TEIModules

User's Guide

André Röhrig

TEIModules: User's Guide

by André Röhrig

Version ${project.version}

Table of Contents

List of Tables

Foreword

The intention of this document is first to give a guide to the user of how to use the here mentioned Pepper modules and how to utilize a mapping performed by them. Second this document shall give a closer view in the details of such a mapping in a declarative way, to give the user a chance to understand how specific data will be mapped by the presented Pepper modules.

Chapter 1. Overview

This project contains the Pepper modules listed in Table 1.1, “Pepper modules contained in this project”. A single module can be identified via its coordinates (module-name, format-name, format-version) also given in Table 1.1, “Pepper modules contained in this project”. You can use these coordinates in a Pepper workflow description file to identify the modules in a Pepper conversion process. A description of how to model a workflow description file can be found under https://korpling.german.hu-berlin.de/saltnpepper/.

Table 1.1. Pepper modules contained in this project

Name of Pepper module	Type of Pepper module	Format (if module is im- or exporter)
TEIImporter	importer	TEI 2.6.0

Chapter 2. Changes

This chapter contains the changes in version ${project.version} compared to the previous version.

Chapter 3, *TEIImporter*

- This is the first release. Thus there are no changes.

Chapter 3. TEIImporter

General information about this importer.

Mapping to Salt

The fact that TEI is a XML-format results in the decision to primarily use "SStructure" while mapping TEI to Salt. There are two important exceptions to this: Tokens("SToken" is used) and the unary "break" elements like <lb> and <pb>(these cannot be mapped as "SStructure" because their semantic does not fit into the hierarchy provided by XML). Instead, "SSpan" is used. Tokens can be defined and interpreted in many different ways and thus customization through properties deal with the problems occuring because of this.

Properties

Because TEI is a very complex format the behavior of the TEIImporter depends to a great extent on the properties that the user can use to customize the behaviour of the TEIImporter. The table Table 3.1, “properties to customize importer behaviour” contains an overview of all usable properties to customize the behaviour of the TEIImporter. The following section contains a close description to each single property and describes the resulting differences in the mapping to the Salt model.

Table 3.1. properties to customize importer behaviour

Name of property	Type of property	optional/ mandatory	default value
TEIImporter.DefaultTokenization	Boolean	optional	false
TEIImporter.SubTokenization	Boolean	optional	true
TEIImporter.SurplusRemoval	Boolean	optional	true
TEIImporter.UnclearAsToken	Boolean	optional	true
TEIImporter.ForeignAsToken	Boolean	optional	true
TEIImporter.UseTokenizer	Boolean	optional	false
TEIImporter.UseTokenizerLang	String	optional	en

TEIImporter.DefaultTokenization

The user declares that there is one and only element responsible for mapping tokens to Salt. Default is <w>.

TEIImporter.SubTokenization

In this scenario, units smaller than ‘words’ exist. Elements within <w> etc. are possible.

TEIImporter.SurplusRemoval

Will text from <surplus> appear in Salt? If this is set "true" (default), then <surplus>-text will be removed.

TEIImporter.UnclearAsToken

Does <unclear> exclusively create one token annotated as "unclear"? If this is set "true" (default), thentext in <unclear> will appear as a token.

TEIImporter.ForeignAsToken

Does <foreign> exclusively create one token annotated as "foreign"? If this is set "true" (default), then text in <foreign> will appear as a token.

TEIImporter.UseTokenizer

Do you want the tokenizer to tokenize text? This option is useful, if your TEI document contains sections of text that are not tokenized.

TEIImporter.UseTokenizerLang

The tokenizer currently has support for four languages: English, German, Italian, French. To choose a language, use the respective ISO 639-1 language code(en, de, it, fr). If no value or a non-supported value is set, the tokenizer will default to English.