|  |  |  |  |
| --- | --- | --- | --- |
| **Übungsprotokoll**  **INSY - Informationssysteme** | | | |
|  | **Übungsdatum:**  KW 20/2021 –  KW 20/2021 | **Klasse:**  3AHIT | **Name:**  Felix Schneider |
| **Abgabedatum:**  19.05.2021 | **Gruppe:**  INSY\_2 | **Note:** |
| **Leitung:**  DI (FH) Alexander MESTL | **Mitübende:**  - | | |
| **Übungsbezeichnung**:  XML | | | |

**Inhaltsverzeichnis:**

[1 Aufgabenstellung 3](#_Toc106116333)

[1.1 XML Schema 3](#_Toc106116334)

[1.2 XPath 3](#_Toc106116335)

[2 Abstract (English) 3](#_Toc106116336)

[2.1 XML Schema 3](#_Toc106116337)

[2.2 XPath 4](#_Toc106116338)

[3 Theoretische Grundlagen 5](#_Toc106116339)

[3.1 XML Schema – kurz und knackig 5](#_Toc106116340)

[3.2 XPath – kurz und knackig 5](#_Toc106116341)

[4 Übungsdurchführung 6](#_Toc106116342)

[4.1 einfache Datentypen 6](#_Toc106116343)

[4.1.1 Übung 1 – Color 6](#_Toc106116344)

[4.1.2 Übung 2 – Personenname 7](#_Toc106116345)

[4.2 komplexe Datentypen 8](#_Toc106116346)

[4.2.1 Übung 1 – Book 8](#_Toc106116347)

[4.2.2 Übung 2 – Book2 9](#_Toc106116348)

[4.2.3 Übung 4 – Person 10](#_Toc106116349)

[4.3 XML (II) - XPath-Aufgaben 12](#_Toc106116350)

[4.3.1 Knotenbaum malen 12](#_Toc106116351)

[4.3.2 Lokalisierungspfade 13](#_Toc106116352)

[4.3.2.1 1. Beispiel 13](#_Toc106116353)

[4.3.2.2 2. Beispiel 17](#_Toc106116354)

[4.3.2.3 5. Beispiel 20](#_Toc106116355)

[5 Ergebnisse 26](#_Toc106116356)

[6 Code 26](#_Toc106116357)

[7 Kommentar 26](#_Toc106116358)

# Aufgabenstellung

## XML Schema

Mit Hilfe eines Schemas kann man die Ausgestaltung eines XML-Dokumentes definieren.

Nach dem Durcharbeiten des Kapitels 26 sollte es euch möglich sein, die drei Aufgaben zu den "einfachen Datentypen" auf den Seiten 24 und 25 des Aufgabendokuments "insyExercise3.pdf" (neu in den Kursmaterialien) zu lösen.

Zum besseren Verständnis hab ich ein Beispiel gemacht und euch einen Screenshot davon angefügt. Ich verwende IntelliJ, aber ihr könnt die IDE / den Editor selber frei wählen. Wie man sich Einträge für neue XML- und XSD-Dateien im "New"-Menü anlegt, findet ihr unter dem Jetbrains-Link!

Nach den einfachen Datentypen wenden wir uns den komplexen Datentypen zu. Das sind Datentypen, die aus mehreren Elementen einfacher Datentypen bestehen - das sollte nach dem Durchlesen des Kapitels 26 bereits bekannt sein.

Dazu machen wir die Beispiele Nr. 1 auf Seite 26, Nr. 2 auf Seite 27 und Nr. 4 auf Seite 28 des Übungsdokuments "insyExercise3.pdf".

Je nach IDE wird bereits während des Schreibens der XML-Datei bei gegebenem Schema die Korrektheit der Daten überprüft und/oder über einen eigenen Aufruf "Validierung" (oder so ähnlich). Wer nur mit einem Texteditor arbeitet, kann seine XML-Dokumente auch online validieren lassen, z. B. auf [https://www.xmlvalidation.com](https://www.xmlvalidation.com/).

