

# Jenkins

[jenkins-ci.org](https://jenkins-ci.org)

# CI Server

Continuous builds  
or tests...

It can poll your source code repository  
Or you can setup hooks

# Testing Perl

You could use it to build your release packages  
I mostly just use it for testing.

Install `TAP::Formatter::JUnit` for the simplest integration. There is now a TAP plugin for Jenkins but I haven't tried it.

I also use `local::lib` to allow my tests to install dependencies.

# A Jenkins Job

## Example Job

**Poll SCM** H/5 \* \* \* \* # H makes it random

### Execute Shell commands

```
eval `/opt/perl5/bin/perl -Mlocal::lib=~/.perl5`
```

```
PERL5LIB=../../Jenkins-Config/workspace/lib:../../Jenkins-API/workspace/lib /opt/perl5/bin/cpanm --installdeps . -l ~/.perl5
```

```
/opt/perl5/bin/prove -I ../../Jenkins-Config/workspace/lib -I ../../Jenkins-API/workspace/lib -I
```

```
/var/lib/jenkins/perl5/lib/perl5/i686-linux -I
```

```
/var/lib/jenkins/perl5/lib/perl5 --timer --formatter=TAP::
```

```
Formatter::JUnit -l t > ${JOB_NAME}-${BUILD_NUMBER}-junit.xml
```

### Publish JUnit test result summary

```
*junit.xml
```

# Poking Jenkins

Jenkins is very accessible programmatically. I haven't found a place I can't stick `/api` on the end of a url and get a help page on the API.

Jenkins::API is a simple module for talking to Jenkins.

# Jenkins::API

```
use Jenkins::API;

my $jenkins = Jenkins::API->new({ base_url => 'http://jenkins:8080' });
my $status = $jenkins->current_status();
my @not_succeeded = grep { $_->{color} ne 'blue' } @{$status->{jobs}};
# {
#   'color' => 'red',
#   'name' => 'Test-Project',
#   'url' => 'http://jenkins:8080/job/Test-Project/',
# }

my $success = $jenkins->create_job($project_name, $config_xml);
...
```

# To Come

## Jenkins::Config

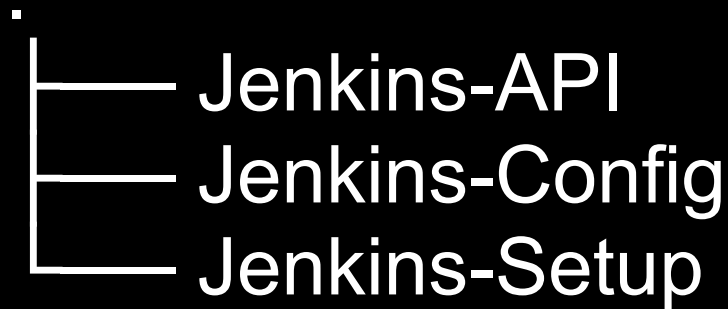
Designed to help with building the job configuration file. Currently uses XML::Compile which produces XML that works. It really needs some work to make it generally useful.

There is no overall xsd for the config file because it's clauses depend on the plugins you have installed. I used trang to create mine.

# Jenkins::Setup

Setup your modules on Jenkins as a Job.  
Relies on conventions to add local dependencies as dependencies in Jenkins

Reads META.yml for project info





# Code?

Jenkins::API is on CPAN

Jenkins::Config and Jenkins::Setup are on github.

<https://github.com/colinnewell>