



# Curso Multimedia Home Platform 1.1.2

## System Information II. JavaTV

Accediendo a su contenido con JavaTV

## Curso Multimedia Home Platform 1.1.2

Copyright 2008 © Enrique Pérez Gil

Licensed under the ***Creative Commons Attribution-Non-Commercial-No Derivative Works 3.0 Unported License***. You may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>

This is a human-readable summary of the License applied:

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

**You are free to Share**, to copy, distribute and transmit the work **Under the following conditions:**

- **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial.** You may not use this work for commercial purposes.
- **No Derivative Works.** You may not alter, transform, or build upon this work.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.

## JavaTV SI API. Introducción

- Ofrece un API orientado a Objetos y Tareas más que a las Tablas de DVB-SI.
- En definitiva es una capa “lógica” de lo que se compone un sistema de Broadcasting MPEG
- Es compatible con ATSC y OCAP
- Si trabajas en un entorno puro MHP no es necesario.
- Las clases residen en
  - **javax.tv.service**
  - **javax.tv.service.guide**
  - **javax.tv.service.navigation**
  - **javax.tv.service.transport**

## JavaTV SI API

- Las Clases que me ofrecen la información del API son las siguientes y todas salvo Service heredan de **javax.tv.service.SIElement**
  - javax.tv.service.**Service** no hereda de SIElement.
  - javax.tv.service.guide.**ProgramEvent**: Event(Show)
  - javax.tv.service.guide.**ProgramEventDescription**: descripción de un show
  - javax.tv.service.navigation.**ServiceComponent**: Streams de un Service.
  - javax.tv.service.navigation.**ServiceDetails**: Información de un Service
  - javax.tv.service.navigation.**ServiceDescription**: desc. de un Service.
  - javax.tv.service.transport.**Bouquet**
  - javax.tv.service.transport.**Network**
  - javax.tv.service.transport.**TransportStream**

## Punto de partida. La “BBDD”

- Seguimos un esquema paralelo a DVB SI.
- ¿ De donde parto ? En DVB SI de SIDatabase, aquí de:

**javax.tv.service.SIManager**

Para obtener una instancia la creo:

**javax.tv.service.SIManager.createInstance()**

### OJO:

- Incluye toda la información de todos los Tuners que haya en el sistema, no como en DVB SI
- Cada instancia sólo trabaja en un idioma.

## Punto de partida. La “BBDD”

- Antes de entrar “en harina”, JavaTV API está diseñado para obtener información de los **SIServices** disponibles **en todos los interfaces existentes** y se accede a ellos desde los **Bouquet** y **TransportStreams**.
- Una vez que tenemos el Service tenemos acceso a cierta información: una básica primero y también otra extendida que se solicita.
- Además de cada Service podemos acceder a la información de Events...

## SIManager API

- ¿ Qué ofrece SIManager ? (interesante: echadle un ojo al código fuente)
  - public static SIManager **createInstance()**
    - Acceso a SIManager. NO CONVIENE CREAR MAS DE UNO: OCUPA MUCHA MEMORIA.
  - public void **setPreferredLanguage**(String language);
  - public String **getPreferredLanguage()**;
    - Establece el idioma en el que deseamos recibir la información. Si no está el que haya. **Si deseas info en dos idiomas necesitas 2 SIManager.**
  - public void **registerInterest**(Locator locator, boolean active)
    - Solicitar interés en un **SIElement** o notificar que ya no lo hay. Puede o no hacer caso (no hay manera de saberlo)
  - public String[] **getSupportedDimensions()**;
  - public RatingDimension **getRatingDimension**(String name)
    - Lista de criterios de **clasificación de contenidos** disponibles usados para **control parental**. p.e. MPAA (Motion Picture Association of America) Movie Ratings , <http://www.mpa.org/FilmRatings.asp>

## SIManager API

- ¿ Qué ofrece SIManager ?
  - public javax.tv.service.transport.Transport[] **getTransports()**;
    - Lista de **Transport** Disponibles. No es lo que os imagináis sino una colección de: **BouquetCollection, NetworkCollection, TransportStreamCollection**
  - public javax.tv.service.Service **getService**(Locator locator)
    - Solicita información de un **Service**.
  - public SIRequest **retrieveServiceDetails**(Locator locator,SIRequestor requestor)
    - Solicita un **ServiceDetails** (extends **SIElement**)

## SIManager API

- ¿ Qué ofrece SIManager ?

