GSoC 2020 - Checks API Plugin for Jenkins

Kezhi Xiong

Project Mentors:
Ullrich Hafner
Tim Jacomb

DEVOPS WORLD by CloudBees

About me...

@XiongKezh

kezhi-xiong-44234a188

@AugustX_

I’m a incoming graduate student at Northeastern University (Boston) and currently in gap year…

I joined Jenkins as a Google Summer of Code 2020 Student. I’m an open source enthusiast who enjoys sharing ideas and projects with other developers. Currently, I’m working on Jenkins Checks API Plugin.

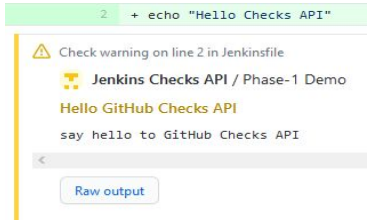
In my spare time, I watch soccer (Chelsea fan) and Motorsports (F1, WRC)...

© 2020 All Rights Reserved.

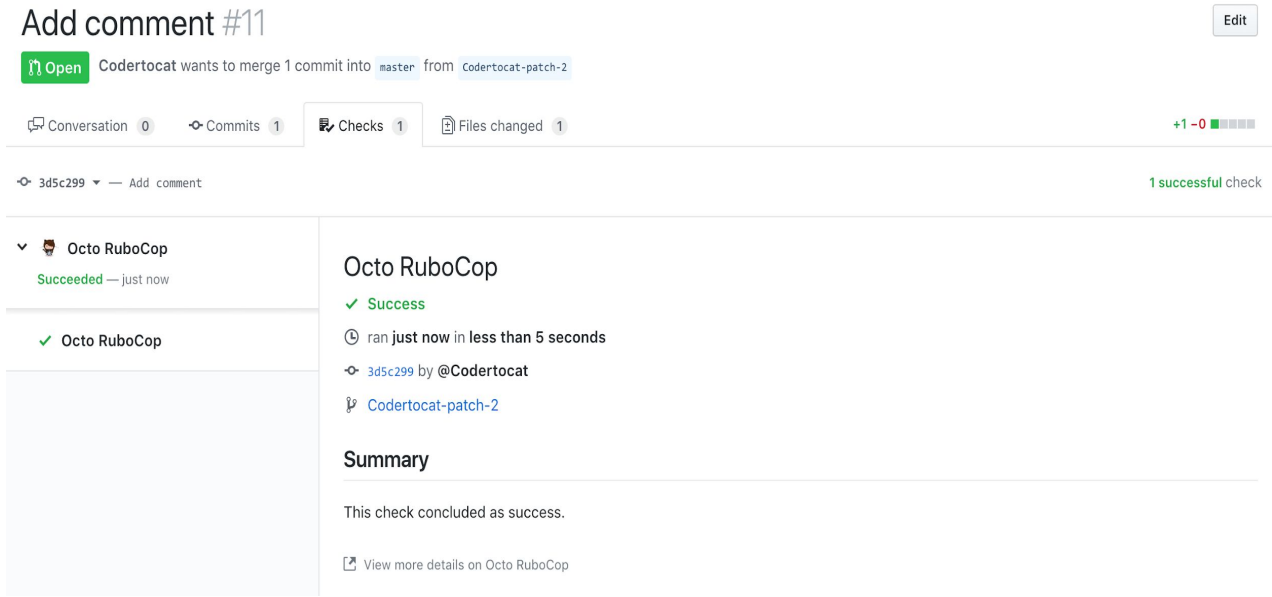
01 **Introduction** of GitHub Checks API

A highly customized way to integrate CI tools or make reports for PR:

- Status
  - Queued
  - In Progress
  - Completed
- Conclusion
  - Success
  - Failed
  - Neutral
  - Canceled
  - Timed Out
- Annotations
  - Make warnings / notices on specific lines of code



