

Docs » Data Handling Category » UBL Project » Validation

Validation

Validating a UBL document's XML against the official OASIS UBL 2.1 XSD

Anonymous · 12/10/2026

Validation

`SchemaValidator` validates a document's XML directly against the official OASIS UBL 2.1 XSD bundled with the package - the same schema real UBL consumers (Peppol Access Points, tax authorities) validate against.

Validating a Document

```
use Derafu\Ubl\Service\SchemaValidator;

$validator = new SchemaValidator();

$validator->validate($xml); // Throws Derafu\Ubl\Exception\ValidationException if
invalid.
```

Nothing is returned on success: a document either validates, or the call throws.

Resolving the Document Type

Which XSD file to validate against is resolved automatically from the XML's own root element, the same way `Deserializer` resolves a target class for xml (see Self-Describing Content in **Formats**). An explicit class can be passed as a second argument to skip or override that:

```
use Derafu\Ubl\Oasis\Doc\Quotation;

$validator->validate($xml, Quotation::class);
```

Handling Validation Errors

A failed validation throws `Derafu\Ubl\Exception\ValidationException`, which exposes the real list of `LibXMLError` objects libxml reported, not only a flat message:

```
use Derafu\Ubl\Exception\ValidationException;
```

```
try {  
    $validator->validate($xml);  
} catch (ValidationException $e) {  
    foreach ($e->getErrors() as $error) {  
        printf("Line %d: %s\n", $error->line, trim($error->message));  
    }  
}
```

A malformed XML string, or a well-formed one whose root element matches no known UBL document, also throws `ValidationException` - with `getErrors()` empty in both cases, since neither one ever reaches schema validation itself: the document type could not be resolved to begin with.

Last updated on 12/10/2026