- public SIRequest **retrieveSIElement**(Locator locator, SIRequestor requestor)
  - Solicita información acerca de uno o varios **SIElement** identificados por el Locator
- public SIRequest **retrieveProgramEvent**(Locator locator, SIRequestor requestor)
  - Solicita un **ProgramEvent** para un Locator
- public ServiceList **filterServices**(ServiceFilter filter);
  - Solicita una serie de javax.tv.service.**Service** mediante un filtro para el cual si es null ofrece todos los conocidos.
  - El Filtro de filterServices es francamente simple:

```
public abstract class ServiceFilter {  
    public abstract boolean accept(javax.tv.service.Service service);  
}
```

## JavaTV API. Antes de seguir, Modelo asíncrono de peticiones

- Similar a DVB SI, pero con alguna “pega”
- En DVB-SI implementamos el listener `SIRetrievalListener` y en el momento de hacer la solicitud se nos devuelve un `SIRequest` por si queremos cancelar. En javaTV implementamos el listener `javax.tv.service.SIRequestor` y en el momento de hacer la solicitud se nos devuelve un `javax.tv.service.SIRequest`:

```
public interface SIRequestor {  
    public void notifySuccess(SIRetrieval[] result);  
    public void notifyFailure(SIRequestFailureType reason);  
}  
public interface SIRequest{  
    public boolean cancel();  
}
```

- Que no cunda el pánico: `public interface SIElement extends SIRetrieval`
- **La pega:** cuando hacemos una petición **no podemos pasar un appData** para saber a qué solicitud corresponde la respuesta-> hay que crear instancias de objetos Listener específicos si queremos saberlo.
- **Una ventaja:** `SIRequestor` nos facilita la vida al separar el éxito del fracaso

- Vamos a seguir el API a partir de **SIManager.getTransport()**. Ofrece una buena visión y los elementos a analizar son accesibles por otros métodos.

## TransportStreamCollection API

- public **SIRequest retrieveTransportStream**(Locator locator, SIRequestor requestor)
  - Solicita un `javax.tv.service.transport.TransportStream` de la colección
- public abstract **SIRequest retrieveTransportStreams**(**SIRequestor** requestor);
  - Devuelve el array de `TransportStream` existentes. Sólo incluye aquellos para los que `javax.tv.service.ReadPermission(ts.getLocator())= true`
- public void **addTransportStreamChangeListener**(TransportStreamChangeListener liste);
- public void **removeTransportStreamChangeListener**(TransportStreamChangeListener li);
  - Suscripción/de-suscripción de notificación de cambios en la colección de TS: añadidos, eliminados o actualizados (`SICChangeType`). No se garantiza que se detecten estos cambios (ved el API)

```
public interface TransportStreamChangeListener extends SICChangeListener {  
    public void notifyChange(TransportStreamChangeEvent event);  
}
```

## TransportStreamCollection API

- TransportStreamChangeEvent: Ofrece “bastante” información:

```
public class TransportStreamChangeEvent extends TransportSIChangeEvent {
    public TransportStreamCollection getTransportStreamCollection() // fuente del evento
    public TransportStream getTransportStream() //objeto afectado
}
public class TransportSIChangeEvent extends SIChangeEvent {
    public Transport getTransport() // la collection en sí.
}
public class SIChangeEvent extends EventObject {
    public SIElement getSIElement()
    public SIChangeType getChangeType()
}
public class SIChangeType {
    public static final SIChangeType ADD;
    public static final SIChangeType REMOVE;
    public static final SIChangeType MODIFY;
}
```

## javax.tv.service.transport.TransportStream API

- **Parte común: SIElement extends SIRetrieveable**

- public javax.tv.locator.Locator **getLocator()**;
  - Locator
- public javax.tv.service.ServiceInformationType **getServiceInformationType()**;

- Describe el formato por el que ha llegado: ATSC, DVB-SI...

```
public class ServiceInformationType {  
    public static final ServiceInformationType ATSC_PSIP;  
    public static final ServiceInformationType DVB_SI;  
    public static final ServiceInformationType SCTE_SI;  
    public static final ServiceInformationType UNKNOWN;  
}
```

- **Parte común: SIRetrieveable**

- public Date **getUpdateTime()**;
  - Momento en el que este objeto se actualizó a partir de datos del Broadcast

