Escenic Content Engine

Bean Reference

5.2.7.2


vizrt

Disclaimer

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied.

This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Technical Support

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at www.vizrt.com.

Last Updated

21.12.2011


vizrt

Table of Contents

1 Introduction

This manual contains reference material for the Java **beans** supplied with the Escenic Content Engine. A bean is a Java object that complies with the JavaBeans naming conventions. Any object that complies with these conventions can be easily accessed using Java Server Pages (JSP) tags and the JSP expression language.

The beans supplied with the Content Engine fall into two categories:

Core Beans
: These beans represent the core components of an Escenic publication: the publication itself, content items and sections. They are automatically created by the system when a request is received and made available as request scope attributes for use in your Java Server Pages.

Utility Beans
: These beans represent other data structures created by the Content Engine for specific purposes: search results, tree structures and so on.

A bean is used by retrieving its **properties**, values that can be used in the construction of web pages. A content item, for example, is represented by a **`PresentationArticle`** bean (see **`section 2.2`**), which exposes all the content item's fields in a property called **`fields`**. The article in an Escenic request is made available as a request scope **`PresentationArticle`** variable called **`article`**. This means that you can retrieve the content of article's **`title`** field with the following expression:

```
${article.fields.title}
```

In order to make proper use of the Content Engine's beans you need background knowledge both about the Content Engine itself and about the technologies it is based on such as Java Server Pages, JSTL, the JSP expression language, Java Beans and so on. You can start by reading the **Escenic Content Engine Template Developer Guide**, followed by the **Escenic Content Engine Advanced Developer Guide** if you want more information. The **Template Developer Guide** contains a **What Next?** section containing links and references to sources of general background information.

This manual contains brief descriptions of all the Content Engine beans, followed by descriptions of the bean properties, including examples of how to retrieve the properties.

A number of the beans and properties described in this manual are **deprecated**: that is, you are recommended to avoid using them. They are old beans/properties that are no longer needed and may eventually be withdrawn. They are currently still available in order to ensure that older applications that use them will still work. You should not, however, use them in new applications.

All deprecated beans and properties are clearly marked.

2 Core Beans

The core beans represent central components of Escenic publications.

2.1 Link

Represents a hypertext link.

`Link` has the properties described in the following sections.

2.1.1 href

The link's URI.

Type: **`java.net.URI`**

Example usage

```
${myLink.href}
```

2.1.2 localFile

Checks whether or not the link points to a file in the local file system.

Type: **`boolean`**

Example usage

```
${myLink.localFile}
```

2.1.3 mimeType

The link's MIME type.

Type: **`java.lang.String`**

Example usage

```
${myLink.mimeType}
```

2.1.4 title

The link's title.

Type: **`java.lang.String`**

Example usage

```
${myLink.title}
```

2.2 PresentationArticle

Represents a **content item** - the basic component of an Escenic publication.

The actual content of a content item is stored in a **`PresentationArticle`**'s **fields**, and the particular fields a content item has depend on its type: a review article, for example, will usually have different fields than a news article, and will definitely have different fields than an image or media object content item.

All of a **`PresentationArticle`**'s content and relations can be accessed via its properties.

In version 5 of the Escenic Content Engine, a **`PresentationArticle`** can represent any type of content item - not just a text article such as a magazine article or newspaper story but also an image, media object or link. In older applications, however, these kinds of content item are represented by **`PresentationRelationImage`**, **`PresentationRelationMedia`**, **`PresentationRelationLink`** and so on. These older object types are still supported to ensure compatibility with older applications. Their use is deprecated, however, and you should not use them in new applications.

`PresentationArticle` has the properties described in the following sections.

2.2.1 activatedDate

The date this content item was activated. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.activatedDate}
```

2.2.2 activatedDateAsDate

The date this content item was activated.

Type: **`java.util.Date`**

Example usage

```
${myPresentationArticle.activatedDateAsDate}
```

2.2.3 agreementInfo

This content item's agreement information or **`null`** if the content item has no agreement.

Type: **`neo.xredsys.api.AgreementInfo`**

Example usage

```
${myPresentationArticle.agreementInfo}
```

2.2.4 agreementRequired

`true` if an agreement is required to view this content item, otherwise **`false`**.

Type: **`boolean`**

Example usage

```
${myPresentationArticle.agreementRequired}
```

2.2.5 allRelatedObjects

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the objects of any type (links, images, media objects and other content items) related to this content item.

Type: **`java.util.List<PresentationRelation>`**

Example usage

To get all the objects:

```
${myPresentationArticle.allRelatedObjects}
```

To get one particular object:

```
${myPresentationArticle.allRelatedObjects[index]}
```

where *index* is the index of the object you want.

2.2.6 allRelatedObjectsCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of objects of any type (links, images, media objects and other content items) related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.allRelatedObjectsCount}
```

2.2.7 articleId

The database id of this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.articleId}
```

2.2.8 articleTypeName

The type of this content item (as defined in the **`content-type`** resource).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.articleTypeName}
```

2.2.9 articles

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the content items related to this content item.

Type: **`java.util.List<PresentationRelationArticle>`**

Example usage

To get all the content items:

```
${myPresentationArticle.articles}
```

To get one particular content item:

```
${myPresentationArticle.articles[index]}
```

where *index* is the index of the content item you want.

2.2.10 articlesCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of content items related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.articlesCount}
```

2.2.11 author

The author of this content item. If the content item has more than one author then the first author in the list is returned.

Type: **`neo.xredsys.api.Person`**

Example usage

```
${myPresentationArticle.author}
```

2.2.12 authors

All the authors of this content item.

Type: `java.util.List<Person>`

Example usage

To get all the authors:

```
${myPresentationArticle.authors}
```

To get one particular author:

```
${myPresentationArticle.authors[index]}
```

where *index* is the index of the author you want.

2.2.13 authorsCount

The number of authors responsible for this content item.

Type: `int`

Example usage

```
${myPresentationArticle.authorsCount}
```

2.2.14 createdDate

The date this content item was created. Created in this context means "created in the Escenic system". For a content item that has been imported from an external source, therefore, it is the import date and not the original creation date. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.createdDate}
```

2.2.15 createdDateAsDate

The date this content item was created. Created in this context means "created in the Escenic system". For a content item that has been imported from an external source, therefore, it is the import date and not the original creation date.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.createdDateAsDate}
```

2.2.16 expireDate

The date on which this content item expired/will expire. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.expireDate}
```

2.2.17 expireDateAsDate

The date on which this content item expired/will expire.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.expireDateAsDate}
```

2.2.18 fieldNames

The names of all this content item's fields. The names are listed in the order they are defined in `content-types.xml`, and in upper case letters.

Type: `java.util.List<String>`

Example usage

To get all the names:

```
${myPresentationArticle.fieldNames}
```

To get one particular name:

```
${myPresentationArticle.fieldNames[index]}
```

where *index* is the index of the name you want.

2.2.19 fields

All the fields in this content item. The actual type of the Java `Objects` representing the fields depends upon field type as follows:

Field type	Java Object
BASIC	`java.lang.String`
NUMBER	a subclass of `java.lang.Number`
COMPLEX	`java.util.Map`
BOOLEAN	`java.lang.Boolean`
ENUMERATION	a `java.lang.String` or `java.util.List` of Strings
URI	`java.net.URI`

Field type	Java Object
LINK	`com.escenic.domain.Link` (see section 2.1)
DATE	`java.util.Date`
SCHEDULE	`neo.xredsys.presentation.PresentationSchedule` (see section 2.9)

Type: `java.util.Map<String,Object>`

Example usage

To get all the fields:

```
${myPresentationArticle.fields}
```

To get one particular field:

```
${myPresentationArticle.fields.field}
```

where *field* is the key of the field you want.

2.2.20 firstPublishedDate

The date on which this content item was first published. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.firstPublishedDate}
```

2.2.21 firstPublishedDateAsDate

The date on which this content item was first published.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.firstPublishedDateAsDate}
```

2.2.22 hashKey

The hash key of this object.

Type: `neo.xredsys.api.IOHashKey`

Example usage

```
${myPresentationObject.hashKey}
```

2.2.23 homeSection

The home section of this content item.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPresentationArticle.homeSection}
```

2.2.24 id

The id of the presentation object

Type: `int`

Example usage

```
${myPresentationObject.id}
```

2.2.25 lastModifiedDate

The date on which this content item was last modified. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.lastModifiedDate}
```

2.2.26 lastModifiedDateAsDate

The date on which this content item was last modified.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.lastModifiedDateAsDate}
```

2.2.27 live

`true` if this content item is published and active and its home section is published, otherwise `false`.

Type: `boolean`

Example usage

```
${myPresentationArticle.live}
```

2.2.28 owner

`true` if this content item is in its owning publication, **`false`** if it is a cross-published content item that is being accessed from a foreign publication.