- Actions
  - Re-run
  - Automatic Formatting / Fixation



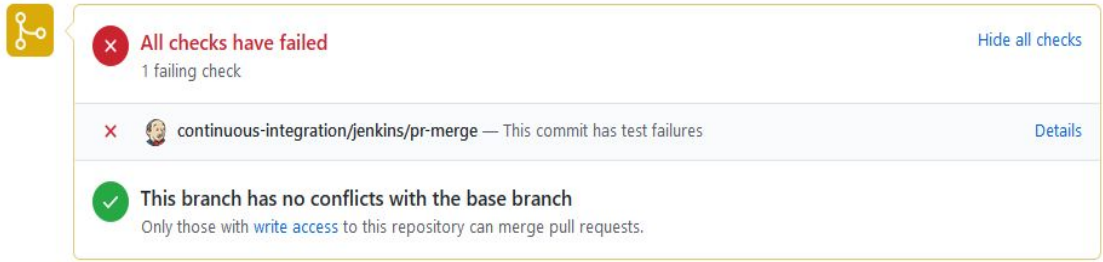
From https://developer.github.com/apps/quickstart-guides/creating-ci-tests-with-the-checks-api

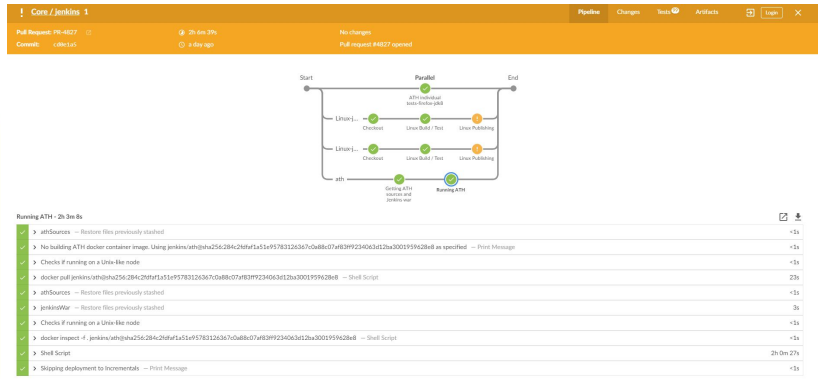
© 2020 All Rights Reserved.

02 Motivation

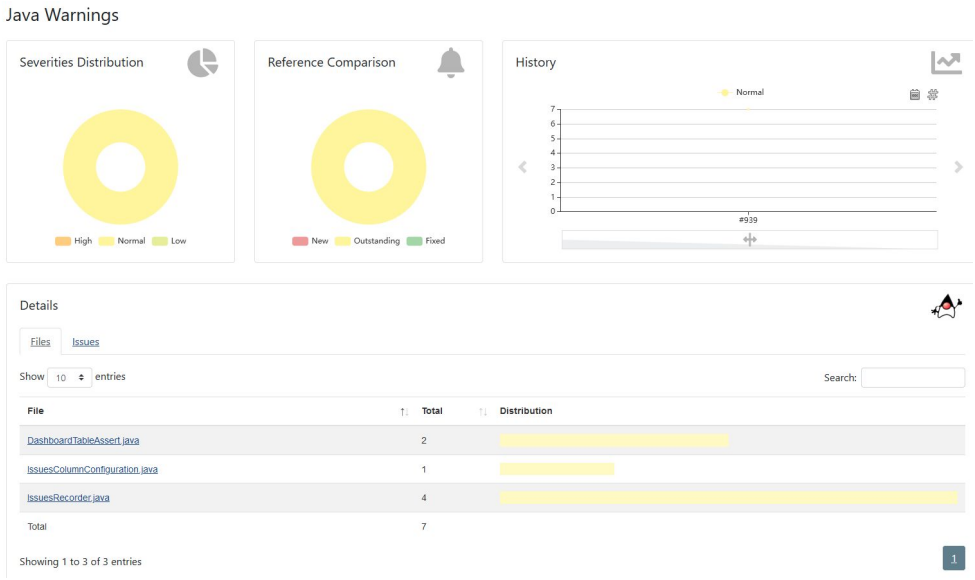
How Jenkins is integrated now -- GitHub Status API

- Only a status and a short message





What Jenkins is capable of now -- Warnings Plugin, Code Coverage API Plugin ...



