headers['X-Hub-Signature-256'] = 'sha256='

```

```

85      b15df5baab94e31571b043e810545ec49075dedb0dd7aa78b8f1855012
86      48c918'
87      @github_webhook_hash[:pull_request][:head][:ref] = 'feature/420-some-feature-no-issue'
88      assert_difference('PullRequest.count', 0) do
89          post :github_webhook_catcher, params: @
90          github_webhook_hash, as: :json
91      end
92  end
93
94  def test_create_pull_request_no_issue_in_branch_name
95      @request.headers['X-Hub-Signature-256'] = 'sha256=
96      2d44814be7f48dce09dc87544720819beab8be0747381d0c45c4301f1e
97      ef2a1b'
98      @github_webhook_hash[:pull_request][:head][:ref] = 'feature/some-feature-no-issue'
99      assert_difference('PullRequest.count', 0) do
100         post :github_webhook_catcher, params: @
101         github_webhook_hash, as: :json
102     end
103     # Just because there is no issue, doesn't mean it
104     # should fail
105     assert @response.status == 200
106 end
107
108 def test_pr_with_invalid_sha
109     @request.headers['X-Hub-Signature-256'] = 'sha256=
110     invalid'
111     assert_difference('PullRequest.count', 0) do
112         post :github_webhook_catcher, params: @
113         github_webhook_hash, as: :json
114     end
115     assert @response.status == 403
116 end
117
118 def test_create_deploys
119     @request.headers['X-Semaphore-Signature-256'] =
120     f88bf226e3fd7cbf28de748adfdd65a4372184d8daac01e2bd2aab1537
121     f9981d'
122     assert_difference('Deployment.count', 3) do
123         post :semaphore_webhook_catcher, params: @
124         semaphore_webhook_hash, as: :json
125     end
126 end
127
128 def test_deploy_with_invalid_sha
129     @request.headers['X-Semaphore-Signature-256'] =
130     invalid'
131     assert_difference('Deployment.count', 0) do

```

```
120      post :semaphore_webhook_catcher, params: @
121      semaphore_webhook_hash, as: :json
122    end
123    assert @response.status == 403
124  end
125  def test_deploy_no_pr
126    PullRequest.destroy_all
127    @request.headers['X-Semaphore-Signature-256'] =
128      f88bf226e3fd7cbf28de748adfdd65a4372184d8daac01e2bd2aab1537
129      f9981d'
130    assert_difference('Deployment.count', 0) do
131      post :semaphore_webhook_catcher, params: @
132      semaphore_webhook_hash, as: :json
133    end
134  end
135
```

```
1 # frozen_string_literal: true
2
3 # Plugin's routes
4 # See: http://guides.rubyonrails.org/routing.html
5 resources :example, only: [:index]
6
7 post 'github_webhook', to: 'webhooks#github_webhook_catcher'
8
9 post 'semaphore_webhook', to: 'webhooks#
semaphore_webhook_catcher'
```

```
1 GITHUB_WEBHOOK_SECRET: 'test'  
2 GITHUB_ACCESS_TOKEN: '  
3 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'  
3 SEMAPHORE_WEBHOOK_SECRET: 'test'  
4
```

```
1 GITHUB_WEBHOOK_SECRET: 'your_secret'  
2 GITHUB_ACCESS_TOKEN: 'your_token'  
3 SEMAPHORE_WEBHOOK_SECRET: 'your_secret'  
4
```

```
1 # English strings go here for Rails i18n
2 en:
3   # my_label: "My label"
4
```

```
1 version: v1.0
2 name: Initial Pipeline
3 agent:
4   machine:
5     type: e1-standard-2
6     os_image: ubuntu2004
7 blocks:
8   - name: Checks
9     task:
10    jobs:
11      - name: Tests
12        commands:
13          - bin/test
14      - name: Linter
15        commands:
16          - bin/fastcheck
17 prologue:
18   commands:
19     - 'git clone git@github.com:redmine/redmine.git'
20     - cd redmine
21     - cp config/configuration.yml.example config/
configuration.yml
22     - 'git clone git@github.com:aneshodza/redmine-
postgres-database-yml.git'
23     - cp redmine-postgres-database-yml/database.yml
config/database.yml
24     - rm -rf redmine-postgres-database-yml
25     - bin/bundle install
26     - yarn
27     - sem-service start postgres --username="root" --
password=""
28     - 'bin/rails db:create'
29     - 'bin/rails db:migrate'
30     - cd plugins
31     - checkout
32     - cd ../../..
33     - bin/bundle install
34     - cd plugins/gnosis
35     - bin/setup
36
```