



Service-Interface-API-spec1.0.pdf

Table of Contents

1 Introduction	4
1.1 Namespaces	4
1.2 Security	4
1.3 Errors	4
2 Services	6
2.1 Service requests	6
2.1.1 findServices	6
2.1.2 addServices	6
2.1.3 updateService	6
2.1.4 deleteServices	7
2.2 Descriptors on services	7
2.2.1 findServiceDescriptors	7
2.2.2 addServiceDescriptors	7
2.2.3 updateServiceDescriptors	8
2.2.4 deleteServiceDescriptors	8
3 Components	9
3.1 Service components	9
3.1.1 findComponents	9
3.1.2 addComponents	9
3.1.3 updateComponent	10
3.1.4 deleteComponents	10
3.2 Component property declarations	10
3.2.1 findPropertyDeclarations	11
3.2.2 addPropertyDeclarations	11
3.2.3 updatePropertyDeclarations	11
3.2.4 deletePropertyDeclarations	12
3.3 Descriptors on components	12
3.3.1 findComponentDescriptors	12
3.3.2 addComponentDescriptors	12
3.3.3 updateComponentDescriptors	13
3.3.4 deleteServiceDescriptors	13
4 Content	14
4.1 Content and subclasses	14
4.2 Content items	14
4.2.1 findContent	14
4.2.2 addContent	14
4.2.3 updateContent	15
4.2.4 deleteContent	15
4.3 Content properties	15
4.3.1 findContentProperties	16
4.3.2 addContentProperties	16

4.3.3 updateContentProperties	16
4.3.4 deleteContentProperties	16
4.4 Descriptors on content	17
4.4.1 findContentDescriptors	17
4.4.2 addContentDescriptors	17
4.4.3 updateContentDescriptors	18
4.4.4 deleteContentDescriptors	18
5 Branding and skins	19
5.1 Generic screen areas	19
5.1.1 findScreenAreas	19
5.1.2 addScreenAreas	19
5.1.3 updateScreenAreas	20
5.1.4 deleteScreenAreas	20
6 Administration	21
6.1 findTransactions	21
7 Complex Types	22
7.1 Area	22
7.2 Component	22
7.2.1 PluginComponent	22
7.2.2 TVChannelPackage	23
7.2.3 VODComponent	23
7.3 Content	23
7.3.1 Plugin	24
7.3.2 TVChannel	24
7.3.3 Movie	24
7.4 Credentials	24
7.5 Descriptor	25
7.6 Dimension	25
7.7 Image	25
7.8 Payment	25
7.9 Playlist	25
7.10 Position	25
7.11 Price	26
7.12 Property	26
7.13 PropertyDeclaration	26
7.14 Service	26
7.15 Stream	26

7.16 Style.....	26
7.17 Transaction.....	27

1 Introduction

Gives an introduction to the service API specification. Defines namespaces used, security model and error handling.

1.1 Namespaces

An XML *namespace* provide a way to distinguish deterministically between between XML elements that that have the same *local name* but are, in fact from different vocabularies. A namespace declaration links a namespace together with a shorter namespace prefix. A prefix together with a local name is called a *qualified* name and is globally unique.

Example

`addService` accepts an element with the local name `Credentials` in the <http://www.acreo.se/schema/serviceAPI/2007r1> namespace prefixed as `sapi`. The qualified name of the element is `sapi:Credentials`.

```
<addService xmlns:sapi="http://www.acreo.se/schema/serviceAPI/2007r1">  
  <sapi:Credentials>  
    ...
```

Service API specific elements in this specification are all part of the <http://www.acreo.se/schema/serviceAPI/2007r1> namespace (prefixed as `sapi`). Simple type elements are part of the <http://www.w3.org/2001/XMLSchema> namespace (prefixed as `xsd`).

1.1 Security

The backend should be protected at the server level using basic authentication. Each request must contain an authentication block (`sapi:Credentials`) to let the middleware decide who is making the request and if he/she have the correct permissions to perform the action.

1.1 Errors

If an operation invocation fails, a SOAP fault is returned to the client with a fault code, a string and, in some cases, a detailed description.

