

KtXml

Request for Comments

Avoid null values for strings everywhere

- More idiomatic kotlin – better null safety is the main reason for v2
- In most cases (names) empty is not a valid value, so it can be used to replace null, avoiding ambiguity
- Still use null values in
 - `getAttributeValue()`
 - Distinguish absence from “empty”
 - Easy to work around with `?: defaultValue`
 - `require()`
 - Used for wildcards

Turn EventType into an Enum

- Better type safety
- No longer targeting MIDP, so should be a reasonable memory / code size tradeoff

Add an XML_DECL event

- Just represent version, encoding, standalone as attributes instead of cluttering the API with methods infrequently used?

Give up on parser re-use

- Just feels wrong today

~~Don't provide an encoding for the InputStream~~

- ~~• There are clear rules to infer the encoding – providing an explicit extra default encoding seems like an extra option for confusion / ambiguity~~
- ~~• Just use the reader if the encoding should be fixed~~
- ~~• Can check if reader / xml decl encoding match by inspecting XML_DECL if desired.~~

Don't support Reader or InputStream

- Not available for Kotlin Native
- Not sure if CharSequence (current state) is the best alternative...

Turn the factory into an interface and drop the “parser” name part

- “Parser” in the name seems wrong as it also provides serialized factory methods
- Seems to be better for integration with whatever DI or bootstrapping clients want to use
- Should be straightforward to add a meta-implementation to be parser independent
- Java loading mechanism seems hard to replicate for KMP
- Store “configuration” aspects – but supply readers / writers to factory methods.
- Problem / TBD:
 - Factories creating hard dependencies on “unneeded” stuff (wbxml, serializer)
 - Should be simple to cut by proguard / static analysis though

Drop “dynamic” features / properties

- We'll add typed features / properties as needed instead
- Alternatively / additionally, factories can be extended