

## NAME

`Module::Pluggable::Object` - automatically give your module the ability to have plugins

## SYNOPSIS

Simple use `Module::Pluggable` -

```
package MyClass;
use Module::Pluggable::Object;

my $finder = Module::Pluggable::Object->new(%opts);
print "My plugins are: ".join(", ", $finder->plugins)."\n";
```

## DESCRIPTION

Provides a simple but, hopefully, extensible way of having 'plugins' for your module. Obviously this isn't going to be the be all and end all of solutions but it works for me.

Essentially all it does is export a method into your namespace that looks through a search path for `.pm` files and turn those into class names.

Optionally it instantiates those classes for you.

This object is wrapped by `Module::Pluggable`. If you want to do something odd or add non-general special features you're probably best to wrap this and produce your own subclass.

## OPTIONS

See the `Module::Pluggable` docs.

## AUTHOR

Simon Wistow <simon@thegestalt.org>

## COPYING

Copyright, 2006 Simon Wistow

Distributed under the same terms as Perl itself.

## BUGS

None known.

## SEE ALSO

*Module::Pluggable*