Example

```
...
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Client</faultcode>
    <faultstring>
      Illegal service
    </faultstring>
    <detail>
      The specified service does not exist in the
      database.
    </detail>
  </soap:Fault>
</soap:Body>
...
```

If a problem arises with the request on the server, the fault code *Server* is returned to the client together with an appropriate message.

Example

```
...
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Server</faultcode>
    <faultstring>
      Database connection failed
    </faultstring>
  </soap:Fault>
</soap:Body>
...
```

Operation specific faults are up to the implementing middleware to define.

1 Services

Specification of requests/responses concerning services. A Service is a bundle of content end users have access to.

1.1 Service requests

Requests to list or change services in the middleware.

1.1.1 *findServices*

This operation finds services matching the search criteria. To avoid generating too large soap messages the `ignoreImageData` flag can be set to true to exclude binary image data from the response.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	0..1	If set, return only the service with this id
ignoreImageData	xsd:boolean	0..1	Set to true to exclude image binary data from response, false or null otherwise
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
service	sapi:Service	0..*	Zero or more matching Service objects

1.1.1 *addServices*

This operation adds one or more services.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
service	sapi:Service	1..*	One or more Service objects

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of services successfully added

1.1.1 *updateService*

This operation updates a service. If a Service object is present in the request all service properties (including id) will be updated. Descriptors update existing ones of the same language, content items are added or replaced depending on pre-existing content with the same id.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1	id of service to update
service	sapi:Service	0..1	Service object

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of services successfully updated

1.1.1 deleteServices

Deletes one or more services. NOTE: Components and other underlying objects bundled in the service will also be deleted.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1..*	id of services to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of services successfully deleted

1.1 Descriptors on services

Requests to make changes on descriptors contained in services.

1.1.1 findServiceDescriptors

Returns all or one descriptor depending on if the language parameter is set.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
serviceId	xsd:string	0..1	Service id
language	xsd:string	0..*	Zero or more language codes of descriptors to retrieve
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
descriptor	sapi:Descriptor	0..*	Zero or more matching Descriptor objects

1.1.1 addServiceDescriptors

This operation adds one or more descriptors to a service.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object

serviceId	xsd:string	1	Service id
descriptor	sapi:Descriptor	1..*	Descriptor objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully added

1.1.1 *updateServiceDescriptors*

This operation updates an existing descriptor of a service.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1	Service id
descriptor	sapi:Descriptor	1..*	Descriptor objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully updated

1.1.1 *deleteServiceDescriptors*

This operation deletes descriptors from a generic service.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1	Service id
language	xsd:string	0..*	Zero or more language codes of descriptors to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully deleted

1 Components

Specification of requests/responses concerning components. A component defines the behavior of the content it contains.

1.1 Service components

Requests to add, delete, update or list components of a service. All requests accept either generic `sapi:Component` objects or any of the specialized components: `sapi:PluginComponent`, `sapi:TVChannelPackage` and `sapi:VODComponent`.

1.1.1 *findComponents*

This operation finds one or more `sapi:Component` objects matching the search criteria. To avoid generating too large soap messages the `ignoreImageData` flag can be set to true to exclude binary image data from the response.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
type	xsd:string	0..1	If not null only components of the specified type is returned
serviceId	xsd:string	0..1	If not null only components belonging to the service with that id is returned
componentId	xsd:string	0..1	If not null only the component with the id specified is returned
ignoreImageData	xsd:boolean	0..1	Set to true to exclude image binary data from response, false or null otherwise
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
component plugin tvChannel vod	sapi:Component	0..*	Zero or more matching Component objects

1.1.1 *addComponents*

This operation adds one or more components to a service.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1	Service id
component plugin tvChannel vod	sapi:Component	1..*	Component objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of components successfully added

1.1.1 updateComponent

This operation updates a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
component plugin tvChannel vod	sapi:Component	1..*	Component objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of components successfully updated

1.1.1 deleteComponents

This operation deletes components from a service. Also deletes associated content data such as images.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1	Id of the Service to delete components from
componentId	xsd:string	0..*	Zero or more ids of components to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of service components successfully deleted

1.1 Component property declarations

Requests to make changes to property declarations for a component.