Type: **`boolean`**

Example usage

```
${myPresentationArticle.owner}
```

2.2.29 ownerHomeSection

The owner home section of this content item. This is the real home section which may be part of another publication.

Type: **`neo.xredsys.api.Section`**

Example usage

```
${myPresentationArticle.ownerHomeSection}
```

2.2.30 ownerUrl

The absolute URL of this content item **in its owning home section**. This may be different from **`url`** (see **section 2.2.51**) , which returns the content item's URL in its current section. For a cross-published content item both the section and publication components of the URL will differ.

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.ownerUrl}
```

2.2.31 persons

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the persons related to this content item.

Type: **`java.util.Collection<PresentationPerson>`**

Example usage

To get all the persons:

```
${myPresentationArticle.persons}
```

To access each person in turn:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<c:forEach var="person" items="${myPresentationArticle.persons}">
  <!-- do something with each ${person} -->
</c:forEach>
```

2.2.32 personsCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of persons related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.personsCount}
```

2.2.33 publication

The publication this content item belongs to.

Type: **`neo.xredsys.api.Publication`**

Example usage

```
${myPresentationArticle.publication}
```

2.2.34 publicationId

The database id of the publication this content item belongs to.

Type: **`int`**

Example usage

```
${myPresentationArticle.publicationId}
```

2.2.35 publishedDate

The date on which this content item was last published. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.publishedDate}
```

2.2.36 publishedDateAsDate

The date on which this content item was last published.

Type: **`java.util.Date`**

Example usage

```
${myPresentationArticle.publishedDateAsDate}
```

2.2.37 relatedElements

A Map from relation type to a **`PresentationElement`** (see **section 2.3**) that holds the relations to this object.

Type: **`java.util.Map`**

Example usage

```
${myPresentationArticle.relatedElements}
```

2.2.38 relativeURI

The URL of this content item relative to its section. Usually this will be an URI object containing a file name with no path component, such as **`article123.ece`**.

Type: **`java.net.URI`**

Example usage

```
${myPresentationArticle.relativeURI}
```

2.2.39 relativeUrl

This property is deprecated: use the **`relativeURI`** (see **section 2.2.38**) instead.

The URL of this content item relative to its section. Usually this will be a file name with no path component, such as **`article123.ece`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.relativeUrl}
```

2.2.40 sections

This property is deprecated: Replaced by getSectionsList(). Should not be used.

The sections this content item belongs to.

Type: **`neo.xredsys.api.Section`**

Example usage

```
${myPresentationArticle.sections}
```

2.2.41 sectionsList

The sections this content item belongs to.

Type: **`java.util.List<Section>`**

Example usage

To get all the sections:

```
${myPresentationArticle.sectionsList}
```

To get one particular section:

```
${myPresentationArticle.sectionsList[index]}
```

where *index* is the index of the section you want.

2.2.42 source

The name of the external source of an imported content item. A content item that has been imported from Associated Press, for example, might have an external source of "AP". **`source`** together with **`sourceId`** (see **section 2.2.43**) uniquely identifies the original source of the content item.

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.source}
```

2.2.43 sourceId

The id of the external source of an imported content item. The form of the id depends upon the source: it might be some kind of numerical identifier or it might be URI of a news feed, for example. **`sourceId`** together with **`source`** (see **section 2.2.42**) uniquely identifies the original source of the content item.

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.sourceId}
```

2.2.44 stateChangedDate

The date on which the state of this content item last changed. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.stateChangedDate}
```

2.2.45 stateChangedDateAsDate

The date on which the state of this content item last changed.

Type: **`java.util.Date`**

Example usage

```
${myPresentationArticle.stateChangedDateAsDate}
```

2.2.46 stateName

The state of this content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.stateName}
```

2.2.47 tags

A List of presentation tags.

Type: `java.util.List<PresentationTag>`

Example usage

To get all the tags:

```
${myPresentationArticle.tags}
```

To get one particular tag:

```
${myPresentationArticle.tags[index]}
```

where *index* is the index of the tag you want.

2.2.48 title

This property is deprecated: No replacement. Should not be used.

The title of this content item. The title of a content item is the content of the field defined as the title field in its content type definition in the content-types resource.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.title}
```

2.2.49 topic

The topic to which this content item belongs. If the content item belongs to more than one topic then the first topic in the list is returned. If the content item does not belong to a topic then `null`?? is returned. Note that only topics in the same publication as the content item are returned.

Type: `neo.xredsys.api.Topic`

Example usage

```
${myPresentationArticle.topic}
```

2.2.50 topicsList

All the topics this content item belongs to. If the content item does not belong to a topic then an empty list is returned. Note that only topics in the same publication as the content item are returned.

Type: `java.util.List<Topic>`

Example usage

To get all the topics:

```
${myPresentationArticle.topicsList}
```

To get one particular topic:

```
${myPresentationArticle.topicsList[index]}
```

where *index* is the index of the topic you want.

2.2.51 url

The absolute URL of this content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.url}
```

2.3 PresentationElement

Represents a **group**, an **area** or a **summary** in an Escenic **section page** or an article's **related items**. A section page is represented by a `PresentationPool` bean, which owns a single root group (`PresentationElement`), accessed via its `rootElement` property. This root group is usually called the grid}. The following JSP expression returns the `title` field of a summary in the first area of the `header` group of a section page. The section page is held in a variable called `page`:

- A group `PresentationElement` has contents in its `keys` (see **section 2.3.5**) , `areas` (see **section 2.3.1**) and `type` (see **section 2.3.7**) properties.
- An area `PresentationElement` has contents in its `items` (see **section 2.3.4**) and `type` (see **section 2.3.7**) properties.
- A summary `PresentationElement` has contents in its `fields` (see **section 2.3.3**) and `content` (see **section 2.3.2**) properties.

- `page` is the `PresentationPool` object representing the section page.
- `page.rootElement` is a `PresentationElement` object representing the root group or grid.

- **`page.rootElement.areas.header`** is a **`PresentationElement`** object representing the grid's **`header`** area.
- **`page.rootElement.areas.header.items[0]`** is a **`PresentationElement`** object representing the first summary in the **`header`** area.
- **`page.rootElement.areas.header.items[0].fields.title`** is the summary's title field.

The following JSP expression returns the **`title`** field of a summary in the first area of the **`header`** group of a section page. The section page is held in a variable called **`page`**:

You can determine what kind of a section page component a **`PresentationElement`** represents by testing its properties:

```
${page.rootElement.areas.header.items[0].fields.title}
```

In this expression:

`PresentationElement` has the properties described in the following sections.

2.3.1 areas

The areas in a **`PresentationElement`** that represents a group. If this property has zero length then the **`PresentationElement`** represents an area or a summary, not a group.

Type: **`java.util.Map<String,PresentationElement>`**

Example usage

To get all the areas:

```
${myPresentationElement.areas}
```

To get one particular area:

```
${myPresentationElement.areas.area}
```

where *area* is the key of the area you want.

2.3.2 content

The content item referenced by a **`PresentationElement`** that represents a summary.

Type: **`neo.xredsys.presentation.PresentationArticle`**

Example usage

```
${myPresentationElement.content}
```

2.3.3 fields

All the fields in a **`PresentationElement`** that represents a summary. If this property has zero length then the **`PresentationElement`** represents a group or an area, not a summary.

Field type	Java Object
BASIC	**`java.lang.String`**
NUMBER	a subclass of **`java.lang.Number`**
COMPLEX	**`java.lang.Map`**
BOOLEAN	**`java.lang.Boolean`**
ENUMERATION	a **`java.lang.String`** or **`java.lang.List`** of Strings
URI	**`java.net.URI`**
LINK	**`com.escenic.domain.Link`** (see **section 2.1**)
DATE	**`java.util.Date`**
SCHEDULE	**`neo.xredsys.presentation.PresentationSchedule`** (see **section 2.9**)

The actual type of the Java **`Object`**s representing the fields depends upon field type as follows:

Type: **`java.util.Map<String,Object>`**

Example usage

To get all the fields:

```
${myPresentationElement.fields}
```

To get one particular field:

```
${myPresentationElement.fields.field}
```

where *field* is the key of the field you want.

2.3.4 items

The summaries in a **`PresentationElement`** that represents an area. If this property has zero length then the **`PresentationElement`** represents a group or a summary, not an area.

Type: **`java.util.List<PresentationElement>`**

Example usage

To get all the items:

```
${myPresentationElement.items}
```

To get one particular item:

```
${myPresentationElement.items[index]}
```

where *index* is the index of the item you want.

2.3.5 keys

The names of the areas in a **`PresentationElement`** that represents a group. The names are returned in the order they are defined in the **`layout-group`** resource. If this property has zero length then the **`PresentationElement`** represents an area or a summary, not a group.

Type: **`java.util.List<String>`**

Example usage

To get all the names:

```
${myPresentationElement.keys}
```

To get one particular name:

```
${myPresentationElement.keys[index]}
```

where *index* is the index of the name you want.

2.3.6 legacyContent

This property is deprecated: use **`content`** (see **section 2.3.2**) instead.

The content item referenced by a **`PresentationElement`** that represents a summary. In this case, the returned item can be a legacy **`PresentationRelationImage`**, **`PresentationRelationLink`**, **`PresentationRelationMedia`** or **`PresentationRelationMedia`** object.

Type: **`neo.xredsys.presentation.PresentationRelation`**

Example usage

```
${myPresentationElement.legacyContent}
```

2.3.7 type

The type of the **`PresentationElement`**. For a **`PresentationElement`** representing a group or area, this will be the name of the group or area definition on which it is based. Groups and areas are defined in the **`layout-group`** resource file. For a **`PresentationElement`**representing a summary, no value is returned.

Type: **`java.lang.String`**

Example usage

```
${myPresentationElement.type}
```

2.4 PresentationPerson

Represents a person - the author of a content item, for example.

`PresentationPerson` has the properties described in the following sections.

2.4.1 address

The address of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.address}
```

2.4.2 birthDate

This property is deprecated: Use **`birthDateAsDate`** (see **section 2.4.3**) instead.