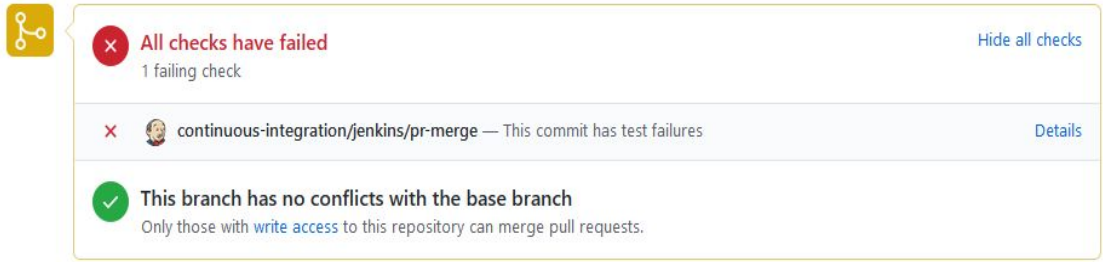
DEVOPSWORLD by CloudBees

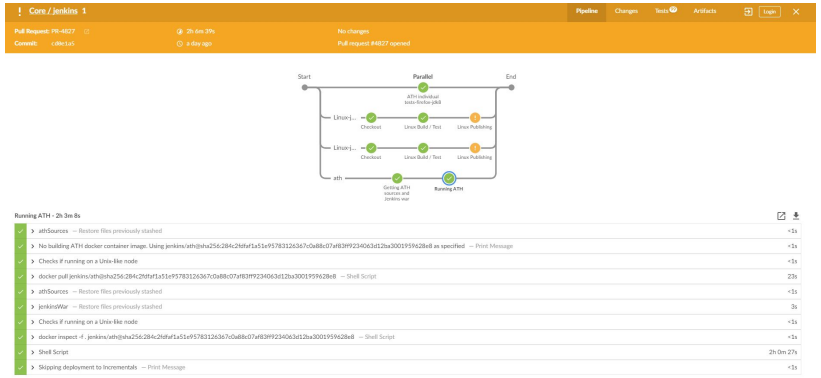
© 2020 All Rights Reserved.

02 Motivation

How Jenkins is integrated now -- GitHub Status API

- Only a status and a short message





What Jenkins is capable of now -- Warnings Plugin, Code Coverage API Plugin ...

GIF from Jeff's slide on GSoC 2018 mentor summit

© 2020 All Rights Reserved.

02 Motivation

How about integrating GitHub with them?

Content of file NullBlamer.java

```
1  package io.jenkins.plugins.analysis.core.scm;
2
3  import edu.hm.hafner.analysis.Report;
4
5  /**
6   * A blamer that does nothing.
7   *
8   * @author Ullrich Hafner
9   */
10 public class NullBlamer implements Blamer {
11     @Override
12     public Blames blame(final Report report) {
13         report.logInfo("Skipping blaming as requested in the job configuration");
14         return new Blames();
```

io.jenkins.plugins.analysis.core.scm.NullBlamer is Serializable; consider declaring a serialVersionUID

This class implements the `Serializable` interface, but does not define a `serialVersionUID` field. A change as simple as adding a reference to a .class object will add synthetic fields to the class, which will unfortunately change the implicit serialVersionUID (e.g., adding a reference to `String.class` will generate a static field `class$java$lang$String`). Also, different source code to bytecode compilers may use different naming conventions for synthetic variables generated for references to class objects or inner classes. To ensure interoperability of Serializable across versions, consider adding an explicit serialVersionUID.

```
15     }
16 }
```

Source Code View from Warnings Plugin

Add comment #12

Edit

Open Codertocat wants to merge 1 commit into master from Codertocat-patch-2

Conversation 0 · Commits 1 · Checks 1 · Files changed 1

Changes from all commits · Jump to... · +1 −0 · Diff settings · Review changes

1 octocat.rb — Copy path · View file

```
@@ -13,5 +13,6 @@ def display
13  13    end
14  14  end
15  15
16      + # Instantiate a new Octocat
16  17    m = Octocat.new("Mona", "cat", "octopus")
```

Check notice on line 17 in octocat.rb
Octo RuboCop / Octo RuboCop
octocat.rb#L17
Style/StringLiterals: Prefer single-quoted strings when you don't need string interpolation or special symbols.