1.1.1 findPropertyDeclarations

Retrieves the property declarations of a component.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
componentId	xsd:string	1	Component id
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
-----------	------	---	-------------

propertyDeclaration	sapi:PropertyDeclaration	0..*	Zero or more matching PropertyDeclaration objects
---------------------	--------------------------	------	---

1.1.1 addPropertyDeclarations

This operation adds property declarations to a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
propertyDeclaration	sapi:PropertyDeclaration	1..*	PropertyDeclaration objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of properties successfully added

1.1.1 updatePropertyDeclarations

This operation updates existing property declarations of a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
propertyDeclaration	sapi:PropertyDeclaration	1..*	PropertyDeclaration objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of PropertyDeclaration objects successfully updated

1.1.1 deletePropertyDeclarations

This operation deletes property declarations from a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
name	xsd:string	0..*	Zero or more names of PropertyDeclaration objects to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of PropertyDeclaration objects successfully deleted

1.1 Descriptors on components

Requests to make changes on descriptors of components.

1.1.1 *findComponentDescriptors*

Returns all or one descriptor depending on if the language parameter is set.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
componentId	xsd:string	0..1	Component id
language	xsd:string	0..*	Zero or more language codes of descriptors to retrieve
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
descriptor	sapi:Descriptor	0..*	Zero or more matching Descriptor objects

1.1.1 *addComponentDescriptors*

This operation adds one or more descriptors to a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
descriptor	sapi:Descriptor	1..*	Descriptor objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully added

1.1.1 *updateComponentDescriptors*

This operation updates an existing descriptor of a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
descriptor	sapi:Descriptor	1..*	Descriptor objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully updated

1.1.1 *deleteServiceDescriptors*

This operation deletes descriptors from a generic service.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id

language	xsd:string	0..*	Zero or more language codes of descriptors to delete
----------	------------	------	--

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully deleted

1 Content

Specification of requests/responses concerning content. A content item holds data needed to present and purchase it through the middleware.

1.1 Content and subclasses

When requesting or sending `sapi:Content` objects, subclasses of the `sapi:Content` class must also be accepted. All content objects must have unique IDs, whether they are of the `sapi:Content`, `sapi:Movie`, `sapi:TVChannel` or `sapi:Plugin` class.

1.1 Content items

Requests to add, delete, update or list content items in a component. All requests accept either generic `sapi:Content` objects or any of the specialized: `sapi:Plugin`, `sapi:TVChannel` and `sapi:Movie`.

1.1.1 *findContent*

This operation finds one or more `sapi:Content` objects matching the search criteria.

To avoid generating too large soap messages the `ignoreImageData` flag can be set to true to exclude binary image data from the response.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
componentId	xsd:string	0..1	If not null only content belonging to the component with that id is returned
contentId	xsd:string	0..1	If not null only the Content with the id specified is returned
ignoreImageData	xsd:boolean	0..1	Set to true to exclude image binary data from response, false or null otherwise
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
content	sapi:Content	0..*	Zero or more matching Content objects

1.1.1 *addContent*

This operation adds one or more content items to a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object

componentId	xsd:string	1	Component id
content plug in movie tvChannel	sapi:Content	1..*	Content objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of content items successfully added

1.1.1 updateContent

This operation updates one or more content items in a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
content	sapi:Content	1..*	Content objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of content items successfully updated

1.1.1 deleteContent

This operation deletes one or more content items from a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Component id
contentId	xsd:string	1..*	One or more ids of Content to delete from a Component

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of content items successfully deleted

1.1

1.2 Content properties

Requests to list or make changes to content properties. A property holds the value of a corresponding property declaration.

1.1.1 findContentProperties

Retrieves the properties of a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
contentId	xsd:string	1	Content id

offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
property	sapi:Property	0..*	Zero or more matching Property objects

1.1.1 addContentProperties

This operation adds properties to a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
contentId	xsd:string	1	Content id
property	sapi:Property	1..*	Property objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of properties successfully added

1.1.1 updateContentProperties

This operation updates existing properties of a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
contentId	xsd:string	1	Content id
property	sapi:Property	1..*	Property objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of properties successfully updated

1.1.1 deleteContentProperties

This operation deletes properties from a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
contentId	xsd:string	1	Content id
name	xsd:string	0..*	Zero or more names of Property objects to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of Property objects successfully deleted

1.1 Descriptors on content

Requests to make changes on descriptors of content items.

