Datenmeister – A documentation

# Introduction

This document gives you a short, unsorted overview through certain properties of the DatenMeister.

# DatenMeister Core

## Views and transformation definition

DatenMeister supports an extensive definition format to create views on certain extents which can be used for reading and writing.

## APIs

### ReferenceReformatter

The reference formatter takes an matrix-encoded list of properties and references and converts these references to subitems.

Example:

<?xml version="1.0" encoding="utf-8"?>

<xmi>

  <item p2:id="Items.Ironore" Material="Ironore" Time="2" />

  <item p2:id="Items.Copperore" Material="Copperore" Time="2" />

  <item p2:id="Items.Ironplate" Material="Ironplate" Time="3.5" Ironore="1" />

  <item p2:id="Items.Copperplate" Material="Copperplate" Time="3.5" Copperore="1" />

  <item p2:id="Items.Steelplate" Material="Steelplate" Time="17.5" Ironplate="5" />

</xmi>

This matrix-encoded list where the property-values are stored as direct properties can also be transferred to a list:

<xmi>

  <item p2:id="Items.Ironore" Material="Ironore" Time="2" />

  <item p2:id="Items.Copperore" Material="Copperore" Time="2" />

  <item p2:id="Items.Ironplate" Material="Ironplate" Time="3.5">

    <part name="Ironore" amount="1" />

  </item>

  <item p2:id="Items.Copperplate" Material="Copperplate" Time="3.5">

    <part name="Copperore" amount="1" />

    </item>

  <item p2:id="Items.Steelplate" Material="Steelplate" Time="17.5">

    <part name="Ironplate" amount="5" />

  </item>

</xmi>

The function is hosted in class “ReferenceReformatter” and has the following configuration.

/// <summary>

/// This is a list of properties that will be transfered as subitems

/// </summary>

public IList<string> FixedProperty { get; } = new List<string>();

/// <summary>

/// The property that will be created an which will host the subitems

/// </summary>

public string SubItemProperty { get; set; }

/// <summary>

/// The name of the property in subitem that will store the reference to the item

/// </summary>

public string ReferenceProperty { get; set; }

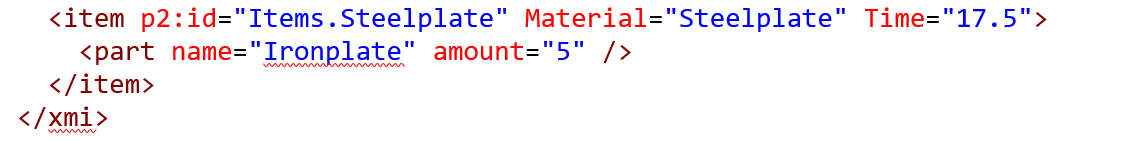
/// <summary>

/// The name of the property in subitem that will store the value of the item

/// </summary>

public string ContentProperty { get; set; }

The properties are explained in the following diagram:



*FixedProperty*

*SubItemProperty*

*ContentProperty*

*ReferenceProperty*

# DatenMeister Web

## Web-URLs

The Datenmeister-Webinterface is reachable via:

http://{host}:{port}/

The parameters are added as a fragment which is attached to the url itself:

http://{host}:{port}/#fragment

The fragment consists of a list of assignments via {key}={value}, the values are urlencoded (RFC 3986) and combined as a query string via the &-symbol. As an example:

<http://localhost:8080/#ws=Data&ext=datenmeister%3A%2F%2F%2Fzipcodes>

The following views are supported:

## Workspace view

Parameter:

* ws: Name of the workspace

Example:

<http://localhost:8080/#ws=Data>

## Extent view

Parameter:

* ws: Name of the workspace
* ext: Url of the extent

Example:

<http://localhost:8080/#ws=Data&ext=datenmeister%3A%2F%2F%2Fzipcodes>

## Item view

Parameter:

* ws: Name of the workspace
* ext: Url of the extent
* item: Extent-Url of the item
* mode:
  + “default”: If the user shall be able to modify the item within the given view
  + “readonly”: If the user shall just see the contents of the items but not be able to modify them. He will have an edit button.

Example:

<http://localhost:8080/#ws=Data&ext=datenmeister%3A%2F%2F%2Fzipcodes&item=datenmeister%3A%2F%2F%2Fzipcodes%23b1d17285-b38a-4178-8644-ea63feae56e1>