## XPath

Aus dem Aufgabendokument "insyExercise3.pdf" ist das zweite Beispiel zu den XPath-Grundlagen auf Seite 33 (der "Knotenbaum") zu lösen.

Danach sind die Aufgaben "3.2. Lokalisierungspfade - 1. Beispiel" auf Seite 34, "3.2. Lokalisierungspfade - 2. Beispiel" auf Seite 35 und "3.2. Lokalisierungspfade - 5. Beispiel" auf Seite 36 durchzuführen. Die xml-Dateien sind davor auf ihre Wohlgeformtheit zu überprüfen und gegebenenfalls zu korrigieren.

# Abstract (English)

## XML Schema

You can use a schema to define the layout of an XML document.

After working through chapter 26 you should be able to solve the three tasks on "simple data types" on pages 24 and 25 of the task document "insyExercise3.pdf" (new in the course materials).

For better understanding I made an example and attached a screenshot of it for you. I use IntelliJ, but you are free to choose the IDE / editor yourself. How to create entries for new XML and XSD files in the "New" menu you can find under the Jetbrains link!

After the simple data types we turn to the complex data types. These are data types that consist of multiple elements of simple data types - this should already be familiar after reading through Chapter 26.

For this we do the examples no. 1 on page 26, no. 2 on page 27 and no. 4 on page 28 of the exercise document "insyExercise3.pdf".

Depending on the IDE, the correctness of the data is already checked during the writing of the XML file for a given schema and/or via a separate call "validation" (or something like that). Those who only work with a text editor can also have their XML documents validated online, e.g. at <https://www.xmlvalidation.com>.

## XPath

From the task document "insyExercise3.pdf", the second example on XPath basics on page 33 (the "node tree") is to be solved.

Afterwards the tasks "3.2. localization paths - 1st example" on page 34, "3.2. localization paths - 2nd example" on page 35 and "3.2. localization paths - 5th example" on page 36 have to be performed. The xml files must be checked beforehand for their well-formedness and corrected if necessary.

# Theoretische Grundlagen

detailliert: INSY-Script

## XML Schema – kurz und knackig

Es gibt zwei Datentypen: einfache, komplexe.

* Einfach Datentypen enthalten Standarddatentypen, wie zum Beispiel int, string, …
* Komplexe Datentypen enthalten verschachtelt einfache Datentypen.

Mit restriction kann man simpleTypes einschränken (Länge, Enum, …)

Mit selection kann man einstellen, welche Datentypen nacheinander im komplexen Datentyp vorkommen. Dabei nutzt man xs:element, um ein Element innerhalb des komplexen Typens zu beschreiben. Wenn ein Element dabei öfter vorkommen können soll, verwendet man das Attribut maxOccurs=“unbounded“, wie Sie beim letzten Beispiel sehen werden.

## XPath – kurz und knackig

Mithifle von XPath kann man Elemente selektieren (vergleichbar mit dem find-Befehl bei MongoDB).

Wikipedia:

Die XML Path Language (XPath) ist eine vom W3-Konsortium entwickelte Abfragesprache, um Teile eines XML-Dokumentes zu adressieren und auszuwerten. XPath dient als Grundlage einer Reihe weiterer Standards wie XSLT, XPointer und XQuery.

# Übungsdurchführung

## einfache Datentypen

### Übung 1 – Color

Ein Bild, das Text enthält.

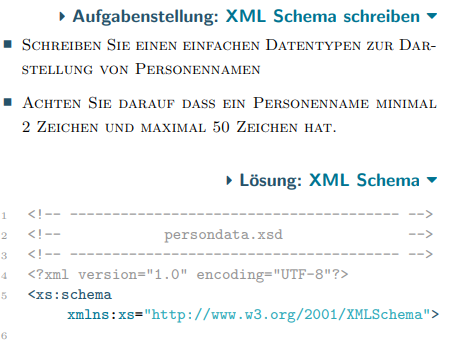
Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

