



Mojolicious::Validator

```
use Mojolicious::Lite;
```

```
app->validator->add_check(  
    range => sub {  
        my ($validation, $name, $value, $min, $max) = @_;  
        return $value < $min || $value > $max;  
    }  
);
```

```
post '/signup' => sub {  
    my $c      = shift;  
    my $validator = $c->validation;  
  
    $validator->required('name')->size(1, 10)->like(qr/^[a-z]/);  
    $validator->required('state')->in(qw/UT TX CA OR ID/);  
    $validator->required('age')->range(18, 40);  
  
    return $c->render('400') if $validator->has_error;  
  
    $c->render(text => "Welcome " . $validator->param('name'));  
};  
  
app->start;
```

```
app->validator->add_check(  
    range => sub {  
        my ($validation, $name, $value, $min, $max) = @_  
        return $value < $min || $value > $max;  
    }  
);
```

```
$validator->required('age')->range(18, 40);
```

```
$validator->required('name')->size(1,10)->like(qr/^[a-z]/);
```

```
$validator->required('state')->in(qw/UTX CA OR ID/);
```

```
return $c->render('400') if $validator->has_error;
```



```
$c->render(text=>'Welcome', $validator->param('name'));
```

# Mojolicious::Validator

```
use Mojolicious::Lite;

app->validator->add_check(
    range => sub {
        my ($validation, $name, $value, $min, $max) = @_;
        return $value < $min || $value > $max;
    }
);

post '/signup' => sub {
    my $c      = shift;
    my $validator = $c->validation;

    $validator->required('name')->size(1, 10)->like(qr/^[a-z]/);
    $validator->required('state')->in(qw/UT TX CA OR ID/);
    $validator->required('age')->range(18, 40);

    return $c->render('400') if $validator->has_error;

    $c->render(text => "Welcome " . $validator->param('name'));
};

app->start;
```

# Mojo::Server::Daemon

```
use Mojo::Server::Daemon;

my $handler = sub {
    my ($d, $tx) = @_;

    $tx->res->code(200);
    $tx->res->headers->content_type($tx->req->headers->content_type);
    $tx->res->body($tx->req->body);
    $tx->resume;
};

my $daemon = Mojo::Server::Daemon->new(listen => ["http://*:8080"]);
$daemon->on(request => $handler);
$daemon->start;

Mojo::IOLoop->start;
```