ErrorReporting SPI for Slack

Table of Contents

This module (`incode-module-errorrptslack-impl`) provides an implementation of Apache Isis' `ErrorReportingService` SPI. Whenever an error page is displayed to the user, the implementation will generate a unique id (a guid) and post a message to a specified Slack channel (to notify production support). It will also display a random image.

API & Implementation

This module provides `ErrorReportingServiceForSlack` domain service as an implementation of the `ErrorReportingService`:

```
public interface ErrorReportingService {
    Ticket reportError(final ErrorDetails errorDetails);
}
```

where the returned `Ticket` determines what is shown on the error page for the end-user. In the case of this implementation:

```
public class Ticket implements Serializable {
    private final String reference;                 ①
    private final String userMessage;               ②
    private final String details;                   ③
    private final StackTracePolicy stackTracePolicy;  ④
    private final String kittenUrl;                 ⑤
    ...
}
```

① the random UUID. This is posted to the Slack also, so if the user quotes this number then the support team can find additional details (such as the stack trace)

② the main error message to display to the user. This will default to:

"Our apologies, an error has occurred. The development team has been notified (via the Slack messaging system).

but can be configured to something else if required (see below).

③ optionally, additional details (shown in a smaller font). This defaults to:

"Quote the reference below to point us to more detailed information about the problem. | In the meantime, and by way of an apology, here's a picture of a kitten for you to look at.

but can be configured to something else if required (see below).

④ whether the end user should be able to view the stack trace. This implementation ALWAYS HIDES the stack trace.

⑤ the URL of an image to display, if any. This will default to:

http://www.randomkittengenerator.com/cats/rotator.php

but can be configured to something else if required (see below).

How to configure/use

Classpath

Update your classpath by adding this dependency in your `dom` project's `pom.xml`:

```
<dependency>
    <groupId>org.incode.module.errorrptslack</groupId>
    <artifactId>incode-module-errorrptslack-impl</artifactId>
</dependency>
```

Check for later releases by searching Maven Central Repo.

Configuration Properties

The following configuration properties are required (either in `isis.properties` or through the `AppManifest`):

```
isis.service.slack.authToken=XXXXXX                    ①
isis.service.errorReporting.slack.channel=ProdSupport
```

① there is no "errorReporting" part to this config property; it is to configure the underlying Slack Module.

Optionally the message text and details can be changed:

```
isis.service.errorReporting.slack.userMessage=...
isis.service.errorReporting.slack.details=...
isis.service.errorReporting.slack.kittenUrl=...
```

The timeout for waiting for Slack to acknowledge the message can also be tweaked:

```
isis.service.errorReporting.slack.timeoutMs=500
```

Bootstrapping

In the `AppManifest`, update its `getDependencies()` method, eg:

```
@Override
public Set<Module> getDependencies() {
    return Sets.newHashSet(
            ...
            new org.incode.module.errorrptjira.ErrorReportingSlackModule(),
            ...
    );
}
```

Known issues

None known at this time.

Dependencies

Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/spi/errorrptslack/impl -D excludeTransitive=true
```

which, excluding Apache Isis itself, depends only on the Slack module.