Proposal: autoload_relative

In my experience, autoloads often reflect an existing hierarchical structure. If a project does not use Zeitwerk, and the user declares autoloads for a class or module, chances are they are for child constants. As an example, see the ActiveRecord module. (Those ones do not have a second argument because we define wrapper that derives it by convention, here).

I think it would be convenient to have an autoload_relative in the line of Kernel#require_relative. It would make existing patterns more concise, and as a practical consequence, you skip $LOAD_PATH lookups too.

Related issues:
Related to Ruby master - Feature #15330: autoload_relative

History
#1 - 06/19/2022 08:04 AM - ioquatix (Samuel Williams)
Yes, great idea, and I can confirm I've run into issues with autoload loading the wrong files if you combine autoload with require_relative.

Also, since this does not exist yet, we can be more strict on how this works, as in autoload_relative can raise an exception if the constant is not defined correctly.

#2 - 06/19/2022 02:19 PM - marcandre (Marc-Andre Lafortune)
Note that this is a duplicate of https://bugs.ruby-lang.org/issues/15330, which includes some real world usage stats.

Making it strict seems like a good idea too.

#3 - 06/19/2022 04:31 PM - fxn (Xavier Noria)
Oh, I should have searched before! Can't close this one myself.

#4 - 06/19/2022 11:22 PM - ioquatix (Samuel Williams)
Great minds think alike! Thanks @marcandre (Marc-Andre Lafortune).

@fxn (Xavier Noria) I think the next step is to make a PR and then get Matz to review it.

#5 - 06/20/2022 07:28 AM - byroot (Jean Boussier)
- Related to Feature #15330: autoload_relative added