



Fento

An adjusted approach for Java
objects / Xml Document Binding.

Xml Prague 2026
Jorge Sanchez
Vionta.net

Introduction

Introducing Fento.

Why should we consider this approach ?

Why should we care about Xml at this point ?

Why did I got into this?

Keys:

- Java annotations.
- Modern Xpath for describing document parts.
- Document is preserved, accessed and optionally updated. This is not an object serialization tool.
- Object and document data structure is not expected to be aligned.
- Repository abstraction.

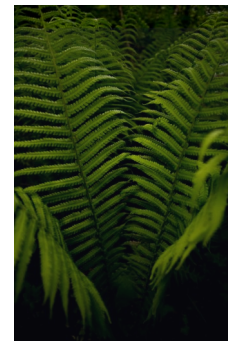
How does it look like ?

Repository configuration:

- Repository parameters.
- load / persist operation calls.

Annotated POJO classes:

- Mapping classes must implement the Serializable interface.
- No args constructor must be provided.
- Bind Annotation on mapped classes and fields.
 - The “expression” provides the Xpath Mapping.
 - It may include the namespaces or use the shorthand “*:” (not XPath 1.0 any more).

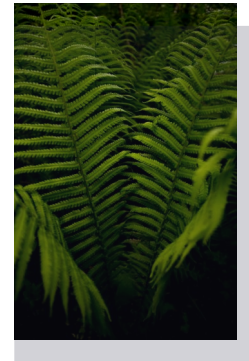
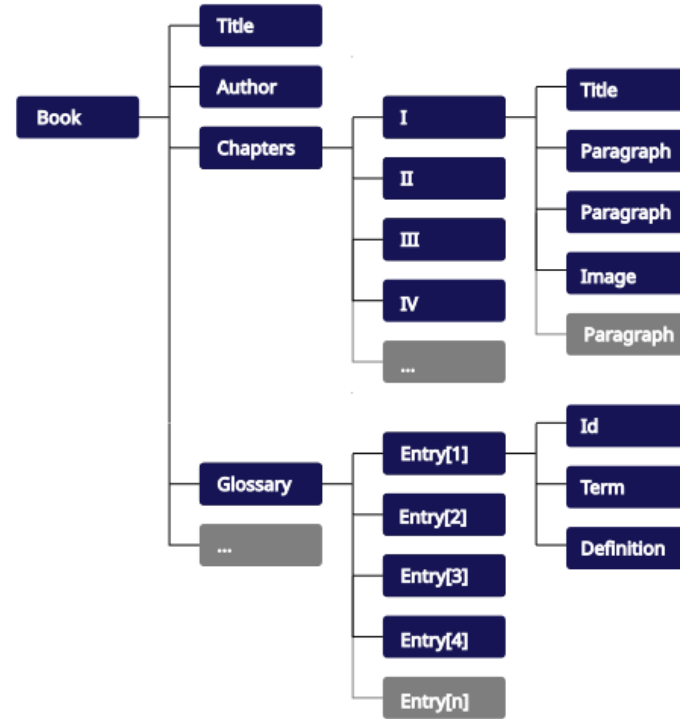


Why do we need XPath ?

Java and Xml data exploration differ.

Relations between variables are usually not observed or tracked on a object structure
 ¿ Which is a String parent or sibling ?

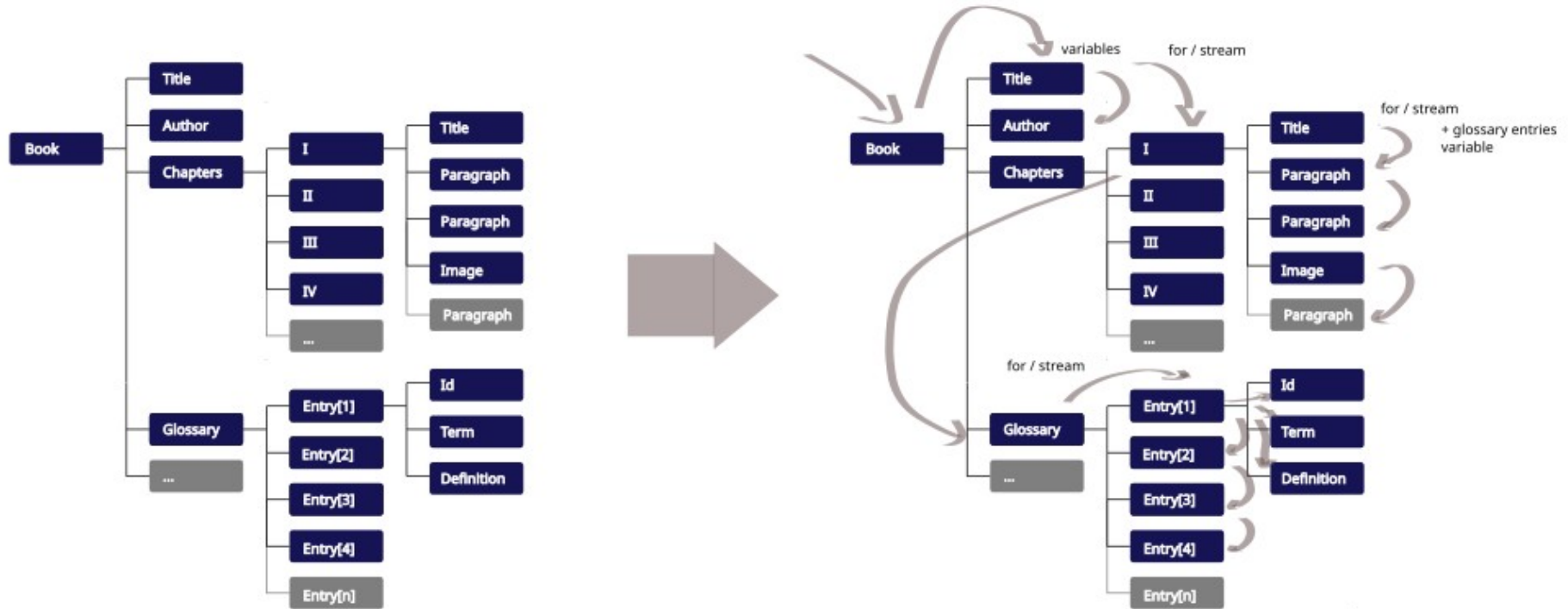
People from Java background may not be aware of how relevant relations between elements are in a document context.