### Übung 2 – Personenname



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

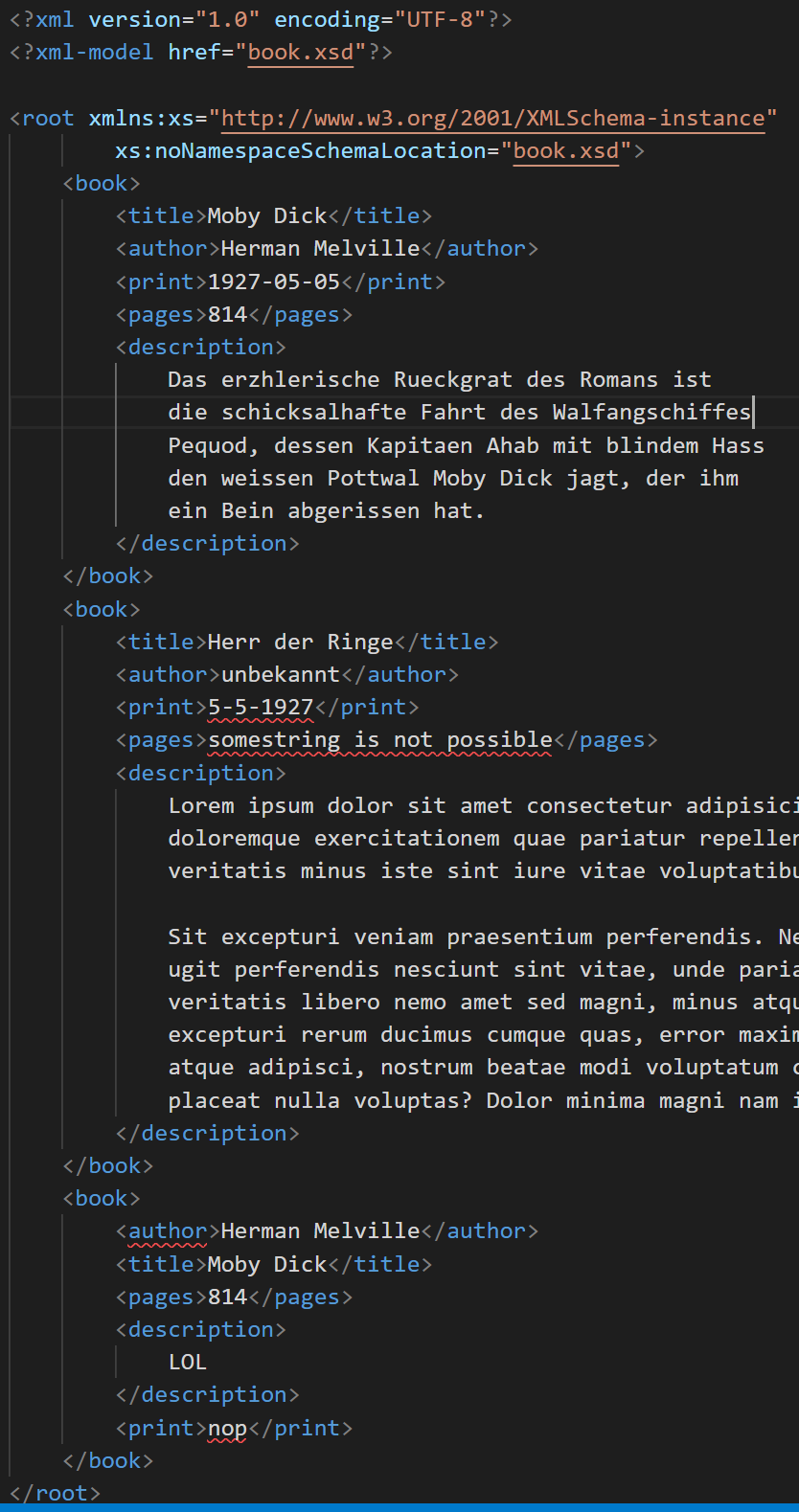
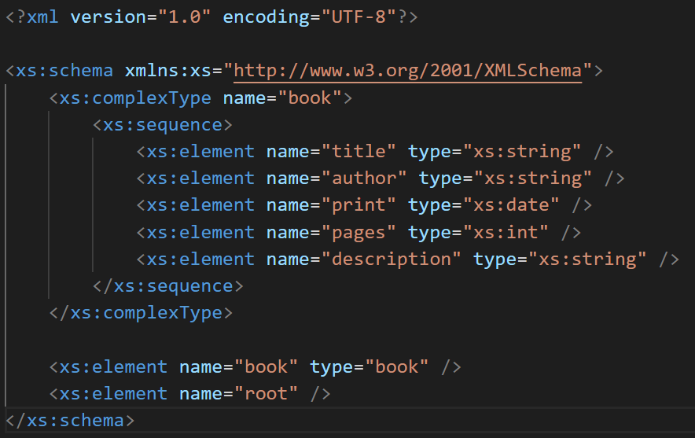
Automatisch generierte Beschreibung