The birth date of this **`PresentationPerson`**. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.birthDate}
```

2.4.3 birthDateAsDate

The birth date of this **`PresentationPerson`**.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPerson.birthDateAsDate}
```

2.4.4 creationDate

The date when this **`PresentationPerson`** was created. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.creationDate}
```

2.4.5 creationDateAsDate

The date when this **`PresentationPerson`** was created.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPerson.creationDateAsDate}
```

2.4.6 description

The description of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.description}
```

2.4.7 email

This **`PresentationPerson`**'s email address.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.email}
```

2.4.8 firstName

The first name of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.firstName}
```

2.4.9 hashKey

The hash key of this object.

Type: **`neo.xredsys.api.IOHashKey`**

Example usage

```
${myPresentationObject.hashKey}
```

2.4.10 id

The id of the presentation object

Type: **`int`**

Example usage

```
${myPresentationObject.id}
```

2.4.11 lastModifiedDate

The date when this **`PresentationPerson`** was last modified. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.lastModifiedDate}
```

2.4.12 lastModifiedDateAsDate

The date when this **`PresentationPerson`** was last modified.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPerson.lastModifiedDateAsDate}
```

2.4.13 occupation

The occupation of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.occupation}
```

2.4.14 personName

The first and last name of this **`PresentationPerson`**, separated by a space.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.personName}
```

2.4.15 phoneMobile

This **`PresentationPerson`**'s mobile phone number.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.phoneMobile}
```

2.4.16 phonePrivate

This **`PresentationPerson`**'s home phone number.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.phonePrivate}
```

2.4.17 phoneWorkDirect

This **`PresentationPerson`**'s work phone number (direct dialing).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.phoneWorkDirect}
```

2.4.18 postNumber

The postal number of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.postNumber}
```

2.4.19 postPlace

The zip or postal code of this person. For use when zip/postal codes are split into a 'code' part and a 'place' part.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.postPlace}
```

2.4.20 profiles

All of this **`PresentationPerson`**'s profiles.

Type: **`java.util.Set`**

Example usage

```
${myPresentationPerson.profiles}
```

2.4.21 publicationId

The database ID of the **`Publication`** this **`PresentationPerson`** belongs to.

Type: **`int`**

Example usage

```
${myPresentationPerson.publicationId}
```

2.4.22 surName

The last name of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.surName}
```

2.5 PresentationPool

Represents a **section page** - the "front page" of a section of an Escenic publication.

The bean has only one property that you should use in new applications, **`rootElement`** (see **section 2.5.13**) . **All other properties are deprecated and only remain available for reasons of backwards compatibility.**

`PresentationPool` has the properties described in the following sections.

2.5.1 activateDate

The date when this pool was activated or going to be activated.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPool.activateDate}
```

2.5.2 articleCount

The article count of this pool.

Type: **`int`**

Example usage

```
${myPresentationPool.articleCount}
```

2.5.3 changedDate

The date when this pool was last updated.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPool.changedDate}
```

2.5.4 creationDate

The date this pool was created for the first time.

Type: `java.util.Date`

Example usage

```
${myPresentationPool.creationDate}
```

2.5.5 expireDate

The date when this pool is due to expire (or expired).

Type: `java.util.Date`

Example usage

```
${myPresentationPool.expireDate}
```

2.5.6 frontpageLayout

This property is deprecated: Replaced by `rootElement` (see **section 2.5.13**) . Do not use.

The frontpage layout of this pool

Type: `neo.xredsys.api.Layout`

Example usage

```
${myPresentationPool.frontpageLayout}
```

2.5.7 hashKey

The hash key of this object.

Type: `neo.xredsys.api.IOHashKey`

Example usage

```
${myPresentationObject.hashKey}
```

2.5.8 id

The id of the presentation object

Type: `int`

Example usage

```
${myPresentationObject.id}
```

2.5.9 poolId

The unique ID of the pool. This id is unique for all pools in the database.

Type: `int`

Example usage

```
${myPresentationPool.poolId}
```

2.5.10 poolTypeName

Return the name of the pooltype

Type: `java.lang.String`

Example usage

```
${myPresentationPool.poolTypeName}
```

2.5.11 publication

The publication that this pool belongs to.

Type: `neo.xredsys.api.Publication`

Example usage

```
${myPresentationPool.publication}
```

2.5.12 publicationId

The publication id of the publication that this pool belongs to.

Type: `int`

Example usage

```
${myPresentationPool.publicationId}
```

2.5.13 rootElement

The root group of this section page.

Type: `neo.xredsys.presentation.PresentationElement`

Example usage

```
${myPresentationPool.rootElement}
```

2.5.14 ruleTypeName

The ruletype name of this pool.

Type: `java.lang.String`

Example usage

```
${myPresentationPool.ruleTypeName}
```

2.5.15 section

The section that this pool belongs to.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPresentationPool.section}
```

2.5.16 sectionId

The section id of the section this pool belongs to.

Type: `int`

Example usage

```
${myPresentationPool.sectionId}
```

2.6 PresentationProfile

Represent a special content item used to contain a personal profile.

`PresentationProfile` has the properties described in the following sections.

2.6.1 activatedDate

The date this content item was activated. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.activatedDate}
```

2.6.2 activatedDateAsDate

The date this content item was activated.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.activatedDateAsDate}
```

2.6.3 agreementInfo

This content item's agreement information or `null` if the content item has no agreement.

Type: `neo.xredsys.api.AgreementInfo`

Example usage

```
${myPresentationArticle.agreementInfo}
```

2.6.4 agreementRequired

`true` if an agreement is required to view this content item, otherwise **`false`**.

Type: **`boolean`**

Example usage

```
${myPresentationArticle.agreementRequired}
```

2.6.5 allRelatedObjects

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the objects of any type (links, images, media objects and other content items) related to this content item.

Type: **`java.util.List<PresentationRelation>`**

Example usage

To get all the objects:

```
${myPresentationArticle.allRelatedObjects}
```

To get one particular object:

```
${myPresentationArticle.allRelatedObjects[index]}
```

where *index* is the index of the object you want.

2.6.6 allRelatedObjectsCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of objects of any type (links, images, media objects and other content items) related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.allRelatedObjectsCount}
```

2.6.7 articleId

The database id of this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.articleId}
```

2.6.8 articleTypeName

The type of this content item (as defined in the **`content-type`** resource).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.articleTypeName}
```

2.6.9 articles

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the content items related to this content item.

Type: **`java.util.List<PresentationRelationArticle>`**

Example usage

To get all the content items:

```
${myPresentationArticle.articles}
```

To get one particular content item:

```
${myPresentationArticle.articles[index]}
```

where *index* is the index of the content item you want.

2.6.10 articlesCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of content items related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.articlesCount}
```

2.6.11 author

The author of this content item. If the content item has more than one author then the first author in the list is returned.

Type: **`neo.xredsys.api.Person`**

Example usage

```
${myPresentationArticle.author}
```

2.6.12 authors

All the authors of this content item.

Type: `java.util.List<Person>`

Example usage

To get all the authors:

```
${myPresentationArticle.authors}
```

To get one particular author:

```
${myPresentationArticle.authors[index]}
```

where *index* is the index of the author you want.

2.6.13 authorsCount

The number of authors responsible for this content item.

