NAME

CPANPLUS::Internals::Fetch

SYNOPSIS

```
    my $output = $cb->_fetch(
                        module      => $modobj,
                        fetchdir    => '/path/to/save/to',
                        verbose     => BOOL,
                        force       => BOOL,
                    );

    $cb->_add_fail_host( host => 'foo.com' );
    $cb->_host_ok(       host => 'foo.com' );
```

DESCRIPTION

CPANPLUS::Internals::Fetch fetches files from either ftp, http, file or rsync mirrors.

This is the rough flow:

```
    $cb->_fetch
        Delegate to File::Fetch;
```

METHODS

$path = _fetch(module => $modobj, [fetchdir => '/path/to/save/to', fetch_from => 'scheme://path/to/fetch/from', verbose => BOOL, force => BOOL, prefer_bin => BOOL])

`_fetch` will fetch files based on the information in a module object. You always need a module object. If you want a fake module object for a one-off fetch, look at `CPANPLUS::Module::Fake`.

`fetchdir` is the place to save the file to. Usually this information comes from your configuration, but you can override it expressly if needed.

`fetch_from` lets you specify an URI to get this file from. If you do not specify one, your list of configured hosts will be probed to download the file from.

`force` forces a new download, even if the file already exists.

`verbose` simply indicates whether or not to print extra messages.

`prefer_bin` indicates whether you prefer the use of commandline programs over perl modules. Defaults to your corresponding config setting.

`_fetch` figures out, based on the host list, what scheme to use and from there, delegates to `File::Fetch` do the actual fetching.

Returns the path of the output file on success, false on failure.

Note that you can set a `blacklist` on certain methods in the config. Simply add the identifying name of the method (ie, `lwp`) to: $conf->_set_fetch(blacklist => ['lwp']);

And the `LWP` function will be skipped by `File::Fetch`.

_add_fail_host(host => $host_hashref)

Mark a particular host as bad. This makes `CPANPLUS::Internals::Fetch` skip it in fetches until this cache is flushed.

_host_ok(host => $host_hashref)

Query the cache to see if this host is ok, or if it has been flagged as bad.

Returns true if the host is ok, false otherwise.