## komplexe Datentypen

### Übung 1 – Book

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



### Ein Bild, das Text enthält. Automatisch generierte BeschreibungÜbung 2 – Book2

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

### Übung 4 – Person

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

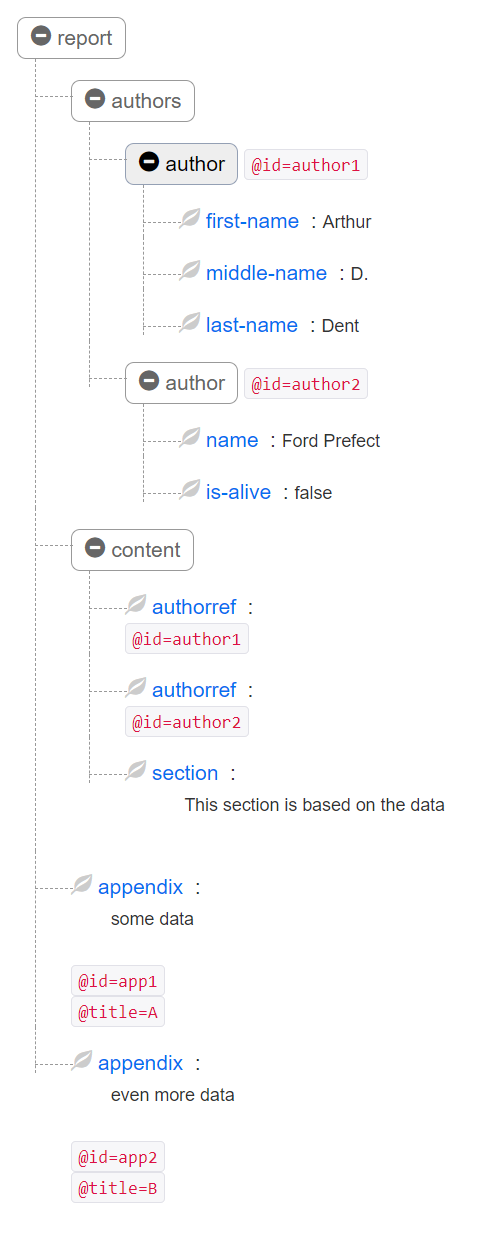
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## XML (II) - XPath-Aufgaben

### Knotenbaum malen

Dieser Baum wurde mithilfe von <https://www.xmlviewer.org/> erstellt:



Dieser Baum wird von oben nach unten durchnummeriert (die Reihenfolge ist liniear).