Type: `int`

Example usage

```
${myPresentationArticle.authorsCount}
```

2.6.14 createdDate

The date this content item was created. Created in this context means "created in the Escenic system". For a content item that has been imported from an external source, therefore, it is the import date and not the original creation date. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.createdDate}
```

2.6.15 createdDateAsDate

The date this content item was created. Created in this context means "created in the Escenic system". For a content item that has been imported from an external source, therefore, it is the import date and not the original creation date.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.createdDateAsDate}
```

2.6.16 expireDate

The date on which this content item expired/will expire. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.expireDate}
```

2.6.17 expireDateAsDate

The date on which this content item expired/will expire.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.expireDateAsDate}
```

2.6.18 fieldNames

The names of all this content item's fields. The names are listed in the order they are defined in `content-types.xml`, and in upper case letters.

Type: `java.util.List<String>`

Example usage

To get all the names:

```
${myPresentationArticle.fieldNames}
```

To get one particular name:

```
${myPresentationArticle.fieldNames[index]}
```

where *index* is the index of the name you want.

2.6.19 fields

All the fields in this content item. The actual type of the Java `Objects` representing the fields depends upon field type as follows:

Field type	Java Object
BASIC	`java.lang.String`
NUMBER	a subclass of `java.lang.Number`
COMPLEX	`java.util.Map`
BOOLEAN	`java.lang.Boolean`
ENUMERATION	a `java.lang.String` or `java.util.List` of Strings
URI	`java.net.URI`

Field type	Java Object
LINK	`com.escenic.domain.Link` (see section 2.1)
DATE	`java.util.Date`
SCHEDULE	`neo.xredsys.presentation.PresentationSchedule` (see section 2.9)

Type: `java.util.Map<String,Object>`

Example usage

To get all the fields:

```
${myPresentationArticle.fields}
```

To get one particular field:

```
${myPresentationArticle.fields.field}
```

where *field* is the key of the field you want.

2.6.20 firstPublishedDate

The date on which this content item was first published. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.firstPublishedDate}
```

2.6.21 firstPublishedDateAsDate

The date on which this content item was first published.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.firstPublishedDateAsDate}
```

2.6.22 hashKey

The hash key of this object.

Type: `neo.xredsys.api.IOHashKey`

Example usage

```
${myPresentationObject.hashKey}
```

2.6.23 homeSection

The home section of this content item.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPresentationArticle.homeSection}
```

2.6.24 id

The id of the presentation object

Type: `int`

Example usage

```
${myPresentationObject.id}
```

2.6.25 lastModifiedDate

The date on which this content item was last modified. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.lastModifiedDate}
```

2.6.26 lastModifiedDateAsDate

The date on which this content item was last modified.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.lastModifiedDateAsDate}
```

2.6.27 live

`true` if this content item is published and active and its home section is published, otherwise `false`.

Type: `boolean`

Example usage

```
${myPresentationArticle.live}
```

2.6.28 owner

`true` if this content item is in its owning publication, **`false`** if it is a cross-published content item that is being accessed from a foreign publication.

Type: **`boolean`**

Example usage

```
${myPresentationArticle.owner}
```

2.6.29 ownerHomeSection

The owner home section of this content item. This is the real home section which may be part of another publication.

Type: **`neo.xredsys.api.Section`**

Example usage

```
${myPresentationArticle.ownerHomeSection}
```

2.6.30 ownerUrl

The absolute URL of this content item **in its owning home section**. This may be different from **`url`** (see **section 2.2.51**) , which returns the content item's URL in its current section. For a cross-published content item both the section and publication components of the URL will differ.

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.ownerUrl}
```

2.6.31 person

The person described in this **`PresentationProfile`**. In most case the returned object will be a **`PresentationUser`** (see **section 2.12**) .

Type: **`neo.xredsys.presentation.PresentationPerson`**

Example usage

```
${myPresentationProfile.person}
```

2.6.32 persons

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the persons related to this content item.

Type: **`java.util.Collection<PresentationPerson>`**

Example usage

To get all the persons:

```
${myPresentationArticle.persons}
```

To access each person in turn:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<c:forEach var="person" items="${myPresentationArticle.persons}">
  <!-- do something with each ${person} -->
</c:forEach>
```

2.6.33 personsCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of persons related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.personsCount}
```

2.6.34 profileName

The name of this **`PresentationProfile`**. The profile name is the title of the profile article.

Type: **`java.lang.String`**

Example usage

```
${myPresentationProfile.profileName}
```

2.6.35 publication

The publication this content item belongs to.

Type: **`neo.xredsys.api.Publication`**

Example usage

```
${myPresentationArticle.publication}
```

2.6.36 publicationId

The database id of the publication this content item belongs to.

Type: **`int`**

Example usage

```
${myPresentationArticle.publicationId}
```

2.6.37 publishedDate

The date on which this content item was last published. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.publishedDate}
```

2.6.38 publishedDateAsDate

The date on which this content item was last published.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.publishedDateAsDate}
```

2.6.39 relatedElements

A Map from relation type to a `PresentationElement` (see **section 2.3**) that holds the relations to this object.

Type: `java.util.Map`

Example usage

```
${myPresentationArticle.relatedElements}
```

2.6.40 relativeURI

The URL of this content item relative to its section. Usually this will be an URI object containing a file name with no path component, such as `article123.ece`.

Type: `java.net.URI`

Example usage

```
${myPresentationArticle.relativeURI}
```

2.6.41 relativeUrl

This property is deprecated: use the `relativeURI` (see **section 2.2.38**) instead.

The URL of this content item relative to its section. Usually this will be a file name with no path component, such as `article123.ece`.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.relativeUrl}
```

2.6.42 sections

This property is deprecated: Replaced by getSectionsList(). Should not be used.

The sections this content item belongs to.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPresentationArticle.sections}
```

2.6.43 sectionsList

The sections this content item belongs to.

Type: `java.util.List<Section>`

Example usage

To get all the sections:

```
${myPresentationArticle.sectionsList}
```

To get one particular section:

```
${myPresentationArticle.sectionsList[index]}
```

where *index* is the index of the section you want.

2.6.44 source

The name of the external source of an imported content item. A content item that has been imported from Associated Press, for example, might have an external source of "AP". `source` together with `sourceId` (see **section 2.2.43**) uniquely identifies the original source of the content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.source}
```

2.6.45 sourceId

The id of the external source of an imported content item. The form of the id depends upon the source: it might be some kind of numerical identifier or it might be URI of a news feed, for example. `sourceId` together with `source` (see **section 2.2.42**) uniquely identifies the original source of the content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.sourceId}
```

2.6.46 stateChangedDate

The date on which the state of this content item last changed. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.stateChangedDate}
```

2.6.47 stateChangedDateAsDate

The date on which the state of this content item last changed.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.stateChangedDateAsDate}
```

2.6.48 stateName

The state of this content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.stateName}
```

2.6.49 tags

A List of presentation tags.

Type: `java.util.List<PresentationTag>`

Example usage

To get all the tags:

```
${myPresentationArticle.tags}
```

To get one particular tag:

```
${myPresentationArticle.tags[index]}
```

where *index* is the index of the tag you want.

2.6.50 title

This property is deprecated: No replacement. Should not be used.

The title of this content item. The title of a content item is the content of the field defined as the title field in its content type definition in the content-types resource.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.title}
```

2.6.51 topic

The topic to which this content item belongs. If the content item belongs to more than one topic then the first topic in the list is returned. If the content item does not belong to a topic then `null`?? is returned. Note that only topics in the same publication as the content item are returned.

Type: `neo.xredsys.api.Topic`

Example usage

```
${myPresentationArticle.topic}
```

2.6.52 topicsList

All the topics this content item belongs to. If the content item does not belong to a topic then an empty list is returned. Note that only topics in the same publication as the content item are returned.

Type: `java.util.List<Topic>`

Example usage

To get all the topics:

```
${myPresentationArticle.topicsList}
```

To get one particular topic:

```
${myPresentationArticle.topicsList[index]}
```

where *index* is the index of the topic you want.

2.6.53 url

The absolute URL of this content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.url}
```

2.7 PresentationRelationArticle

This bean is deprecated: use `PresentationArticle.relatedElements` (see **section 2.2.37**) to obtain relations instead.

This is an article that is related to another article.

`PresentationRelationArticle` has the properties described in the following sections.

2.7.1 activatedDate

The date this content item was activated. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.activatedDate}
```

2.7.2 activatedDateAsDate

The date this content item was activated.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.activatedDateAsDate}
```

2.7.3 agreementInfo

This content item's agreement information or `null` if the content item has no agreement.

Type: `neo.xredsys.api.AgreementInfo`

Example usage

```
${myPresentationArticle.agreementInfo}
```

2.7.4 agreementRequired

`true` if an agreement is required to view this content item, otherwise `false`.

Type: `boolean`

Example usage

```
${myPresentationArticle.agreementRequired}
```

2.7.5 allRelatedObjects

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the objects of any type (links, images, media objects and other content items) related to this content item.

Type: **`java.util.List<PresentationRelation>`**

Example usage

To get all the objects:

```
${myPresentationArticle.allRelatedObjects}
```

To get one particular object:

```
${myPresentationArticle.allRelatedObjects[index]}
```

where *index* is the index of the object you want.

2.7.6 allRelatedObjectsCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of objects of any type (links, images, media objects and other content items) related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.allRelatedObjectsCount}
```

2.7.7 articleId

The database id of this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.articleId}
```

2.7.8 articleTypeName

The type of this content item (as defined in the **`content-type`** resource).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.articleTypeName}
```

2.7.9 articles

This property is deprecated: use `relatedElements` (see **section 2.2.37**) instead.