## javax.tv.service.transport.TransportStream API

- **Parte propia:**
  - public int **getTransportStreamID()**;
    - Transport Stream ID
  - public String **getDescription()**;
    - Descripción

## Ejercicios Bloque JAVSI-1

## BouquetCollection API

- public SIRequest **retrieveBouquets**(SIRequestor requestor);
  - Solicita los Bouquets existentes de la colección
- public void **addBouquetChangeListener**(BouquetChangeListener listener);
- public void **removeBouquetChangeListener**(BouquetChangeListener listener);
  - Suscripción/de-suscripción de notificación de cambios en la colección de Bouquets: añadidos, eliminados o actualizado (SIChangeType). No se garantiza que se detecten estos cambios (ved el API). Sigue mismo esquema de TransportStreamListener

```
public interface BouquetChangeListener extends SIChangeListener {
    public void notifyChange(BouquetChangeEvent event);
}
public class BouquetChangeEvent extends TransportSIChangeEvent {
    public BouquetCollection getBouquetCollection()
    public Bouquet getBouquet()
}
```

## javax.tv.service.transport.Bouquet API

- **Parte común: SIElement.** Ídem a TransportStream.
- **Parte propia:**
  - public int **getBouquetID()**;
    - Transport Stream ID
  - public String **getName()**;
    - Nombre

## Ejercicios Bloque JAVSI-2

## NetworkCollection API

- public SIRequest **retrieveNetwork**(Locator locator, SIRequestor requestor)
  - Solicita el Network indicado
- public SIRequest **retrieveNetworks**(SIRequestor requestor);
  - Solicita los Network existentes de la colección
- public void **addNetworkChangeListener**(NetworkChangeListener listener);
- public void **removeNetworkChangeListener**(NetworkChangeListener listener);
  - Suscripción/de-suscripción de notificación de cambios en la colección de Networks: añadidos, eliminados o actualizado (SIChangeType). No se garantiza que se detecten estos cambios (ved el API). Sigue mismo esquema de TransportStreamListener

```
public interface NetworkChangeListener extends SIChangeListener {
    public void notifyChange(NetworkChangeEvent event);
}
public class NetworkChangeEvent extends TransportSIChangeEvent {
    public NetworkCollection getNetworkCollection()
    public Network getNetwork ()
}
```

## javax.tv.service.transport.Network API

- **Parte común: SIElement.** Ídem a TransportStream.
- **Parte propia:**
  - public int **getNetworkID()**;
    - network ID
  - public String **getName()**;
    - Nombre
  - public SIRequest **retrieveTransportStreams**(SIRequestor requestor);
    - Los transport Streams de la Network. Ya hemos visto el API de TransportStream

## Ejercicios Bloque JAVSI-3

## javax.tv.service.Service API

- Acceso a Service: a través de **SIManager** como ya hemos visto:

```
ServiceList filterServices(ServiceFilter filter);  
Service getService(Locator locator)
```

- El API

- public SIRequest **retrieveDetails**(SIRequestor requestor);
  - Devuelve lista de javax.tv.service.navigation.ServiceDetails para este Service, tantos como haya disponibles
- public String **getName**();
  - Nombre
- public boolean **hasMultipleInstances**();
  - Si es posible sintonizarlo en diferentes TS

## javax.tv.service.Service API

- EI API

- public ServiceType **getServiceType()**;

- Si DATA, VOD...

```
public class ServiceType {  
    public static final ServiceType DIGITAL_TV;  
    public static final ServiceType DIGITAL_RADIO;  
    public static final ServiceType NVOD_REFERENCE;  
    public static final ServiceType NVOD_TIME_SHIFTED;  
    public static final ServiceType ANALOG_TV;  
    public static final ServiceType ANALOG_RADIO;  
    public static final ServiceType DATA_BROADCAST;  
    public static final ServiceType DATA_APPLICATION;  
    public static final ServiceType UNKNOWN;  
}
```

- public Locator **getLocator()**;

- Su Locator

## `javax.tv.service.navigation.ServiceDetails`

- Clase que Representa al Service recuperado del TS. Obtenemos ServiceDetails a partir de:
  - **Service:** `public SIRequest retrieveDetails(SIRequestor requestor);`
    - ServiceDetails del Service,
  - **SIManager:** `public SIRequest retrieveServiceDetails(Locator locator,SIRequestor requestor)`
    - Para uno en concreto