Check notice on line 17 in octocat.rb
Octo RuboCop / Octo RuboCop
octocat.rb#L17
Style/StringLiterals: Prefer single-quoted strings when you don't need string interpolation or special symbols.

Check notice on line 17 in octocat.rb
Octo RuboCop / Octo RuboCop
octocat.rb#L17
Style/StringLiterals: Prefer single-quoted strings when you don't need string interpolation or special symbols.

```
17  18  m.display
```

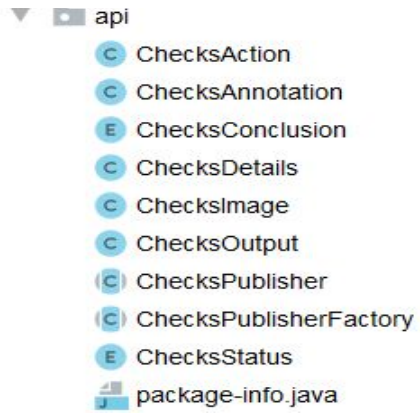
GitHub Check Run Annotations

© 2020 All Rights Reserved.

03 Plugins

A General API

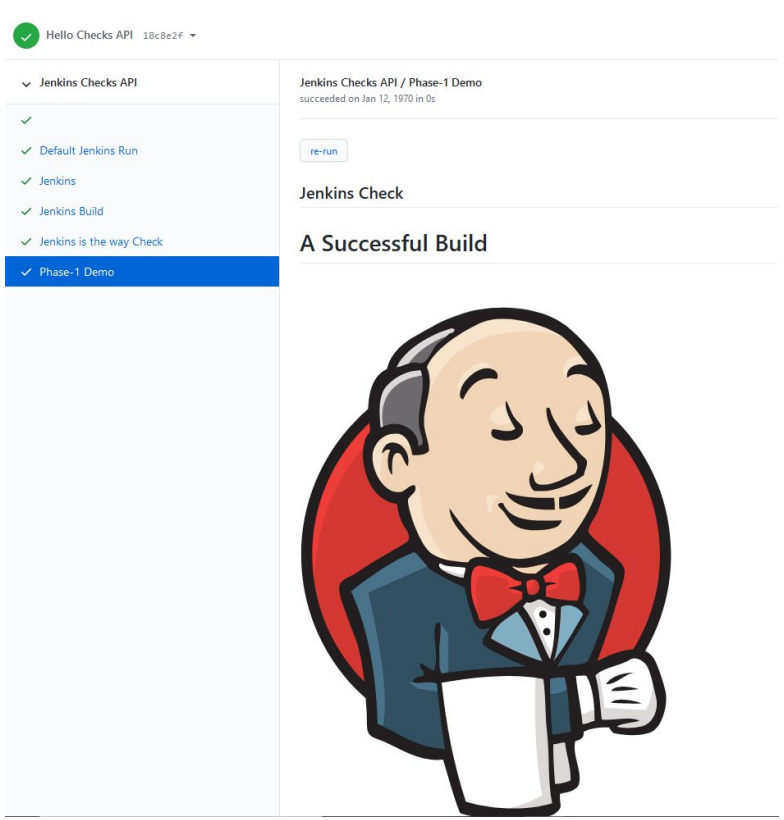
jenkinsci / checks-api-plugin



Prepared for the similar concept in different platforms like GitLab, BitBucket, etc.