All the content items related to this content item.

Type: `java.util.List<PresentationRelationArticle>`

Example usage

To get all the content items:

```
${myPresentationArticle.articles}
```

To get one particular content item:

```
${myPresentationArticle.articles[index]}
```

where *index* is the index of the content item you want.

2.7.10 articlesCount

This property is deprecated: use `relatedElements` (see **section 2.2.37**) instead.

The number of content items related to this content item.

Type: `int`

Example usage

```
${myPresentationArticle.articlesCount}
```

2.7.11 author

The author of this content item. If the content item has more than one author then the first author in the list is returned.

Type: `neo.xredsys.api.Person`

Example usage

```
${myPresentationArticle.author}
```

2.7.12 authors

All the authors of this content item.

Type: `java.util.List<Person>`

Example usage

To get all the authors:

```
${myPresentationArticle.authors}
```

To get one particular author:

```
${myPresentationArticle.authors[index]}
```

where *index* is the index of the author you want.

2.7.13 authorsCount

The number of authors responsible for this content item.

Type: `int`

Example usage

```
${myPresentationArticle.authorsCount}
```

2.7.14 createdDate

The date this content item was created. Created in this context means "created in the Escenic system". For a content item that has been imported from an external source, therefore, it is the import date and not the original creation date. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.createdDate}
```

2.7.15 createdDateAsDate

The date this content item was created. Created in this context means "created in the Escenic system". For a content item that has been imported from an external source, therefore, it is the import date and not the original creation date.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.createdDateAsDate}
```

2.7.16 expireDate

The date on which this content item expired/will expire. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.expireDate}
```

2.7.17 expireDateAsDate

The date on which this content item expired/will expire.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.expireDateAsDate}
```

2.7.18 fieldNames

The names of all this content item's fields. The names are listed in the order they are defined in `content-types.xml`, and in upper case letters.

Type: `java.util.List<String>`

Example usage

To get all the names:

```
${myPresentationArticle.fieldNames}
```

To get one particular name:

```
${myPresentationArticle.fieldNames[index]}
```

where *index* is the index of the name you want.

2.7.19 fields

All the fields in this content item. The actual type of the Java `Objects` representing the fields depends upon field type as follows:

Field type	Java Object
BASIC	`java.lang.String`
NUMBER	a subclass of `java.lang.Number`
COMPLEX	`java.util.Map`
BOOLEAN	`java.lang.Boolean`
ENUMERATION	a `java.lang.String` or `java.util.List` of Strings
URI	`java.net.URI`
LINK	`com.escenic.domain.Link` (see section 2.1)
DATE	`java.util.Date`
SCHEDULE	`neo.xredsys.presentation.PresentationSchedule` (see section 2.9)

Type: `java.util.Map<String,Object>`

Example usage

To get all the fields:

```
${myPresentationArticle.fields}
```

To get one particular field:

```
${myPresentationArticle.fields.field}
```

where *field* is the key of the field you want.

2.7.20 firstPublishedDate

The date on which this content item was first published. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.firstPublishedDate}
```

2.7.21 firstPublishedDateAsDate

The date on which this content item was first published.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.firstPublishedDateAsDate}
```

2.7.22 hashKey

The hash key of this object.

Type: `neo.xredsys.api.IOHashKey`

Example usage

```
${myPresentationObject.hashKey}
```

2.7.23 homeSection

The home section of this content item.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPresentationArticle.homeSection}
```

2.7.24 id

The id of the presentation object

Type: `int`

Example usage

```
${myPresentationObject.id}
```

2.7.25 lastModifiedDate

The date on which this content item was last modified. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.lastModifiedDate}
```

2.7.26 lastModifiedDateAsDate

The date on which this content item was last modified.

Type: **`java.util.Date`**

Example usage

```
${myPresentationArticle.lastModifiedDateAsDate}
```

2.7.27 live

`true` if this content item is published and active and its home section is published, otherwise **`false`**.

Type: **`boolean`**

Example usage

```
${myPresentationArticle.live}
```

2.7.28 owner

`true` if this content item is in its owning publication, **`false`** if it is a cross-published content item that is being accessed from a foreign publication.

Type: **`boolean`**

Example usage

```
${myPresentationArticle.owner}
```

2.7.29 ownerHomeSection

The owner home section of this content item. This is the real home section which may be part of another publication.

Type: **`neo.xredsys.api.Section`**

Example usage

```
${myPresentationArticle.ownerHomeSection}
```

2.7.30 ownerUrl

The absolute URL of this content item **in its owning home section**. This may be different from **`url`** (see **section 2.2.51**) , which returns the content item's URL in its current section. For a cross-published content item both the section and publication components of the URL will differ.

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.ownerUrl}
```

2.7.31 persons

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

All the persons related to this content item.

Type: **`java.util.Collection<PresentationPerson>`**

Example usage

To get all the persons:

```
${myPresentationArticle.persons}
```

To access each person in turn:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<c:forEach var="person" items="${myPresentationArticle.persons}">
  <!-- do something with each ${person} -->
</c:forEach>
```

2.7.32 personsCount

This property is deprecated: use **`relatedElements`** (see **section 2.2.37**) instead.

The number of persons related to this content item.

Type: **`int`**

Example usage

```
${myPresentationArticle.personsCount}
```

2.7.33 placement

This property is deprecated: use **`PresentationArticle.relatedElements`** (see **section 2.2.37**) instead.

The name of the **`PresentationArticle`** field that holds the relation to this object.

Type: **`java.lang.String`**

Example usage

```
${myPresentationRelation.placement}
```

2.7.34 priority

This property is deprecated: use index in the list of relations instead, if needed.

The priority of the related object. Priority determines the order of a **`PresentationArticle`**'s related objects.

Type: **`int`**

Example usage

```
${myPresentationRelation.priority}
```

2.7.35 publication

The publication this content item belongs to.

Type: **`neo.xredsys.api.Publication`**

Example usage

```
${myPresentationArticle.publication}
```

2.7.36 publicationId

The database id of the publication this content item belongs to.

Type: **`int`**

Example usage

```
${myPresentationArticle.publicationId}
```

2.7.37 publishedDate

The date on which this content item was last published. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationArticle.publishedDate}
```

2.7.38 publishedDateAsDate

The date on which this content item was last published.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.publishedDateAsDate}
```

2.7.39 relatedElements

A Map from relation type to a **`PresentationElement`** (see **section 2.3**) that holds the relations to this object.

Type: `java.util.Map`

Example usage

```
${myPresentationArticle.relatedElements}
```

2.7.40 relativeURI

The URL of this content item relative to its section. Usually this will be an URI object containing a file name with no path component, such as **`article123.ece`**.

Type: `java.net.URI`

Example usage

```
${myPresentationArticle.relativeURI}
```

2.7.41 relativeUrl

This property is deprecated: use the **`relativeURI`** (see **section 2.2.38**) instead.

The URL of this content item relative to its section. Usually this will be a file name with no path component, such as **`article123.ece`**.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.relativeUrl}
```

2.7.42 sections

This property is deprecated: Replaced by getSectionsList(). Should not be used.

The sections this content item belongs to.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPresentationArticle.sections}
```

2.7.43 sectionsList

The sections this content item belongs to.

Type: `java.util.List<Section>`

Example usage

To get all the sections:

```
${myPresentationArticle.sectionsList}
```

To get one particular section:

```
${myPresentationArticle.sectionsList[index]}
```

where *index* is the index of the section you want.

2.7.44 source

The name of the external source of an imported content item. A content item that has been imported from Associated Press, for example, might have an external source of "AP". **`source`** together with **`sourceId`** (see **section 2.2.43**) uniquely identifies the original source of the content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.source}
```

2.7.45 sourceId

The id of the external source of an imported content item. The form of the id depends upon the source: it might be some kind of numerical identifier or it might be URI of a news feed, for example. **`sourceId`** together with **`source`** (see **section 2.2.42**) uniquely identifies the original source of the content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.sourceId}
```

2.7.46 state

This property is deprecated: should always return published.

The state of the related object.

