

## *Programa Saiotek 2006*

# SMARTLAB

Entorno de Trabajo Inteligente  
Colaborativo y Programable

---

Interfaces bundles OSGi Integrables  
con Web Interface Gateway





## HISTORIAL DE CAMBIOS

<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>	<b>Fecha</b>	<b>Comentarios</b>
V0.1	Versión inicial	Xabier Laiseka	17/12/2007	

## TABLA DE CONTENIDOS

Historial de cambios .....	3
Tabla de contenidos .....	4
1 Introducción.....	5
1.1 Interfaz DeviceService .....	5
1.2 Interfaz Displayable .....	5
1.2.1 Plantillas HTML.....	6

## 1 INTRODUCCIÓN

---

En este documento se van a explicar que requisitos se deben cumplir para desarrollar dispositivos y servicios con interfaz gráfica en forma de gadget web para la plataforma SmartLab.

El primer requisito que debe cumplir un servicio o dispositivo es ser registrado como un servicio OSGi con una propiedad llamada "type". Esta propiedad puede contener dos valores, "device" (dispositivo físico) y "service" (dispositivo lógico), según cuál sea su naturaleza. De ésta manera, será añadido automáticamente al sistema.

Por otro lado, existen dos interfaces Java. La primera es necesaria para todos los dispositivos y servicios (DeviceService), la otra (Displayable) es para los dispositivos que desean ser visualizados en la interfaz gráfica. Actualmente, no se tienen en cuenta los servicios que implementan esta interfaz.

### 1.1 Interfaz DeviceService

La interfaz DeviceService debe ser implementada por todos los servicios y dispositivos del sistema (es.deusto.tecnologico.smartlab.devicefinder.model.DeviceService). Esta interfaz contiene los siguientes métodos:

- public String getName(): devuelve el nombre del dispositivo o servicio.
- public void setId(Long id): este método es llamado cuando se registra el servicio. Se le pasa como argumento el identificativo del dispositivo o servicio.
- public Long getId(): devuelve el identificativo que es asignado mediante el método setId.

Esta interfaz es ofrecida por el bundle SmartLabCommons.

### 1.2 Interfaz Displayable

La interfaz Displayable debe ser implementada por todos los dispositivos que se quieran visualizar en la interfaz gráfica (es.deusto.tecnologico.smartlab.devicefinder.Displayable). Esta interfaz contiene los siguientes métodos:

- public String getDisplayableGui(): devuelve la interfaz HTML del dispositivo. Puede que en un futuro, se le pase un String para poder pedir diferentes tipos de interfaz gráfica.

- `public String getDisplayableName():` devuelve un nombre para la interfaz gráfica.
- `public Size getDisplayableSize():` devuelve el tamaño de la interfaz HTML. Al igual que en el método `getDisplayableGui`, en un futuro, se le pasará un `String` para poder pedir diferentes tipos de interfaz gráfica.

### 1.2.1 Plantillas HTML

En el método `getDisplayableGui` para la generación del código HTML se utilizan plantillas. Estas plantillas son código HTML con etiquetas adicionales añadidas que serán reemplazadas en el método mencionado. Estas etiquetas tienen el siguiente formato: “<%nombre\_etiqueta%>”.

Para facilitar este reemplazo se ha creado una clase de utilidad que lo hace automáticamente (`es.deusto.tecnologico.smartlab.util.HtmlPreprocessor`). Esta clase contiene los siguientes métodos:

- `public static String preprocess(String text, Map<String, String> values)`
- `public static String preprocess(InputStream is, Map<String, String>) throws IOException`

Estos dos métodos devuelven en un `String` el HTML preprocesado, es decir, cambiando las etiquetas adicionales por un valor. Para realizar esta función se les deben pasar dos parámetros a estas funciones. El primero es el texto a preprocesar, que en el primer método se pasa mediante un `String` y en el segundo mediante un `InputStream`. El segundo parámetro es un `Map`. Las claves de este `Map` son los nombres de las etiquetas (sin “<%” y “%>”) a reemplazar y el valor asociado es el valor por el que se debe reemplazar la etiqueta.