1.1.1 *findContentDescriptors*

Returns all or one descriptor depending on if the language parameter is set.

Input Parameters

parameter	type	#	description
credentials	sapi:credentials	1	Authentication object
contentId	xsd:string	1	Content id
language	xsd:string	0..*	Zero or more language codes of descriptors to retrieve
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
descriptor	sapi:Descriptor	0..*	Zero or more matching Descriptor objects

1.1.1 *addContentDescriptors*

This operation adds one or more descriptors to a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
contentId	xsd:string	1	Content id
descriptor	sapi:Descriptor	1..*	Descriptor objects to add

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully added

1.1.1 *updateContentDescriptors*

This operation updates an existing descriptor of a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
contentId	xsd:string	1	Content id
descriptor	sapi:Descriptor	1..*	Descriptor objects to update

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully updated

1.1.1 *deleteContentDescriptors*

This operation deletes descriptors from a content item.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
contentId	xsd:string	1	Content id
language	xsd:string	0..*	Zero or more language codes of descriptors to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of descriptors successfully deleted

1 Branding and skins

Specification of requests/responses for customizing screen areas used by services and components. Customizable areas enable service providers to apply their own branding and profiling.

1.1 Generic screen areas

1.1.1 *findScreenAreas*

This operation returns screen areas belonging to a service, having a specific id or type.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	0..1	id of the Component using the Area
areaId	xsd:string	0..1	Screen area id
type	xsd:string	0..1	Screen area type. One of the following: static list preview popup
offset	xsd:int	0..1	Result offset
limit	xsd:int	0..1	Maximum number of results

Output Parameters

parameter	type	#	description
area	sapi:Area	0..*	Zero or more Area objects

1.1.1 *addScreenAreas*

This operation adds screen areas to a component.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
componentId	xsd:string	1	Id of the Component using the area
area	sapi:Area	1..*	Screen Area object

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of screens successfully added

1.1.1 *updateScreenAreas*

This operation updates screen areas.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
area	sapi:Area	1..*	Screen Area object

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of screens successfully updated

1.1.1 deleteScreenAreas

This operation deletes screen areas.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
areaId	xsd:string	1..*	Ids of screen areas to delete

Output Parameters

parameter	type	#	description
return	xsd:int	1	Number of screens successfully deleted

1 Administration

Specification of requests/responses to handle administrative tasks such as viewing transactions.

1.1 findTransactions

This operation returns a number of transactions.

Input Parameters

parameter	type	#	description
credentials	sapi:Credentials	1	Authentication object
serviceId	xsd:string	1	If not null return only transactions generated by the <i>Service</i> with this id
customerRef	xsd:string	0..1	Customer reference, if null transactions for all customers are returned. The value is either a PIN code, MAC or IP address
start	xsd:dateTime	0..1	Start date/time to look for transactions
end	xsd:dateTime	0..1	End date/time to look for transactions

Output Parameters

parameter	type	#	description
transaction	sapi:Transaction	0..*	Zero or more Transaction objects

1 Complex Types

Descriptions of complex types used by the service API.

Service API specific elements in this specification are all part of the <http://www.acreo.se/schema/serviceAPI/2007r1> namespace (prefixed as `sapi`). Simple type elements are part of the <http://www.w3.org/2001/XMLSchema> namespace (prefixed as `xsd`).