Type: `java.lang.String`

Example usage

```
${myPresentationRelation.state}
```

2.7.47 stateChangedDate

The date on which the state of this content item last changed. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.stateChangedDate}
```

2.7.48 stateChangedDateAsDate

The date on which the state of this content item last changed.

Type: `java.util.Date`

Example usage

```
${myPresentationArticle.stateChangedDateAsDate}
```

2.7.49 stateName

The state of this content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.stateName}
```

2.7.50 tags

A List of presentation tags.

Type: `java.util.List<PresentationTag>`

Example usage

To get all the tags:

```
${myPresentationArticle.tags}
```

To get one particular tag:

```
${myPresentationArticle.tags[index]}
```

where *index* is the index of the tag you want.

2.7.51 title

This property is deprecated: No replacement. Should not be used.

The title of this content item. The title of a content item is the content of the field defined as the title field in its content type definition in the content-types resource.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.title}
```

2.7.52 topic

The topic to which this content item belongs. If the content item belongs to more than one topic then the first topic in the list is returned. If the content item does not belong to a topic then `null`?? is returned. Note that only topics in the same publication as the content item are returned.

Type: `neo.xredsys.api.Topic`

Example usage

```
${myPresentationArticle.topic}
```

2.7.53 topicsList

All the topics this content item belongs to. If the content item does not belong to a topic then an empty list is returned. Note that only topics in the same publication as the content item are returned.

Type: `java.util.List<Topic>`

Example usage

To get all the topics:

```
${myPresentationArticle.topicsList}
```

To get one particular topic:

```
${myPresentationArticle.topicsList[index]}
```

where *index* is the index of the topic you want.

2.7.54 url

The absolute URL of this content item.

Type: `java.lang.String`

Example usage

```
${myPresentationArticle.url}
```

2.8 PresentationRelationImage

This bean is deprecated: it is only available to ensure backwards compatibility with earlier versions of the Content Engine. In new applications you should use `PresentationElement` (see section 2.3) for all related content items.

Represents a relation between an article and an image.

`PresentationRelationImage` has the properties described in the following sections.

2.8.1 alignment

The alignment of the related image. Possible values are:

- left
- right
- center

If the `PresentationRelationImage` has no alignment set, then an empty string is returned.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.alignment}
```

2.8.2 alttext

The alternative text that can be displayed instead of the related image.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.alttext}
```

2.8.3 author

The name of the artist or designer to whom the related image is attributed.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.author}
```

2.8.4 caption

The caption of the related image. If the `PresentationRelationImage` has no caption, then the description of the image itself is returned.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.caption}
```

2.8.5 copyright

The copyright of the related image.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.copyright}
```

2.8.6 credit

The credit information for the related image.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.credit}
```

2.8.7 declaredCaption

The caption of the related image. If the `PresentationRelationImage` has no caption, then an empty string is returned.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.declaredCaption}
```

2.8.8 hashKey

The hash key of this object.

Type: `neo.xredsys.api.IOHashKey`

Example usage

```
${myPresentationObject.hashKey}
```

2.8.9 height

The height of the image version used in this relation.

Type: `int`

Example usage

```
${myPresentationRelationImage.height}
```

2.8.10 id

The id of the presentation object

Type: `int`

Example usage

```
${myPresentationObject.id}
```

2.8.11 name

The name of the related image.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.name}
```

2.8.12 path

The path of the related image.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.path}
```

2.8.13 photographer

The name of the photographer to whom the related image is attributed.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.photographer}
```

2.8.14 placement

This property is deprecated: use `PresentationArticle.relatedElements` (see **section 2.2.37**) instead.

The name of the `PresentationArticle` field that holds the relation to this object.

Type: `java.lang.String`

Example usage

```
${myPresentationRelation.placement}
```

2.8.15 preferredVersion

The name of the preferred version of the related image (one of an image's versions is designated the "preferred" version). The version actually in use in this relation may not be the preferred version. To find out which version is actually in use, use **`versionName`** (see **section 2.8.19**) .

Type: **`java.lang.String`**

Example usage

```
${myPresentationRelationImage.preferredVersion}
```

2.8.16 priority

This property is deprecated: use index in the list of relations instead, if needed.

The priority of the related object. Priority determines the order of a **`PresentationArticle`**'s related objects.

Type: **`int`**

Example usage

```
${myPresentationRelation.priority}
```

2.8.17 state

This property is deprecated: should always return published.

The state of the related object.

Type: **`java.lang.String`**

Example usage

```
${myPresentationRelation.state}
```

2.8.18 url

The absolute URL of the related image.

Type: **`java.lang.String`**

Example usage

```
${myPresentationRelationImage.url}
```

2.8.19 versionName

The version name of the related image. An image usually exists in several versions, each of which is a different size: this tells you which of these versions is in use here.

Type: `java.lang.String`

Example usage

```
${myPresentationRelationImage.versionName}
```

2.8.20 width

The width of the image version used in this relation.

Type: `int`

Example usage

```
${myPresentationRelationImage.width}
```

2.9 PresentationSchedule

Represents a **schedule**, consisting of a series of one or more events represented by `PresentationScheduleInstance` (see **section 2.10**) s. Each schedule instance is a date/time range representing the start date/time and end date/time of one of the events in the schedule.

`PresentationSchedule` has the properties described in the following sections.

2.9.1 instance

A `PresentationScheduleInstance` (see **section 2.10**) . If `PresentationSchedule.recurring` (see **section 2.9.3**) is true, then it will return the first instance defined by the schedule; otherwise it returns the only instance in the schedule.

Type: `neo.xredsys.presentation.PresentationScheduleInstance`

Example usage

```
${myPresentationSchedule.instance}
```

2.9.2 instances

All the `PresentationScheduleInstance` (see **section 2.10**) s defined by this schedule.

Type: `java.util.SortedSet`

Example usage

```
${myPresentationSchedule.instances}
```

2.9.3 recurring

Checks whether this is a recurring schedule that represents a sequence of events, or a non-recurring schedule that represents a single event.

Type: `boolean`

Example usage

```
${myPresentationSchedule.recurring}
```

2.10 PresentationScheduleInstance

Represents a **schedule instance**, one of the events defined in a `PresentationSchedule` (see **section 2.9**) . It contains the start and end times of an event.

`PresentationScheduleInstance` has the properties described in the following sections.

2.10.1 endDateTime

The end date and time of an event.

Type: `java.util.Date`

Example usage

```
${myPresentationScheduleInstance.endDateTime}
```

2.10.2 startDateTime

The start date and time of an event.

Type: `java.util.Date`

Example usage

```
${myPresentationScheduleInstance.startDateTime}
```

2.11 PresentationTag

Represents a **tag**. Tags are keywords that can be freely attached to content items, providing a simple means of grouping related content items.

`PresentationTag` has the properties described in the following sections.

2.11.1 name

The name of this presentation tag.

Type: `java.lang.String`

Example usage

```
${myPresentationTag.name}
```

2.12 PresentationUser

Represents a user of the Escenic Content Engine.

`PresentationUser` has the properties described in the following sections.

2.12.1 address

The address of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.address}
```

2.12.2 birthDate

This property is deprecated: Use **`birthDateAsDate`** (see **section 2.4.3**) instead.

The birth date of this **`PresentationPerson`**. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.birthDate}
```

2.12.3 birthDateAsDate

The birth date of this **`PresentationPerson`**.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPerson.birthDateAsDate}
```

2.12.4 creationDate

The date when this **`PresentationPerson`** was created. The date is returned as a string formated in accordance with the Java short date format (**`java.text.DateFormat.SHORT`**).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.creationDate}
```

2.12.5 creationDateAsDate

The date when this **`PresentationPerson`** was created.

Type: **`java.util.Date`**

Example usage

```
${myPresentationPerson.creationDateAsDate}
```

2.12.6 description

The description of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.description}
```

2.12.7 email

This **`PresentationPerson`**'s email address.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.email}
```

2.12.8 firstName

The first name of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.firstName}
```

2.12.9 hashKey

The hash key of this object.

Type: **`neo.xredsys.api.IOHashKey`**

Example usage

```
${myPresentationObject.hashKey}
```

2.12.10 id

The id of the presentation object

Type: **`int`**

Example usage

```
${myPresentationObject.id}
```

2.12.11 lastLoginDate

The date on which this user was last logged in. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationUser.lastLoginDate}
```

2.12.12 lastLoginDateAsDate

The date on which this user was last logged in.

Type: `java.util.Date`

Example usage

```
${myPresentationUser.lastLoginDateAsDate}
```

2.12.13 lastModifiedDate

The date when this `PresentationPerson` was last modified. The date is returned as a string formated in accordance with the Java short date format (`java.text.DateFormat.SHORT`).

Type: `java.lang.String`

Example usage

```
${myPresentationPerson.lastModifiedDate}
```

2.12.14 lastModifiedDateAsDate

The date when this `PresentationPerson` was last modified.

Type: `java.util.Date`

Example usage

```
${myPresentationPerson.lastModifiedDateAsDate}
```

2.12.15 occupation

The occupation of this `PresentationPerson`.

Type: `java.lang.String`

Example usage

```
${myPresentationPerson.occupation}
```

2.12.16 personName

The first and last name of this **`PresentationPerson`**, separated by a space.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.personName}
```

2.12.17 phoneMobile

This **`PresentationPerson`**'s mobile phone number.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.phoneMobile}
```

2.12.18 phonePrivate

This **`PresentationPerson`**'s home phone number.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.phonePrivate}
```

2.12.19 phoneWorkDirect

This **`PresentationPerson`**'s work phone number (direct dialing).