Why do we need XPath ?



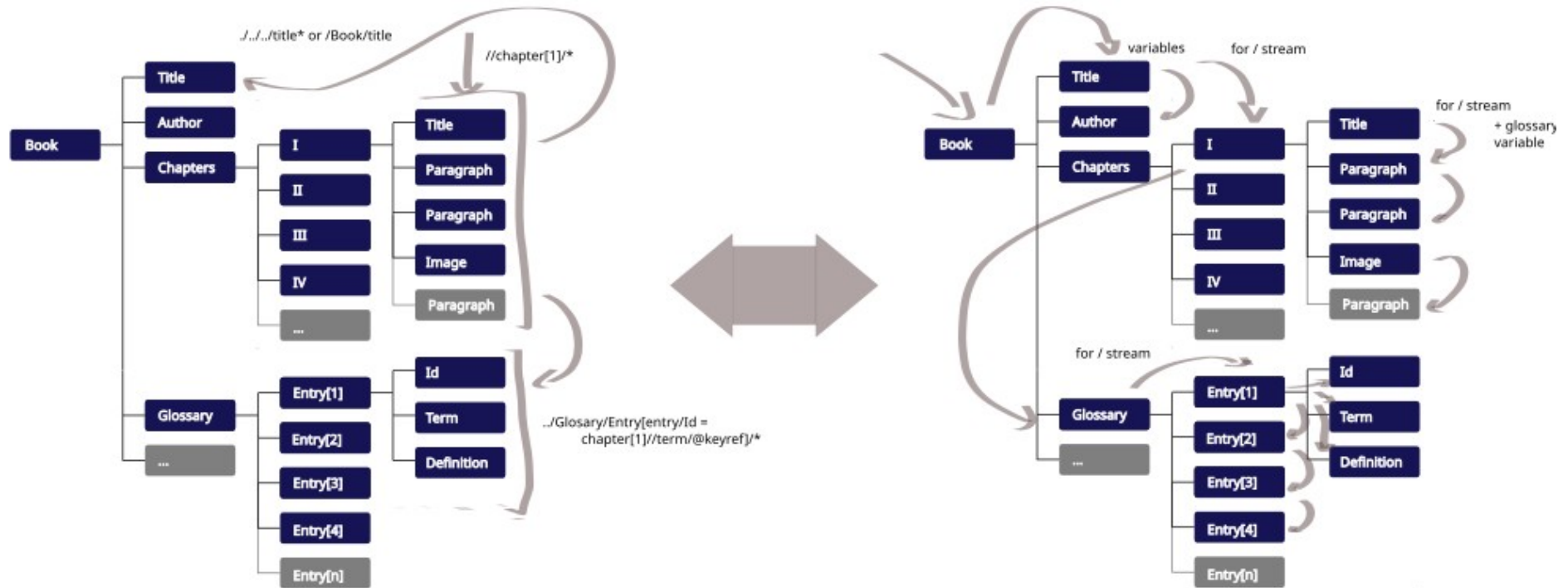
Java coders control the object data exploration with explicit steps.



Why do we need XPath ?



Xpath traversal can flow up or down and rely on element relations.

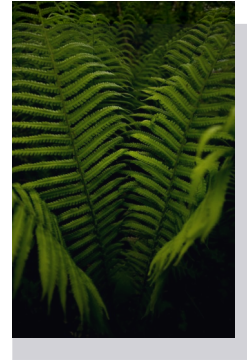


Securing the frontline



Java and the so called RESTful services and APIs are a popular choice for handling input from the web and enforce the security of frontend, facades and API interfaces. The more constrained JSON data model can reduce the exposed surface.