1.1 Area

element	type	#	description
areaId	xsd:string	1	Screen area id
type	xsd:string	1	Area type. One of the following: static list preview popup
dimension	sapi:Dimension	1	Dimension object
position	sapi:Position	1	Position object
background	sapi:Style	0..1	Style object for the screen area background
content	xsd:anyURI	0..*	Source of the area content (e.g. a text-file, html, xml or server-side parsing script)

1.1 Component

A generic component.

element	type	#	description
id	xsd:string	1	Component id
type	xsd:string	1	Component type. One of the following: generic vod plugin tv
propertyDeclaration	sapi:PropertyDeclaration	0..*	Component property declarations
descriptor	sapi:Descriptor	1..*	Descriptor objects describing the component
payment	sapi:Payment	1	Payment object
content	sapi:Content	0..*	Content objects populating the component

1.1.1 PluginComponent

A `PluginComponent` can only contain content of the type `sapi:Plugin`.

element	type	#	description
id	xsd:string	1	Plugin id
descriptor	sapi:Descriptor	1..*	Descriptor objects describing the plugin
payment	sapi:Payment	1	Payment object
plugin	sapi:Plugin	0..*	Plugin objects populating the component
propertyDeclaration	sapi:PropertyDeclaration	0..*	PluginComponent property declarations

1.1.1 TVChannelPackage

A `TVChannelPackage` can only contain content of the type `sapi:TVChannel`.

element	type	#	description
id	xsd:string	1	Component id
descriptor	sapi:Descriptor	1..*	Descriptor objects describing the component
payment	sapi:Payment	1	Payment object
tvChannel	sapi:TVChannel	0..*	TVChannel objects populating the component
propertyDeclaration	sapi:PropertyDeclaration	0..*	TVChannelPackage property declarations

1.1.1 VODComponent

A VODComponent can only contain content of the type sapi:Movie.

element	type	#	description
id	xsd:string	1	VOD component id
descriptor	sapi:Descriptor	1..*	Descriptor objects describing the vod component
payment	sapi:Payment	1	Payment object
movie	sapi:Movie	0..*	Movie objects populating the component
propertyDeclaration	sapi:PropertyDeclaration	0..*	VODComponent property declarations

1.1 Content

element	type	#	description
id	xsd:string	1	Content id
sortOrder	xsd:int	1	Sort order in list
price	sapi:Price	0..1	Content price
descriptor	sapi:Descriptor	1..*	Content descriptor
image	sapi:Image	0..1	image (e.g. icon to show in list)
property	sapi:Property	0..*	Content properties

1.1.1 Plugin

element	type	#	description
id	xsd:string	1	Plugin id
order	xsd:int	1	Sort order in list
price	sapi:Price	0..1	Plugin price
descriptor	sapi:Descriptor	1..*	Plugin descriptor
image	sapi:Image	0..1	image (e.g. icon to show in list)
src	xsd:anyURI	1	Plugin source location
property	sapi:Property	0..*	Plugin properties

1.1.1 TVChannel

element	type	#	description
id	xsd:string	1	TV channel id
order	xsd:int	1	Sort order in list
descriptor	sapi:Descriptor	1..*	TV channel descriptor
image	sapi:Image	0..1	image (e.g. icon to show in list)
src	xsd:anyURI	1	TV channel multicast stream source URI
lcn	xsd:int	0..1	Logical channel number of the TV channel
property	sapi:Property	0..*	tv channel properties

1.1.1 Movie

element	type	#	description
id	xsd:string	1	Content id
order	xsd:int	1	Sort order in list
price	sapi:Price	0..1	Content price
descriptor	sapi:Descriptor	1..*	Content descriptor
image	sapi:Image	0..1	image (e.g. icon to show in list)
src	sapi:Playlist	1	Ordered list of movie stream sources
trailerSrc	sapi:Playlist	0..1	Ordered list of trailer stream sources

length	xsd:int	0..1	Total playtime in minutes
rentalTime	xsd:int	0..1	Number of hours the content is available to the end user after purchase
property	sapi:Property	0..*	Movie properties

1.1 Credentials

element	type	#	description
user	xsd:string	1	User name
password	xsd:string	1	User password (encrypted)

1.1 Descriptor

element	type	#	description
language	xsd:string	1	Language code (ISO 639-2)
caption	xsd:string	1	Caption
description	xsd:string	0..1	Description
extInfo	xsd:string	0..1	Extended information

1.1 Dimension

element	type	#	description
height	xsd:int	1	height in pixels
width	xsd:int	1	width in pixels