An Implementation for GitHub Checks API

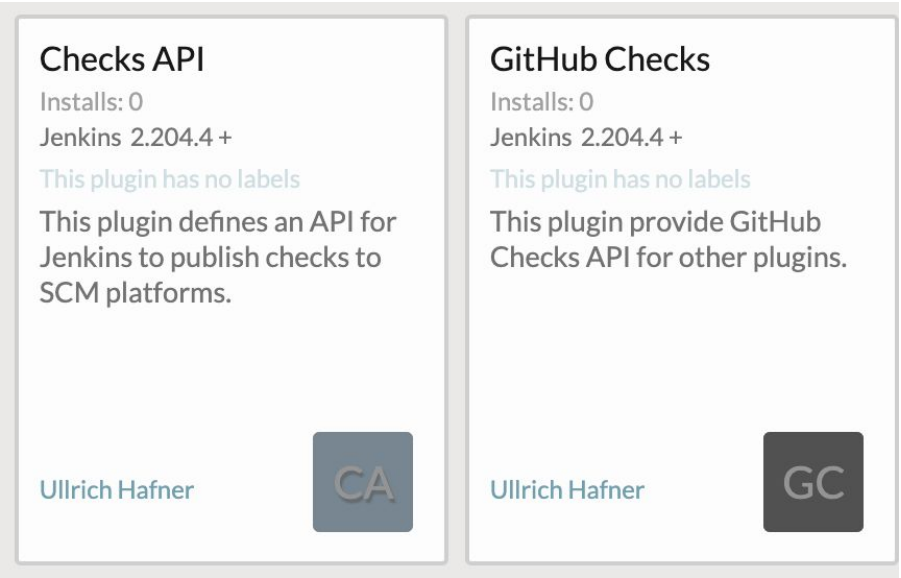
jenkinsci / github-checks-plugin

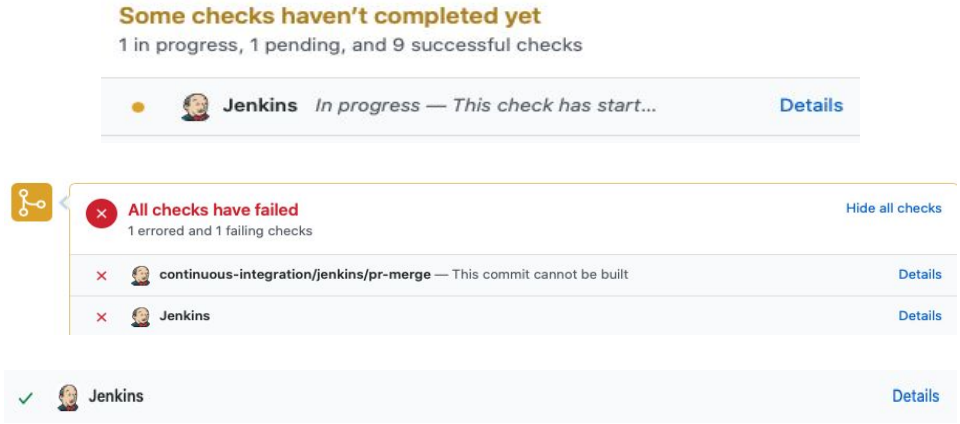


© 2020 All Rights Reserved.

03 Plugins

🎉Released Beta





link

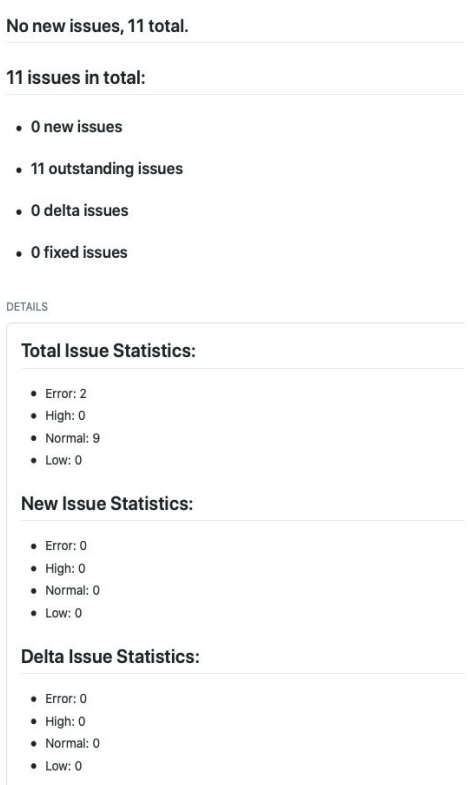
- Install both plugins on your Jenkins
- Use GitHub App credentials for your multi-branch / github org project

