

```
Dir["#{PLUGIN_DIR}/*.rb"].each{ |x| load x }
```

# Registering classes

`self.inherited`

```
class PluginBase
  @registered_plugins = []

  def self.inherited(child)
    PluginBase.registered_plugins << child
  end

  class << self
    attr_reader :registered_plugins
  end
end
```

```
# This would go in PLUGIN_DIR/whatever.rb
```

```
class MyPlugin < PluginBase
  def foo
    1
  end
end
```

```
# See the loaded plugin
```

```
PluginBase.registered_plugins # => [MyPlugin]
```

```
# Plugins subclasses need to be instantiated to be useful
```

```
PluginBase.registered_plugins.map { |klass| klass.new }
```

# Capturing instances

```
class Plugin
  @registered_plugins = {}
  class << self
    attr_reader :registered_plugins
    private :new
  end

  def self.define(name, &block)
    plugin = new
    plugin.instance_eval(&block)
    Plugin.registered_plugins[name] = plugin
  end
end
```

```
Plugin.define "foo" do
  def do_it(x) # This becomes a singleton method
    x * 2
  end
end
```

```
Plugin.registered_plugins.keys # => ["foo"]
plugin = Plugin.registered_plugins["foo"] # => #<Plugin:0xb7de9934>
plugin.do_it "bar " # => "bar bar "
```

Uses

# Campfire Bot

# Issues

- Keeping internal data
- Serialising internal data
- Sandboxing plugins

# Links

- <http://eigenclass.org/hiki.rb?ruby+plugins>
- <http://github.com/timriley/campfire-bot>