

Exceptions

The exceptions of the XML library, when they are thrown, the details that they have and how to translate their messages

Anonymous · 17/10/2026

Exceptions

All the exceptions of the library extend `Derafu\Xml\Exception\XmlException`, so one `catch` takes all of them.

The Exceptions

Exception	When it is thrown
<code>XmlException</code>	A document without document element where it is needed, a validation that fails, a schema that is not found, and failures of PHP while the XML is generated (serializing, canonicalizing).
<code>XmlParseException</code>	<code>loadXml()</code> with an XML that is empty or is not well formed.
<code>XmlQueryException</code>	An XPath expression that is not valid or that fails, and <code>C14NEncoded()</code> with an XPath that finds no node.
<code>XmlEncoderException</code>	An array that can not be converted to XML.

`XmlParseException`, `XmlQueryException` and `XmlEncoderException` extend `XmlException`.

Array that Can Not Be Converted

`XmlEncoder::encode()` throws `XmlEncoderException` when:

- The array has more than one element at its first level, or a list: an XML has only one root element.
- A name of an element or attribute is not text (a number is the position of a value in a list, not a name).
- A value is not text: a string, a number, a boolean or an object that can be a string are valid. An array can not be the value of an attribute.
- `@value` is at the first level of the array, outside of any element.
- A comment has two hyphens in a row or ends with a hyphen.

What the Exceptions Have

```
use Derafu\Xml\Exception\XmlException;

try {
    $document->loadXml('<r><a></r>');
} catch (XmlException $e) {
    $e->getMessage();        // "Failed to load the XML: Error Fatal: Opening and ending
tag mismatch: ..."
    $e->getErrors();          // ["Error Fatal: Opening and ending tag mismatch: a line 1
and r in line 1, column 11 (Code: 76).", ...]
    $e->getXmlDocument();    // The document that caused the exception, if there is one.
}
```

- `getErrors()` gives the errors that explain the exception, as text: the ones of libxml when an XML is loaded, and the validation errors (simplified) when it is validated. It is an empty list if there are none.
- `getXmlDocument()` gives the `XmlDocument` that caused the exception, or `null`.
- `XmlQueryException` also has `getXpathExpression()`.

When there are errors, the message has them (`{errors}`), so the message says what failed:

The XML validation failed using the schema schema.xsd: Field 'wrong' (line 1): was not expected, the expected was 'element'.

Translating the Messages

The exceptions are translatable (they are the ones of the `derafu/translation` library): their messages are in English, and the library has their translation to Spanish. To have the message in another language, use `trans()` with a translator:

```
use Derafu\Translation\TranslatorFactory;
use Derafu\Xml\Translation\XmlTranslationResourceProvider;

$translator = TranslatorFactory::create('es', [], [new
XmlTranslationResourceProvider()]);

try {
    $document->loadXml('');
} catch (XmlException $e) {
    echo $e->getMessage();        // "The XML content that you want to load is
empty."
    echo $e->trans($translator);  // "El contenido XML que se quiere cargar está
vacío."
}
```

The translations are in `resources/translations/errors+intl-icu.es.php`. In an application with the Symfony container, `resources/config/xml-services.yaml` of the library registers `XmlTranslationResourceProvider` with the tag `derafu_translation.resource_provider`, so its translations are loaded with the ones of the other libraries.

The errors of libxml that are in `getErrors()` are the texts that libxml gives (simplified in the validation): they are not translated.

Last updated on 17/10/2026