Tapestry5HowToSendErrorPageAsEmail

Like the extensive and verbose error output of tapestry development mode? Don't want to scare your users away with the page? Want to be notified when your users run into trouble?

This recipe is adapted from the official documentation (http://tapestry.apache.org/overriding-exception-reporting.html)

App Module

integrate this into your app module:

```
     public static void contributeApplicationDefaults(
            MappedConfiguration<String, String> configuration)
    {

        configuration.add(USER_FRIENDLY_EXCEPTION_PAGE, "exception/Errors");
        configuration.add(FULL_EXCEPTION_REPORT_PAGE, "exception/ExceptionReport");
    }

    public RequestExceptionHandler buildAppRequestExceptionHandler(
            @Inject final ResponseRenderer responseRender,
            @Inject final PageDocumentGenerator generator,
            @Inject final ComponentSource componentSource,
            final @Symbol(AppModule.EMAILSERVER) String emailserver,
            final @Symbol(SymbolConstants.PRODUCTION_MODE) boolean productionMode,
            final @Symbol(AppModule.FULL_EXCEPTION_REPORT_PAGE) String exceptionPage,
            final @Symbol(AppModule.USER_FRIENDLY_EXCEPTION_PAGE) String userfriendly,
            final @Inject Response response)
    {

        final String sender = // your sender, symbol etc
        final String rcpt = // recepient of the email

        return new RequestExceptionHandler()
        {
            @Override
            public void handleRequestException(Throwable exception) throws IOException
            {
                // always log to console
                logger.error("Unexpected runtime exception: " + exception.getMessage(), exception);
                // let clients know something is up
                response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);

                // get the page that will render the full non-production mode output
                Component exceptionReport = componentSource.getPage(exceptionPage);
                ((ExceptionReporter) exceptionReport).reportException(exception);

                if (productionMode) {
                    try {

                        String errorTrackId = System.currentTimeMillis() + "";

                        /** render the full exception page **/
                        Document devOutput = generator.render(exceptionPage);
                        /** send email here **/

                        /** this is a custom email builder so you'll have to substitute your own email sending
code
                        Email.from(sender).to(rcpt).withSubject("Error occured [tracking id=%s]", errorTrackId)
                                .withContent("text/html", devOutput.toString()).finish().sendVia(emailserver);
                        **/


                        /** Show your custom, user friendly page **/
                        /**
                        * In the original scenario I would show the tracker-id on the user-friendly error
page
                        * Component userfiendly = componentSource.getPage(userfriendly);
                        *((ErrorTracker) userfiendly).setErrorID(errorTrackId);
```

```
                    **/

                    responseRender.renderPageMarkupResponse(userfriendly);

                } catch (Throwable t) {
                    logger.error("Error in error handling. Ironic? Yes!", t);
                }
            } else {
                /** if in non-production we render the full exceptionPage right away **/
                responseRender.renderPageMarkupResponse(exceptionPage);
            }
        }
    };
}

public void contributeServiceOverride(MappedConfiguration<Class, Object>
        configuration,
        @Local RequestExceptionHandler handler) {

    configuration.add(RequestExceptionHandler.class, handler);

}
```

Exception Report

Unfortunately ExceptionReport (the class that renders the exception pages per default) reads the production-mode symbol itself so we have to copy the class. You can find it in the tapestry-core.jar. There it's class org.apache.tapestry5.corelib.pages.ExceptionReport.

Copy class and tml to your project and modify to always show the extensive report. I did that by stripping out the productionMode property and the <t:if><p:else></t:if> referencing it in the template.

The contributeApplicationDefaults above assumes you have copied it to exception/ExceptionReport in your pages folder.

Error Page

The contributeApplicationDefaults above assumes you have created a page that display user-friendly errors as exception/Errors in your pages folder.