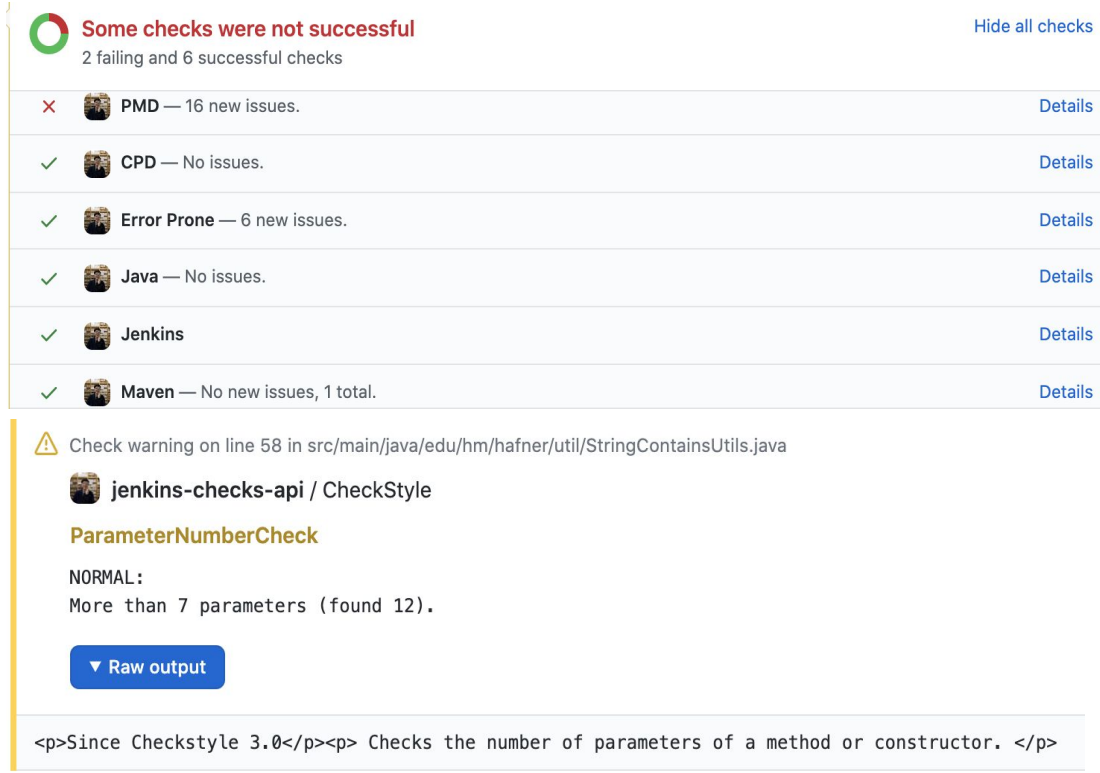
© 2020 All Rights Reserved.

04 Warning Checks

Powered by
Warnings Next Generation Plugin

- Quality Gate
- Issue Statistics
- Severity Statistics
- Annotations for new issues
- ...