### Lokalisierungspfade

#### 1. Beispiel

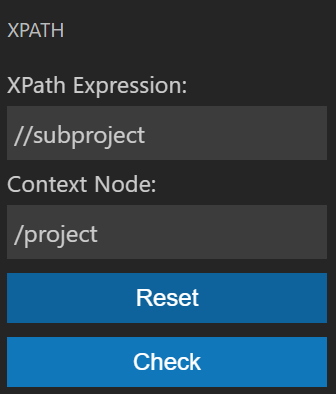
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

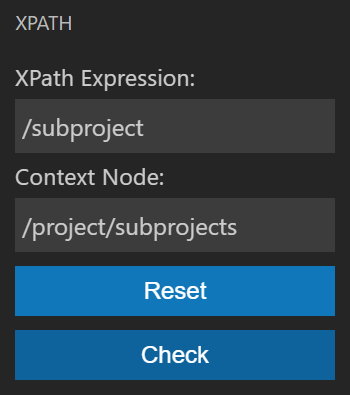
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

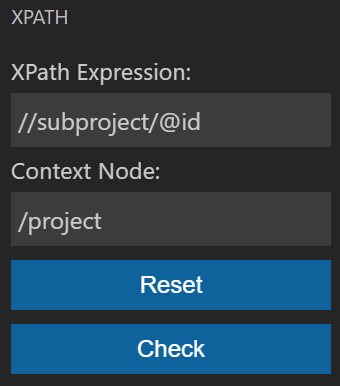
Automatisch generierte Beschreibung

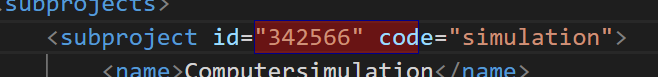
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

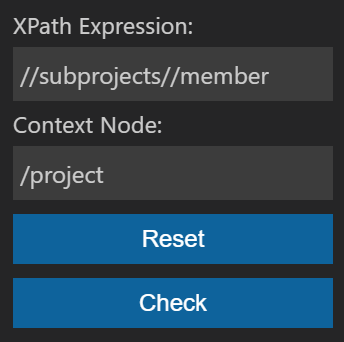




Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

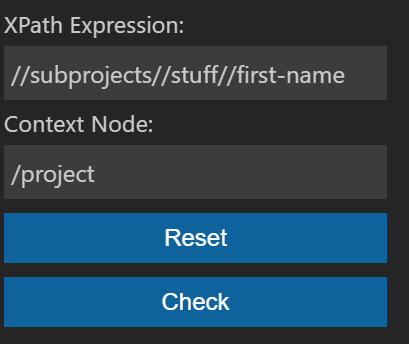
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

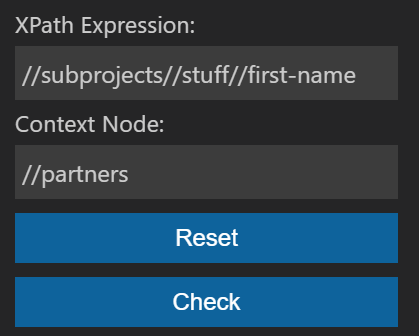
Automatisch generierte Beschreibung

Ein Bild, das Text, Flasche, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

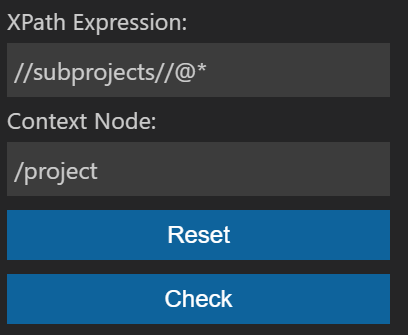
Automatisch generierte Beschreibung





Ein Bild, das Text enthält.