## javax.tv.service.navigation.ServiceDetails API

- **Parte común:** Hereda de SIElement (Ídem a TransportStream )
- **Parte Propia:**
  - public SIRequest **retrieveServiceDescription**(SIRequestor requestor);
    - Devuelve una javax.tv.service.navigation.ServiceDescription si la hay
  - public ServiceType **getServiceType**();
    - Tipo del Service ( ídem a Service)
  - public SIRequest **retrieveComponents**(SIRequestor requestor);
    - Devuelve los javax.tv.service.navigation.ServiceComponent (Streams) del Servicio. Sólo los incluye si javax.tv.service.ReadPermission(c.getLocator())=true
  - public javax.tv.service.guide.ProgramSchedule **getProgramSchedule**();
    - Devuelve un ProgramSchedule asociado

## javax.tv.service.navigation.ServiceDetails API

- **Parte Propia:**

- public String **getLongName()**;
  - Nombre completo
- public Service **getService()**;
  - Objeto Service asociado
- public void **addServiceChangeListener**(ServiceChangeListener list);
- public void **removeServiceChangeListener**(ServiceChangeListener l)
  - Listener para sabe cuando se añaden, elimina o actualizan Componentes.

```
public interface ServiceChangeListener extends SChangeListener {  
    public void notifyChange(ServiceChangeEvent event);  
}
```

## javax.tv.service.navigation.ServiceDetails API

- **Parte Propia:**

- public javax.tv.service.navigation.DeliverySystemType **getDeliverySystemType()**;
  - Cómo se ha transmitido:DBV, ATSC...

```
public class DeliverySystemType {  
    public static final DeliverySystemType SATELLITE;  
    public static final DeliverySystemType CABLE;  
    public static final DeliverySystemType TERRESTRIAL;  
    public static final DeliverySystemType UNKNOWN;  
}
```

## javax.tv.service.navigation.ServiceComponent API

- **Parte común:** Hereda de SIElement (Ídem a TransportStream )
- **Parte Propia:**
  - public String **getName()**;
    - Nombre
  - public String **getAssociatedLanguage()**;
    - Idioma por defecto ( código de 3)
  - public Service **getService()**;
    - Service al que pertenece

## javax.tv.service.navigation.ServiceComponent API

- **Parte Propia:**

- public StreamType **getStreamType()**;
  - Tipo de Stream

```
public class StreamType {  
    public static final StreamType VIDEO;  
    public static final StreamType AUDIO;  
    public static final StreamType SUBTITLES;  
    public static final StreamType DATA;  
    public static final StreamType SECTIONS;  
    public static final StreamType UNKNOWN;  
}
```

## **javax.tv.service.navigation.ServiceDescription API**

- **Parte común:** Hereda de SIRetrievable
  - public Date **getUpdateTime()**;
    - Momento en el que este objeto se actualizó a partir de datos del Broadcast
- **Parte Propia:**
  - public String **getServiceDescription()**;
    - Descripción del Servicio

## Ejercicios Bloque JAVSI-4

## ¿ Qué ofrece javax.tv.service.guide.ProgramSchedule ?

- Representa una colección de Programas ordenados en el tiempo de un determinado Servicio. Ofrece el programa actual, el siguiente y los futuros
- API. No hereda de nadie.
  - public SIRequest **retrieveCurrentProgramEvent**(SIRequestor requestor);
    - Devuelve el javax.tv.service.guide.ProgramEvent actual
  - public SIRequest **retrieveFutureProgramEvent**(Date time,SIRequestor requestor)
    - Siguiendo javax.tv.service.guide.ProgramEvent comenzando en la fecha indicada inclusive
  - public SIRequest **retrieveFutureProgramEvents**(Date begin, Date end,SIRequestor requestor)
    - Todos los javax.tv.service.guide.ProgramEvent en el intervalo
  - public SIRequest **retrieveProgramEvent**(Locator locator, SIRequestor requestor)
    - Evento para ese Locator.