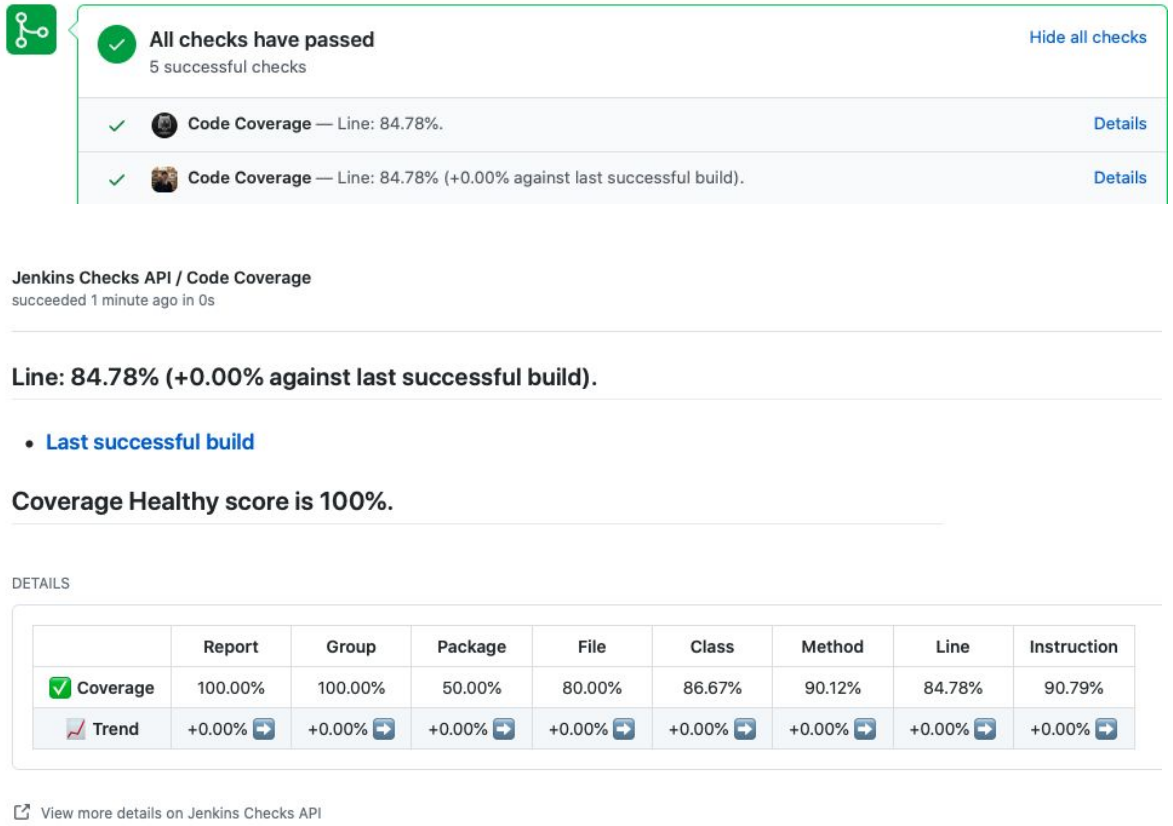
We are helping hundreds of Jenkins Plugins' daily development on ci.jenkins.io 🎉

© 2020 All Rights Reserved.

05 Coverage Checks

Powered by
Code Coverage API Plugin

- Coverage Trend
- Coverage Healthy Score
- Different Coverages
- ...

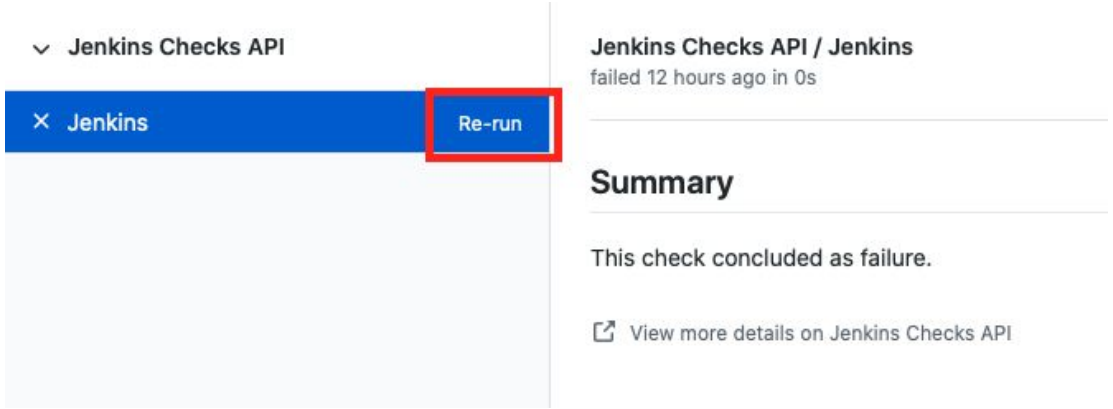


Link

© 2020 All Rights Reserved.

06 **Features** on the way

- Pipeline support
- Request rerun Jenkins build on GitHub
- ...



We hope the next great feature is ...

Powered by
YOUR Plugin!

© 2020 All Rights Reserved.

Thanks!

Gitter channel: https://gitter.im/jenkinsci/github-checks-api