Minion Queue

```perl
#!/usr/bin/env perl
use Mojolicious::Lite;

plugin Minion => { SQLite => 'sqlite:jobs.db' };

post '/jobs' => sub {
    my $c   = shift;

    my $id = $c->app->minion->enqueue(restart_server => [$c->req->json]);

    $c->render(json => { message => "accepted", id => $id }, status => 202);
};

app->start;
```

```
plugin Minion => { SQLite => 'sqlite:jobs.db' };
```

```
    my $id = $c->app->minion->enqueue(restart_server => [$c->req->json]);
```

Minion Queue

```perl
#!/usr/bin/env perl
use Mojolicious::Lite;

plugin Minion => { SQLite => 'sqlite:jobs.db' };

post '/jobs' => sub {
    my $c   = shift;

    my $id = $c->app->minion->enqueue(restart_server => [$c->req->json]);

    $c->render(json => { message => "accepted", id => $id }, status => 202);
};

app->start;
```

Minion Worker

```perl
#!/usr/bin/env perl
use Mojolicious::Lite;

plugin Minion => { SQLite => 'sqlite:jobs.db' };

app->minion->add_task(
    restart_server => sub {
        my ($job, $server) = @_;

        $job->on(finished => sub {
            my ($job, $result) = @_;
            my $id = $job->id;
            $job->app->log->info("Job ID $id finished.");
        });

        my $res = app->ua->get("http://localhost:3001/" . $server->{wait})->res;
        $job->finish($res->json);
    }
);

app->start;
```