## javax.tv.service.guide.ProgramSchedule API

- API. No hereda de nadie.
  - public SIRequest **retrieveNextProgramEvent**(ProgramEvent event, SIRequestor requestor)
    - Evento que sigue al pasado
  - public void **addListener**(ProgramScheduleListener listener);
  - public void **removeListener**(ProgramScheduleListener listener);
    - Listener opera cambios en el Schedule. Esquema habitual.

```
public interface ProgramScheduleListener extends SIChangeListener {  
    public void notifyChange(ProgramScheduleEvent event);  
}
```
  - public Locator **getServiceLocator**();
    - Locator del Service al que pertenece el Locator

## ¿ Qué ofrece `javax.tv.service.guide.ProgramEvent` ?

- Representa un Programa/Show
- **Parte común:** hereda de `SIElement`
- **Parte propia:**
  - `public java.util.Date getStartTime() /getEndTime();`
    - Comienzo y Fin
  - `public long getDuration();`
    - Duración en segundos
  - `public String getName();`
    - Nombre
  - `public SIRequest retrieveDescription(SIRequestor requestor);`
    - Si deseamos obtener una descripción mayor: `javax.tv.service.guide.ProgramEventDescription`

## javax.tv.service.guide.ProgramEvent API

- **Parte propia:**
  - public javax.tv.service.guide.ContentRatingAdvisory **getRating()**;
    - Rating. Ved el API
  - public Service **getService()**;
    - Service al que pertenece
  - public SIRequest **retrieveComponents**(SIRequestor requestor);
    - **javax.tv.service.navigation.ServiceComponent** parte de este Event

## **javax.tv.service.guide.ProgramEventDescription API**

- **Parte común:** Hereda de SIRetrievable
  - public Date **getUpdateTime()**;
    - Momento en el que este objeto se actualizó a partir de datos del Broadcast
- **Parte Propia:**
  - public String **getProgramEventDescription ()**;
    - Descripción del programa

## Ejercicios Bloque JAVSI-5

## Procesado de Mensajes. CUIDADO

- Cuando se reciben las notificaciones de información solicitada, cuidado si realizamos tareas pesadas en la recepción, pues podríamos ralentizar la notificación de eventos y eso no es bueno.
- Una opción es crear una gestión interna en la app para “encolar” los eventos de manera que el proceso pesado de esto sea un problema en todo caso de mi aplicación

|                         |   |
|-------------------------|---|
| <b>ISO/IEC 13818-1</b>  | Part 1. Elementary Streams transport definition   |
| <b>ISO/IEC 13818-6</b>  | Part 6. Extensions for DSM-CC. Digital Storage Media Command and Control  |
| <b>ETSI EN 300 468</b>  | Digital Video Broadcasting (DVB);Specification for Service Information (SI) in DVB systems  |
| <b>ETSI EN 301 192</b>  | DVB specification for data broadcasting   |
| <b>ETSI TR 101 202</b>  | Implementation Guidelines for Data broadcasting   |
| <b>ETSI TR 101 162</b>  | Digital broadcasting systems for television, sound and data services; Allocation of Service Information (SI) codes for Digital Video Broadcasting (DVB) systems |
| <b>ETSI TR 102 154</b>  | Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in Contribution and Primary Dist   |
| <b>ETSI TR 101 211</b>  | Guidelines on implementation and usage of Service Information (SI)  |
| <b>ETSI TR 101 200</b>  | Digital Video Broadcasting (DVB); A guideline for the use of DVB specifications and standards   |
| <b>DAVIC</b>            | Digital Audio Visual Council. davic 1.4.1   |
| <b>HAVI</b>             | Specification of the Home Audio/Video Interoperability (HAVi) Architecture  |
| <b>Interactivetvweb</b> | <a href="http://www.interactivetvweb.org/">http://www.interactivetvweb.org/</a>   |
| <b>Wikipedia DSMCC</b>  | <a href="http://en.wikipedia.org/wiki/DSM-CC">http://en.wikipedia.org/wiki/DSM-CC</a>   |
| <b>MHP 1.1.2</b>        | Multimedia Home Platform, A068r1 & tam668r23_11xdraft_20061115  |
| <b>MHP 1.1.3</b>        | Multimedia Home Platform, A068r3  |
| <b>CDC 1.1</b>          | Connected Device Configuration (CDC) 1.1 (JSR=218).   |
| <b>PBP 1.1</b>          | Personal Basis Profile 1.1 (JSR 217)  |
| <b>MHP.org</b>          | <a href="http://www.mhp.org">www.mhp.org</a>  |
| <b>INTRO MHP 1.1.3</b>  | tam1032r1-mhp-iptv-presentation   |