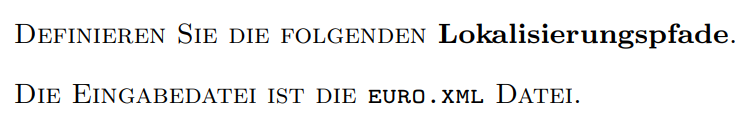
Automatisch generierte Beschreibung

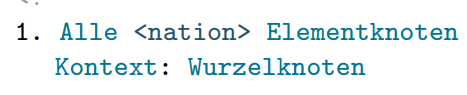


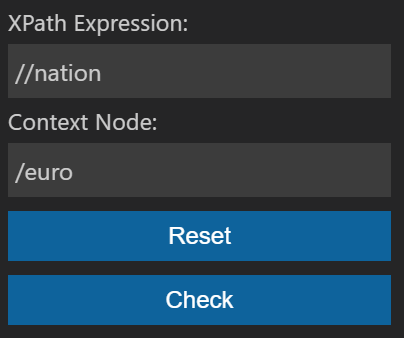
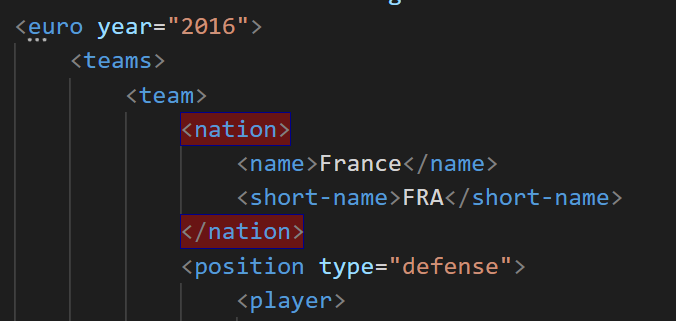
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### 2. Beispiel

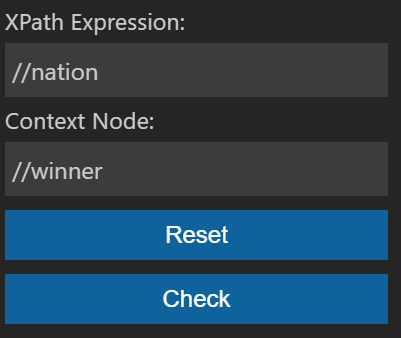






Ein Bild, das Text enthält.

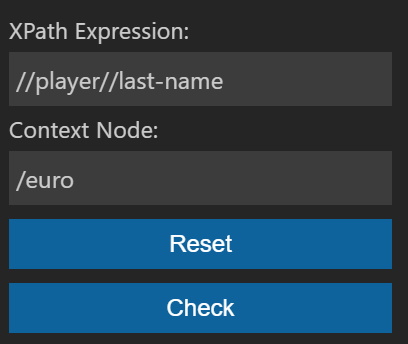
Automatisch generierte Beschreibung



Ein Bild, das Text enthält.

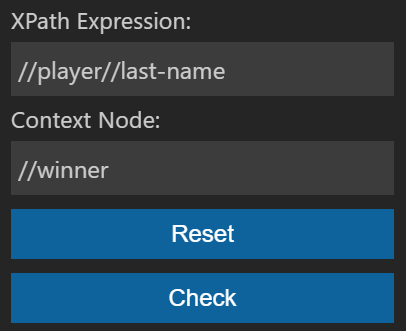
Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

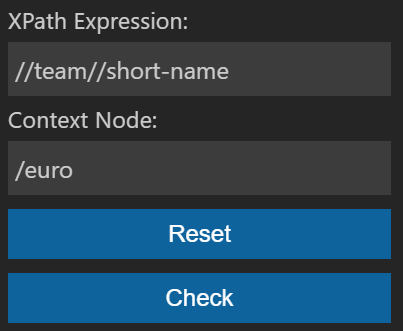
Automatisch generierte Beschreibung

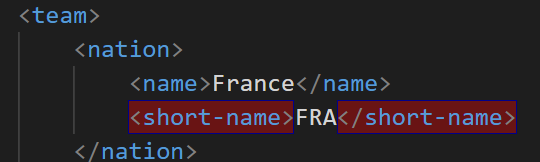




Ein Bild, das Text enthält.

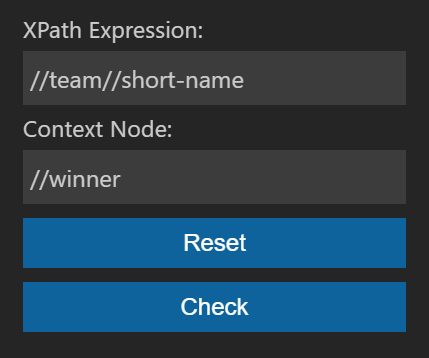
Automatisch generierte Beschreibung





Ein Bild, das Text enthält.

