Handling errors - Java

It is always possible for your request to return an error. If an error occurs, e.g. if your API username/password is incorrect, our wrapper will throw a Java exception. If you're not familiar with handling exceptions in Java, it basically means you need to put the wrapper function calls in a try-catch block as shown in the code example from here.

Example

```
try {
 Hashtable<String, String> result;
 result = tmClient.sendSMS("My message", "447000000000", "Sender", 72, "", "",
null);
} catch(RestClientException e) {
 Hashtable<String, String> errors = tmClient.getLastErrors();
}
```

You must decide how to handle errors according to the logic of your particular application.
For example, if you're building a web interface to send text messages, and a send fails, you may wish to show a visual alert to the user.
Note that more than one error can be reported at once. Therefore although a single Java exception (the first of the errors) is thrown, the complete set of errors encountered is available in an Hashtable of key-value pairs. The key of each pair is the error code and the value is a detailed description of the error. This is why the example displays the errors using the following code:

```
for(Map.Entry<String, String> error: errors.entrySet())
 System.out.println("Error code " + error.getKey() + ": " + error.getValue());
```

- Next: General error codes
- Home: Developers Documentation

You must know!
Our example code is an illustration of how you might integrate with our systems and is not certified for production environments. You are responsible for testing and QA.