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.phoneWorkDirect}
```

2.12.20 postNumber

The postal number of this **`PresentationPerson`**.

Type: **`java.lang.String`**

Example usage

```
${myPresentationPerson.postNumber}
```

2.12.21 postPlace

The zip or postal code of this person. For use when zip/postal codes are split into a 'code' part and a 'place' part.

Type: `java.lang.String`

Example usage

```
${myPresentationPerson.postPlace}
```

2.12.22 profiles

All of this `PresentationPerson`'s profiles.

Type: `java.util.Set`

Example usage

```
${myPresentationPerson.profiles}
```

2.12.23 publicationId

The database ID of the `Publication` this `PresentationPerson` belongs to.

Type: `int`

Example usage

```
${myPresentationPerson.publicationId}
```

2.12.24 surName

The last name of this `PresentationPerson`.

Type: `java.lang.String`

Example usage

```
${myPresentationPerson.surName}
```

2.12.25 userName

The user name of this `PresentationUser`.

Type: `java.lang.String`

Example usage

```
${myPresentationUser.userName}
```

2.13 Publication

The publication object.

Represents an Escenic publication.

A publication is a web site that is managed by the Escenic Content Engine. It consists of a hierarchy of **sections** (represented by `Section` beans), which

in turn contain the **content items** (represented by `PresentationArticle` beans) that make up the web site.

`Publication` has the properties described in the following sections.

2.13.1 articleTypes

Retrieves all article types belonging to this publication.

Type: `java.util.Collection`

Example usage

```
${myPublication.articleTypes}
```

2.13.2 defaultSection

Retrieves the default section of this publication.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPublication.defaultSection}
```

2.13.3 name

The name of this publication.

Type: `java.lang.String`

Example usage

```
${myPublication.name}
```

2.13.4 rootSection

Retrieves the root section of this publication.

Type: `neo.xredsys.api.Section`

Example usage

```
${myPublication.rootSection}
```

2.14 Section

Represents one section of an Escenic publication.

Sections are containers for content items. The content items in a publication always belong to a section, and cannot be directly owned by the publication. A section can contain other sections as well as content items. The sections of a publication normally form a tree structure like the folders in a computer file system.

`Section` has the properties described in the following sections.

2.14.1 activeIndexPage

Retrieves the active index page from this section.

Type: `neo.xredsys.api.IndexPage`

Example usage

```
${mySection.activeIndexPage}
```

2.14.2 articleLayoutName

The name of the layout used for displaying this section's content items.

Type: `java.lang.String`

Example usage

```
${mySection.articleLayoutName}
```

2.14.3 creationDate

The time at which this section was created.

Type: `java.sql.Timestamp`

Example usage

```
${mySection.creationDate}
```

2.14.4 declaredParameterNameSet

Retrieves java.util.Set containing the names of all parameters in this section, disregarding any inherited parameters.

Type: `java.util.Set`

Example usage

```
${mySection.declaredParameterNameSet}
```

2.14.5 directoryName

The directory name of this section.

Type: `java.lang.String`

Example usage

```
${mySection.directoryName}
```

2.14.6 firstPublished

The time at which this section was published for the first time. If the section has been published more than once, then this will be different from the date returned by **publishDate** (see **section 2.14.19**) .

Type: **java.sql.Timestamp**

Example usage

```
${mySection.firstPublished}
```

2.14.7 inboxes

Retrieves a collection of all inboxes related to this section.

Type: **java.util.Collection**

Example usage

```
${mySection.inboxes}
```

2.14.8 indexPages

Retrieves a collection of all index pages related to this section.

Type: **java.util.Collection**

Example usage

```
${mySection.indexPages}
```

2.14.9 keyword

The keyword of this section.

Type: **java.lang.String**

Example usage

```
${mySection.keyword}
```

2.14.10 lastModified

The time at which this section was last modified. Note that this does **not** mean the last time the contents of the section or one of its section pages changed, but the last time the section itself was last modified. This usually means the last time a section administrator modified one of the section's parameters using Web Studio.

Type: **java.sql.Timestamp**

Example usage

```
${mySection.lastModified}
```

2.14.11 layoutName

The name of the layout used for this section.

Type: `java.lang.String`

Example usage

```
${mySection.layoutName}
```

2.14.12 lists

Retrieves a collection of all lists related to this section.

Type: `java.util.Collection`

Example usage

```
${mySection.lists}
```

2.14.13 name

The name of this section.

Type: `java.lang.String`

Example usage

```
${mySection.name}
```

2.14.14 parameterNameSet

Retrieves a `java.util.Set` containing the names of all parameters in this section, included inherited parameters from parent sections.

Type: `java.util.Set`

Example usage

```
${mySection.parameterNameSet}
```

2.14.15 parameters

Retrieves a `java.util.Map` containing all parameters in this section, included inherited parameters from parent sections.

Type: `java.util.Map`

Example usage

```
${mySection.parameters}
```

2.14.16 parent

The parent section of this section.

Type: `neo.xredsys.api.Section`

Example usage

```
${mySection.parent}
```

2.14.17 parentId

The id of the parent section of this section.

Type: `int`

Example usage

```
${mySection.parentId}
```

2.14.18 path

The local file system path of this section.

Type: `java.lang.String`

Example usage

```
${mySection.path}
```

2.14.19 publishDate

The time at which this section was published.

Type: `java.sql.Timestamp`

Example usage

```
${mySection.publishDate}
```

2.14.20 relativePath

The path of this section relative to the `Publication` root.

Type: `java.lang.String`

Example usage

```
${mySection.relativePath}
```

2.14.21 sectionUrl

Retrieves the url specifically configured for this section.

Type: `java.lang.String`

Example usage

```
${mySection.sectionUrl}
```

2.14.22 source

Retrieves the source of this section.

Type: `java.lang.String`

Example usage

```
${mySection.source}
```

2.14.23 sourceId

Retrieves the source id of this section.

Type: `java.lang.String`

Example usage

```
${mySection.sourceId}
```

2.14.24 subSections

The subsections of this section.

Type: `neo.xredsys.api.Section`

Example usage

```
${mySection.subSections}
```

2.14.25 uniqueName

The unique name of this section.

Type: `java.lang.String`

Example usage

```
${mySection.uniqueName}
```

2.14.26 url

The URL of this section.

Type: `java.lang.String`

Example usage

```
${mySection.url}
```

3 Utility Beans

The utility beans represent data structures created by the Content Engine for specific purposes: search results, tree structures and so on.

3.1 SearchResult

Represents a container for search results. It contains the search hits and meta-information about the search.

`SearchResult` has the properties described in the following sections.

3.1.1 metaKeys

The names of all the meta-elements in this search result. The elements in the returned set are of type **`String`**.

Type: **`java.util.Set`**

Example usage

```
${mySearchResult.metaKeys}
```

3.1.2 metaNames

This property is deprecated: use getMetaKeys instead.

The names of all the meta-elements in this search result.

Type: **`java.lang.String`**

Example usage

```
${mySearchResult.metaNames}
```

3.1.3 searchHits

All the search hits in this search result. The elements in the returned collection are of type **`SearchHit`**.

