

NAME

Padre::Task::SyntaxChecker – Generic syntax–checking background processing task

SYNOPSIS

```
package Padre::Task::SyntaxChecker::MyLanguage;
use base 'Padre::Task::SyntaxChecker';

sub run {
    my $self = shift;
    my $doc_text = $self->{text};
    # black magic here
    $self->{syntax_check} = ...;
    return 1;
};

1;

# elsewhere:

# by default, the text of the current document
# will be fetched as will the document's notebook page.
my $task = Padre::Task::SyntaxChecker::MyLanguage->new();
$task->schedule;

my $task2 = Padre::Task::SyntaxChecker::MyLanguage->new(
    text => Padre::Current->document->text_get,
    editor => Padre::Current->editor,
);
$task2->schedule;
```

DESCRIPTION

This is a base class for all tasks that need to do expensive syntax checking in a background task.

You can either let Padre::Task::SyntaxChecker fetch the Perl code for parsing from the current document or specify it as the "text" parameter to the constructor.

To create a syntax checker for a given document type Foo, you create a subclass Padre::Task::SyntaxChecker::Foo and implement the `run` method which uses the `$self->{text}` attribute of the task object for its nefarious syntax checking purposes and then stores the result in the `$self->{syntax_check}` attribute of the object. The result should be a data structure of the form defined in the documentation of the Padre::Document::check_syntax method. See Padre::Document.

This base class implements all logic necessary to update the GUI with the syntax check results in a `finish()` hook. If you want to implement your own `finish()`, make sure to call `$self->SUPER::finish` for this reason.

SEE ALSO

This class inherits from Padre::Task and its instances can be scheduled using Padre::TaskManager.

The transfer of the objects to and from the worker threads is implemented with Storable.

AUTHOR

Steffen Mueller smueller AT cpan DOT org

COPYRIGHT AND LICENSE

Copyright 2008–2010 The Padre development team as listed in Padre.pm.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl 5 itself.

