



黑客派

# Github 装逼指南——Travis CI 和 Codecov

作者: [honglan](#)

原文链接: <https://hacpai.com/article/1512349992886>

来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>[原文]<a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fsegmentfault.com%2Fa%2F1190000004415437" target="\_blank" rel="nofollow ugc">https://segmentfault.com/a/1190000004415437</a></p>  
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>  
<!-- 黑客派PC帖子内嵌-展示 -->  
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>  
<script>  
 (adsbygoogle = window.adsbygoogle || []).push({});  
</script>  
<p>好久没写博客了，趁着年前空闲的时间来一篇轻松点的东西。<br>最近工作中积累了一些 Angular 组件打算整一整把他们开源了，既然要开源那么代码可靠性就很重要了，单测不能少，为了保证每提交的代码都能很好的运行，持续集成不能少。之前看到很多开源项目介绍中都有的一些单测覆盖率和 uild 结果的图标，就像这样：</p>  
<p></p>  
<p>觉得挺酷的。打算在自己的开源组件中也整一套。<br>经过 Google 决定使用 TravisCI 来进行持续集成，Codecov 来统计单测覆盖率。</p>  
<h2 id="Travis-CI">Travis CI</h2>  
<p>Travis CI 是国外新兴的开源持续集成构建项目，支持 GitHub 项目。使用十分方便。</p>  
<ol>  
 <li><p>使用 GitHub 账号登录 <a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Ftravis-ci.org%2F" target="\_blank" rel="nofollow ugc">Travis CI</a>; </p></li>  
 <li><p>登录之后会自动同步 GitHub 项目，选择需要使用 Travis CI 的项目</p></li>  
 <li><p>在项目的根目录新增 <code>.travis.yml</code> 文件，内容如下：</p></li>  
</ol>  
<pre><code class="highlight-chroma">#指定运行环境  
language: node\_js  
#指定nodejs版本，可以指定多个  
node\_js:  
 - 0.12.5

#运行的脚本命令

script:

- npm run ci

#指定分支，只有指定的分支提交时才会运行脚本

branches:

only:

- master

</code></pre>

<p>更多语法请看<a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fdocs.travis-ci.com%2F" target="\_blank" rel="nofollow ugc">这里</a>。使用起来非常方便，这样当你每向 GitHub push 代码的时候，Travis CI 就会自动运行 <code>.travis.yml</code> 里面的 <code>script</code>。自动进行编译以及运行单测。<br>由于 Travis CI 每次 build 之前都会运行 <code>npm install</code> 安装项目依赖的 npm 包，所以在提交代码的时候要保证把所有依赖的包都已在 <code>package.json</code> 中声明了，否则 build 就会失败。</p>

<h2 id="Codecov">Codecov</h2>

<p>Codecov 是一个开源的测试结果展示平台，将测试结果可视化。GitHub 上许多开源项目都使用 Codecov 来展示单测结果。<br>Codecov 跟 Travis CI 一样都支持 GitHub 账号登录，同样会同步 GitHub 中的项目。在 nodejs 环境下使用 Codecov 需要安装对于的 npm 包，运行下面这个命令进安装：</p>

```
<pre><code class="highlight-chroma">npm install codecov --save-dev
</code></pre>
```

<p>这个包的作用是将我们运行单测产生的结果文件上传到 Codecov 上进行可视化展示。同时 code ov 支持的结果文件类型为 <code>cobertura</code>。所以需要保证单测执行的结果文件的类型为 <code>cobertura</code>。<br>前端项目进行单元测试推进 <code>karma</code> + 'jasmine' 的组合。这两个具体是什么东西大家 Google 一下就知道。使用 <code>karma</code> 可以通过 单的配置来运行单测。下面是我一个项目中的配置文件，供大家参考：</p>

```
<pre><code class="highlight-chroma">// Karma configuration
// Generated on Mon Feb 01 2016 21:34:22 GMT+0800 (中国标准时间)
```

```
module.exports = function(config) {
  config.set({
```

```
  })
```

```
}
```

```
</code></pre>
```

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
```

```
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
```

<p>通过 karma 进行单元测试，将命令写到 <code>.travis.yml</code> 中就可以在每次 build 的 候运行单测，同时运行 <code>codecov [cobertura-coverage.xml路径]</code> 就会把单测结果 传到 Codecov。在本地运行 codecov 会失败，需要将这个过程加入到 Travis CI 的 build 脚本中， 能成功上传。因为在本地运行就会被作为私有项目，对于私有项目在上传结果时需要加上 Codecov 供的 token。</p>

<h2 id="在github中加入图标">在 GitHub 中加入图标</h2>

<p>到了最后一步，Travis CI 和 Codecov 都提供图标链接来展示结果。我们只需要将图标链接加入 项目的 README 中就可以看到结果了。<br>对于 Travis CI 来说，点击下图中的图标：</p>

<p><br>就会弹出图标的地址。<br>对于 Codecov 来说，打开项 的设置列表就会看到，如下：</p>

<p></p>

<p>最后只需要将对应的链接加到 README 文件中就可以了。下面是最后的效果：</p>

<p></p>

<p>是不是很赞！<br>项目地址：<a href="https://link.hacpai.com/forward?goto=https%3A%

[F%2Fgithub.com%2FGaojingComponent%2FBGSingleSelector](https://github.com/GaojingComponent/BGSingleSelector) target="\_blank" rel="nofollow ugc">BGSingleSelector</a>，欢迎大家试用提意见，同时不要吝啬 Star。</p>