Type: **`java.util.Collection<SearchHit>`**

Example usage

To get all the hits:

```
${mySearchResult.searchHits}
```

To access each hit in turn:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<c:forEach var="hit" items="${mySearchResult.searchHits}">
```

```
  <!-- do something with each ${hit} -->
</c:forEach>
```

3.2 Toggle

Represents a toggle that can cycle through a series of values.

`Toggle` has the properties described in the following sections.

3.2.1 index

The index of the current value.

Type: **`int`**

Example usage

```
${myToggle.index}
```

3.2.2 value

The current value of the toggle.

Type: **`java.lang.String`**

Example usage

```
${myToggle.value}
```

3.3 View

Represents a selection of the nodes in a **`Tree`** (see **section 3.5**) .

`View` has the properties described in the following sections.

3.3.1 tree

The **`Tree`** (see **section 3.5**) from which this **`View`** was created.

Type: **`com.escenic.common.util.tree.Tree`**

Example usage

```
${myView.tree}
```

3.4 Relationships

Represents a variety of relationships describing a node's location in a **`Tree`** (see **section 3.5**) . The following definitions should help you understand the meaning of **`Relationships`** properties:

- **Current node** is the node being "measured".
- **Base node** is the node relative to which the current node is measured.

- **Level** is the shortest number of steps needed to get from a node to the tree root.
- **Relative level** is the difference between the level of the current node and the level of the base node. If the base node is on level 3 and the current node is on level 5, then the current node's relative level is 2. If the base node is on level 5 and the current node is on level 5, then the current node's relative level is -2.
- **Distance** is the number of steps between the current node and the base node. Distance can only be measured if the current node and base node are in the same line of descent (that is, if one is an ancestor/descendant of the other) or are identical. Distance is always either a positive number or 0.
- **Relative distance** is similar to **distance**, but is a negative number if the current node is an ancestor of the base node.
- The current and base nodes are **siblings** if they have the same parent node.

`Relationships` has the properties described in the following sections.

3.4.1 ancestor

`true` if the current node is an ancestor of the base node.

Type: `boolean`

Example usage

```
${myRelationships.ancestor}
```

3.4.2 base

`true` if the current node is the **base** node.

Type: `boolean`

Example usage

```
${myRelationships.base}
```

3.4.3 child

`true` if the current node is a child of the base node.

Type: `boolean`

Example usage

```
${myRelationships.child}
```

3.4.4 childInView

`true` if the current node has children in the view.

Type: `boolean`

Example usage

```
${myRelationships.childInView}
```

3.4.5 descendant

`true` if the current node is a descendant of the base node.

Type: **`boolean`**

Example usage

```
${myRelationships.descendant}
```

3.4.6 distance

The distance (that is, number of steps) between the current node and the base node.

Type: **`int`**

Example usage

```
${myRelationships.distance}
```

3.4.7 first

`true` if the current node has no preceding siblings.

Type: **`boolean`**

Example usage

```
${myRelationships.first}
```

3.4.8 firstChild

`true` if the current node has no preceding siblings.

Type: **`boolean`**

Example usage

```
${myRelationships.firstChild}
```

3.4.9 firstInView

`true` if the current node is the first node in the view.

Type: **`boolean`**

Example usage

```
${myRelationships.firstInView}
```

3.4.10 hasNoChildren

`true` if the **current** node has no children.

Type: `boolean`

Example usage

```
${myRelationships.hasNoChildren}
```

3.4.11 last

`true` if the current node has no following siblings.

Type: `boolean`

Example usage

```
${myRelationships.last}
```

3.4.12 lastChild

`true` if the current node has no following siblings.

Type: `boolean`

Example usage

```
${myRelationships.lastChild}
```

3.4.13 lastInView

`true` if the current node is the last node in the view.

Type: `boolean`

Example usage

```
${myRelationships.lastInView}
```

3.4.14 level

The level of the current node (the distance from the current node to the root-node).

Type: `int`

Example usage

```
${myRelationships.level}
```

3.4.15 nextInView

`true` if the current node's following sibling is in the view.

Type: `boolean`

Example usage

```
${myRelationships.nextInView}
```

3.4.16 onlyChild

`true` if the current node has no siblings (that is, if it is either the only child of its parent or the root node).

Type: **`boolean`**

Example usage

```
${myRelationships.onlyChild}
```

3.4.17 parent

`true` if the current node is the parent of the base node.

Type: **`boolean`**

Example usage

```
${myRelationships.parent}
```

3.4.18 previousInView

`true` if the current node's preceding sibling is in the view.

Type: **`boolean`**

Example usage

```
${myRelationships.previousInView}
```

3.4.19 relativeDistance

The relative distance between the current node and the base node.

The relative distance is only correct if one of **`ancestor`** (see **`ancestor`** (see **section 3.4.1**))), **`base`** (see **`base`** (see **section 3.4.2**))), **`child`** (see **`child`** (see **section 3.4.3**))), **`descendant`** (see **`descendant`** (see **section 3.4.5**))) or **`parent`** (see **`parent`** (see **section 3.4.17**))) returns **`true`**.

Type: **`int`**

Example usage

```
${myRelationships.relativeDistance}
```

3.4.20 relativeLevel

The level of the current node relative to to the base node. If the base node is regarded as level 0, then all levels above the base node are negative and all levels below it are positive.

Type: `int`

Example usage

```
${myRelationships.relativeLevel}
```

3.4.21 root

`true` if the current node is the root.

Type: `boolean`

Example usage

```
${myRelationships.root}
```

3.4.22 sibling

`true` if the current node is a sibling of the base node.

Type: `boolean`

Example usage

```
${myRelationships.sibling}
```

3.4.23 state

A description of the current node's relationship to the base node:

- "isCurrent" if **`base`** (see **`base`** (see **section 3.4.2**)) returns **`true`**.
- "isRoot" if **`root`** (see **`root`** (see **section 3.4.21**)) returns **`true`**.
- "isInPath" if both **`ancestor`** (see **`ancestor`** (see **section 3.4.1**)) and **`parent`** (see **`parent`** (see **section 3.4.17**)) return **`true`**.
- "isChild" if **`child`** (see **`child`** (see **section 3.4.3**)) returns **`true`**.
- "isSibling" if **`sibling`** (see **`sibling`** (see **section 3.4.22**)) returns **`true`**.
- "isParentInPath" if the **parent** of the current node is an ancestor of the base node.

Type: `java.lang.String`

Example usage

```
${myRelationships.state}
```

3.4.24 unrelated

`true` if the current node is not related to the base node. A node is related to base node if it is an ancestor, a descendant or a sibling.

Type: `boolean`

Example usage

```
${myRelationships.unrelated}
```

3.5 Tree

Represents a tree structure.

`Tree` has the properties described in the following sections.

3.5.1 recursiveView

A **`View`** object that includes the whole tree.

Type: **`com.escenic.common.util.tree.View`**

Example usage

```
${myTree.recursiveView}
```

3.5.2 root

The root of this tree. The root object is the root object because it (by definition) has no parent. If the domain object in fact has a parent in a different context, then it should be noted that in **this** context, the domain object should have no parent.

Type: **`java.lang.Object`**

Example usage

```
${myTree.root}
```

3.6 ResultPage

Represents a page of **`com.escenic.search.Result`** (see **section 3.7**) s.

`ResultPage` has the properties described in the following sections.

3.6.1 fromHits

The index of the first result on this page.

Type: **`int`**

Example usage

```
${myResultPage.fromHits}
```

3.6.2 next

`true` if there is a next page.

Type: **`boolean`**

Example usage

```
${myResultPage.next}
```

3.6.3 nextUrl

The URL of the next page. If the current page is the last page then the URL of the current page is returned.

Type: `java.lang.String`

Example usage

```
${myResultPage.nextUrl}
```

3.6.4 numberOfPages

The total number of pages.

Type: `int`

Example usage

```
${myResultPage.numberOfPages}
```

3.6.5 pageLength

The length of this page. Will always return the max value for this page. Use `size` (see section 3.6.9) to get the actual size of this page.

Type: `int`

Example usage

```
${myResultPage.pageLength}
```

3.6.6 pageNumber

The number of the current page.

Type: `int`

Example usage

```
${myResultPage.pageNumber}
```

3.6.7 previous

`true` if there is a previous page.

Type: `boolean`

Example usage

```
${myResultPage.previous}
```

3.6.8 previousUrl

The URL of the previous page. If the current page is the first page then the URL of the current page is returned.

Type: `java.lang.String`

Example usage

```
${myResultPage.previousUrl}
```

3.6.9 size

The number of `com.escenic.search.Result` (see **section 3.7**) s on the page.

Type: `int`

Example usage

```
${myResultPage.size}
```

3.6.10 toHits

The index of the last result on this page.

Type: `int`

Example usage

```
${myResultPage.toHits}
```

3.6.11 totalHits

The total number of results on the page.

Type: `int`

Example usage

```
${myResultPage.totalHits}
```

3.7 Result

Represents a result from a search operation.

`Result` has the properties described in the following sections.

3.7.1 description

The description of this `Result`.

Type: `java.lang.String`

Example usage

```
${myResult.description}
```

3.7.2 documentId

The document ID of this `Result`.

Type: `java.lang.String`

Example usage

```
${myResult.documentId}
```

3.7.3 fieldNames

The names of this this `Result`'s fields.

Type: `java.util.Set`

Example usage

```
${myResult.fieldNames}
```

3.7.4 publicationId

The ID of the `neo.xredsys.api.Publication` (see **section 2.13**) to which this `Result` belongs.

Type: `java.lang.String`

Example usage

```
${myResult.publicationId}
```

3.7.5 title

The title of this `Result`.

Type: `java.lang.String`

Example usage

```
${myResult.title}
```

3.7.6 url

The URL of the content page referenced by this `Result`.

Type: `java.lang.String`

Example usage

```
${myResult.url}
```