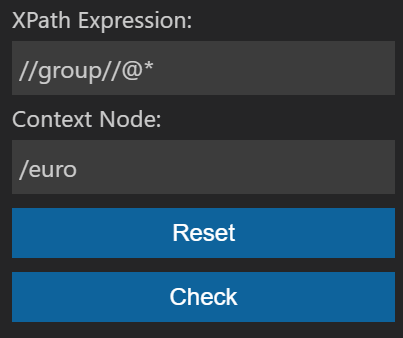
Automatisch generierte Beschreibung





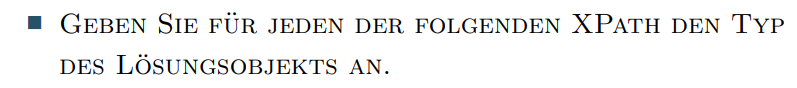
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

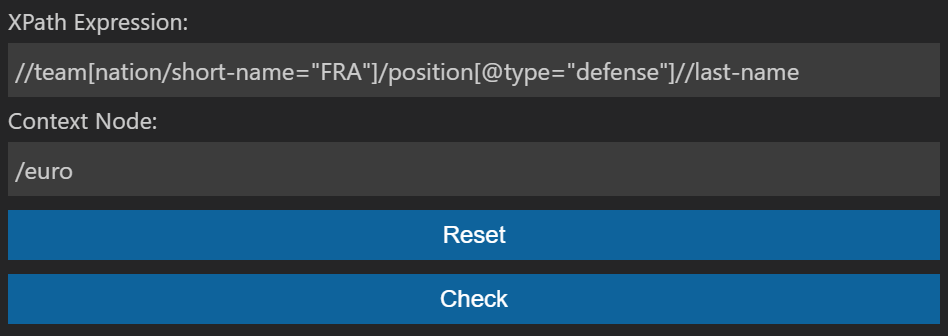
Automatisch generierte Beschreibung

#### 5. Beispiel



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

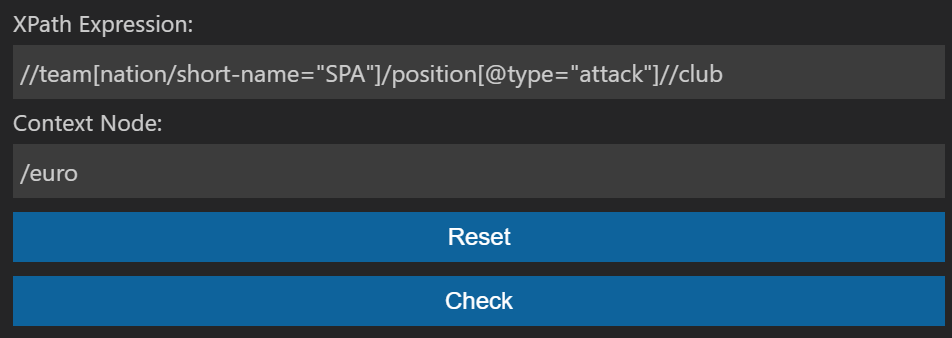


Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

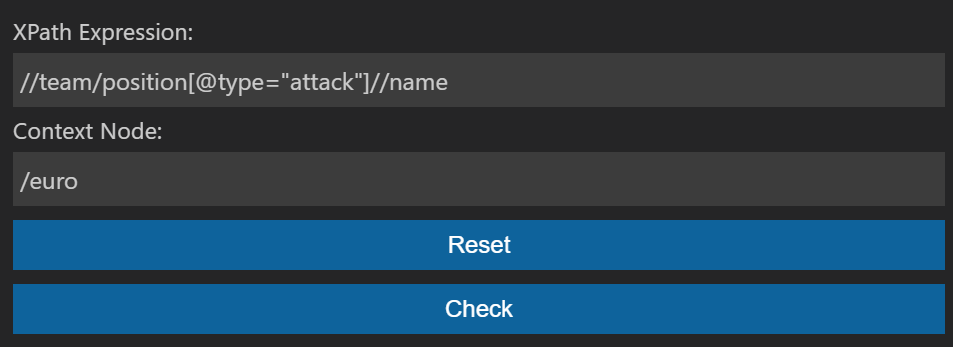


Ein Bild, das Text, silbern, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