Combinations between more restrictive Java interfaces and Xml heterogeneous and flexible data on the inner side can enable solutions could not be solved efficiently with other approaches.



Working with element collections

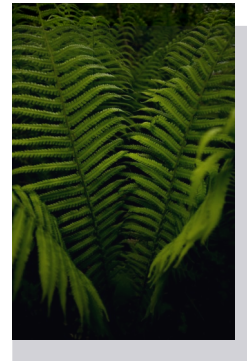


Types of collection mappings:

- By Key (requires one class field to be marked as "key=true")
- Position
- Full Reset (Serialization).

Approach Limitations.

- New Elements placement.
- Sorting, repeating, elements.
- Removing elements.



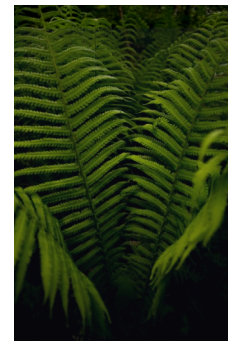


Repository Types

- Disk File/Folder based repository ("FilePathRepository").
- HttpRepository.
- ClassPathRepository.
- ZipArchiveRepository.
- Abstract Repository (Work in progress).

Repository Path Adjustment

```
FilePathRepository repository = new  
FilePathRepository("C:/collections/shows/concerts-{id}.xml");
```



Document complexity

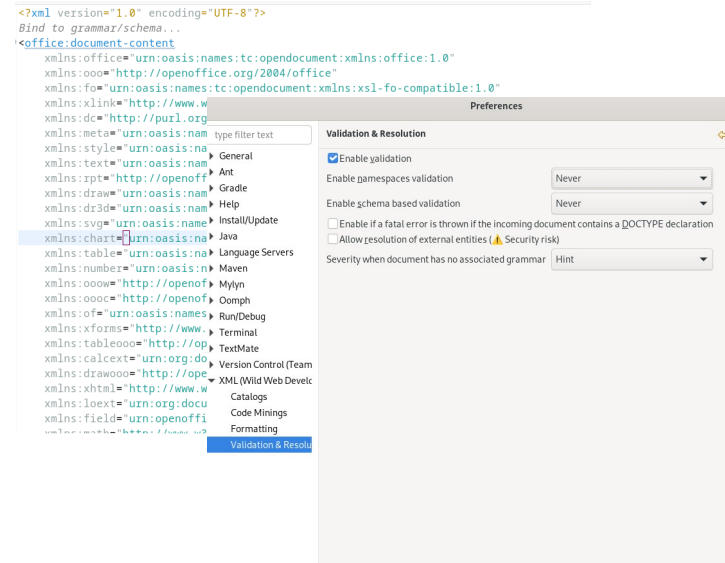


Xml documents can easily overcome the complexity of programming counterparts.

- Xml documents can hold from simple configuration files to dataset files capable of populating a full ER schema.
- Xml documents can include data, metadata, presentation instructions, graphical vector images, diagrams, coding instructions, office documents, etc. by itself.
- Xml documents often have multiple schema references although data structure can be checked for strict schema conformance. In many occasions Xml files are edited by hand allowing to add as many details as needed.

Tearing down and rebuild the documents from scratch may not be viable from a economic perspective.

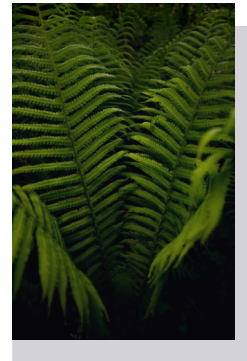
It does not make sense to drop and rebuild everything every time we do some task on xml data.



Limitations



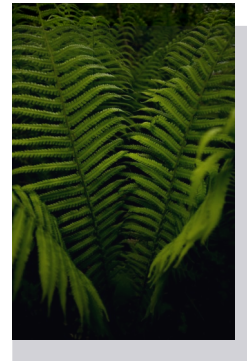
- The approach is not viable for more complex use cases.
 - Explicit navigation details, steps or specific behaviours.
 - Placement for element creation may not be precisely defined in XPath queries.
- Still a lot of small cases to be defined and implemented.
 - Sorting, rearranging, duplicated elements, etc.
- Provided code is a prototype, that needs revision and testing.
- The document context difficult a potential adoption.



Conclusion



- Balanced approach for Xml / Java Binding.
- Easy starting point from Java for a wide range to Xml solutions.
- Avoids complexity and mapping verbosity when addressing complex Xml structures, both with object binding or DOM.
- Preserves data information that may be easily lost with Serialization techniques.
- Fento next steps ?





Contact

Jorge Sanchez
jorgesanchez@vionta.net

Thanks for your attention

Materials

Fento repositories

<https://github.com/vionta/xesta>
<https://github.com/vionta/xesta-samples>

Fento repository (old)

<https://github.com/vionta/fento>
<https://github.com/vionta/fento-examples>

References

XML Prague

<https://www.xmlprague.cz>
XML Prague is a conference held in Prague about markup languages and data on the web.