1.1 Image

element	type	#	description
id	xsd:string	1	Image id
name	xsd:string	0..1	Image name
format	xsd:string	1	Image format. One of the following: JPEG PNG GIF
src	xsd:anyURI	0..1	Image data source location
data	xsd:base64Binary	0..1	Base64encoded image data
attachment	xsd:string	0..1	Pointer for soap attachment containing data
dimension	sapi:Dimension	0..1	Dimension object

1.1 Payment

Payment Properties are all of the type string.

element	type	#	description
authMethod	xsd:string	1	One of the following: PIN MAC IP
priceModel	xsd:string	1	One of the following: free ondemand subscription custom
property	sapi:Property	0..1	Definitions of subscription periods, ladder intervals, time periods etc.

1.1 Playlist

element	type	#	description
id	xsd:string	1	Playlist id
stream	sapi:Stream	1..*	One or more Stream objects

1.1 Position

Positions are relative the top-left corner (x=0, y=0) as reference.

element	type	#	description
left	xsd:int	1	Horizontal position in pixels
top	xsd:int	1	Vertical position in pixels

1.1 Price

element	type	#	description
amount	xsd:string	1	Amount (e.g. "25.00")
currency	xsd:string	1	Currency (e.g. "SEK")

1.1 Property

element	type	#	description
name	xsd:string	1	Name of the property
value	xsd:string	0..1	Property value, might be null

1.1 PropertyDeclaration

element	type	#	description
name	xsd:string	1	Name of the property
type	xsd:string	1	Property type. One of the following: number string boolean list date time datetime hours minutes
filter	xsd:boolean	0..1	true if the property can be used as a filter, false or null otherwise

1.1 Service

element	type	#	description
id	xsd:string	1	Service id
payment	sapi:Payment	1	Payment object
descriptor	sapi:Descriptor	1..*	Descriptor objects describing the service
component	sapi:Component	0..*	Component objects in the service

1.1 Stream

element	type	#	description
id	xsd:string	1	Stream id
sortOrder	xsd:int	0..1	Sort order in list
src	xsd:anyURI	1	Stream source location

1.1 Style

element	type	#	description
color	xsd:string	0..1	String representing a color (e.g. #0000FF, green)
image	sapi:Image	0..1	Image object
position	sapi:Position	0..1	Position object

1.1 Transaction

element	type	#	description
serviceId	xsd:string	1	ID of the service that generated the transaction
componentId	xsd:string	1	ID of the component that generated the transaction

SeqNr	xsd:int	1	Transaction sequence number
time	xsd:dateTime	1	Transaction timestamp
customerRef	xsd:string	1	Customer reference (e.g. Customer PIN)
price	sapi:Price	0..1	Transaction price

1.1

2 Dictionary

Definitions used throughout this document.

Services

A *service* is a collection of service components provided by a content or service provider. The service object holds information about the service and a list of components that are part of it.

Components

A service *component* or simply component is a part of a service providing some specialized content (e.g. video on demand, games or TV channels). A component holds information about the very component as well as information how end users pay for using the content in the component.

Components are either generic or belonging to one of the specialized component classes: `VODComponent`, `TVChannelPackage` or `Plugin`. Each of the specialized component classes define a set of pre-defined properties (that can be extended if the implementer so wishes, see *properties* below).

Content items

Content items hold information needed to present and use it in the middleware. Content items instantiates the properties declared by the component they are associated with (with `sapi:Property`).

Descriptors

Services, components and content all have descriptors used to provide the middleware with information about them. An object can have several descriptors as long as the descriptor's language properties are unique.

Properties

Properties define certain behavior/look/usage of objects when interpreted by the middleware. A special usage of properties are filtering out content for an area from a larger selection. For this purpose there is a filter element that, when set to true, defines a filter of certain type.

Components define valid properties for their content items with `sapi:PropertyDeclaration` elements.

Content items instantiates the properties declared by the component they are added to with `sapi:Property`.

Images

Images are represented by the `sapi:Image` object. The binary data for the image might be transferred either as a base64encoded string or as an attachment (using MTOM, SwA or other technologies). Alternatively image data may be referenced as a URI using `xsd:anyURI`.