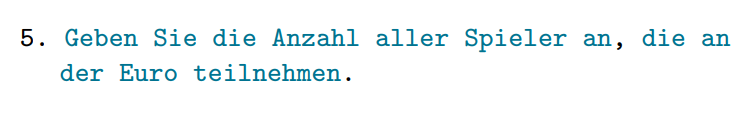
Ein Bild, das Text enthält.

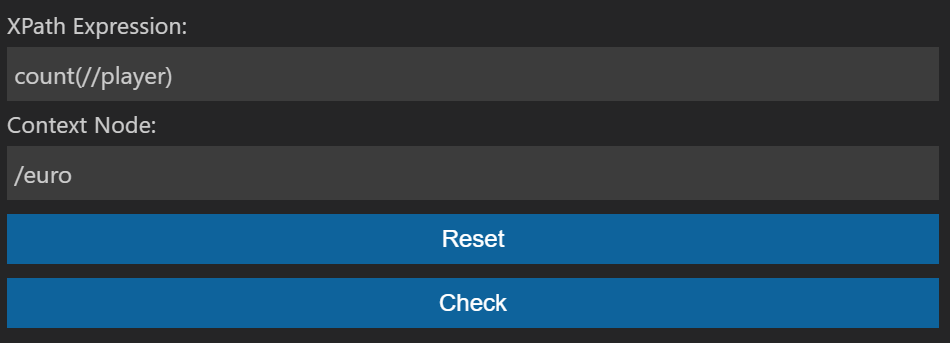
Automatisch generierte Beschreibung

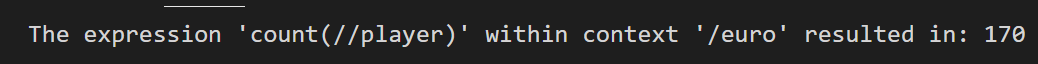


Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

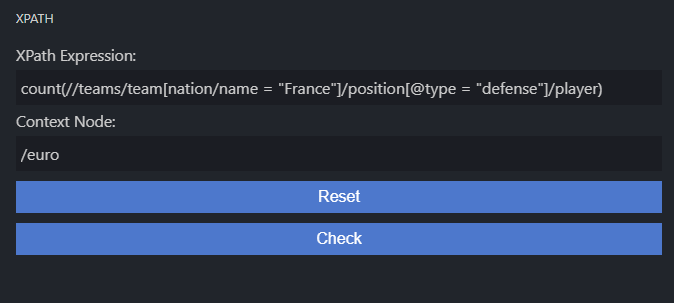


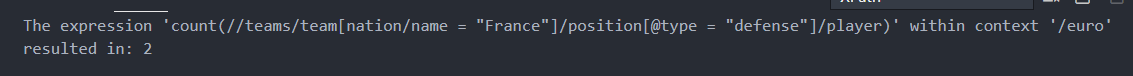




Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

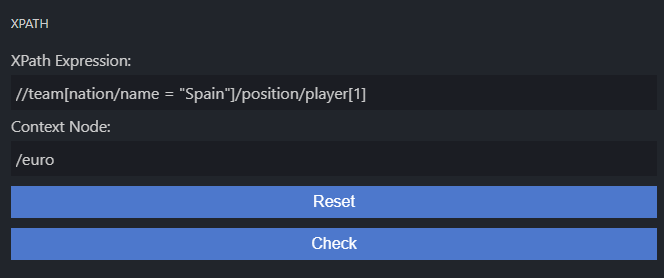






Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Ergebnisse

# Code

# Kommentar