

# **REST Services**

## Control de versiones del documento

Versión	Fecha	Autor	Motivo del cambio
1.0	20/05/2015	Antonio Morcillo Martínez	Creación del documento
1.1	21/05/2015	Antonio Morcillo Martínez	Adaptación a la versión 1.1 de REST Services donde se elimina el proveedor de caché basado en EHCaché y se pasa a utilizar Liferay Cache API.
1.2	05/08/2015	Antonio Morcillo Martínez	Actualización de servicios

<b>REST SERVICES (GL RESTFACADE)</b>	<b>4</b>
<b>INSTALACIÓN</b>	<b>4</b>
<b>MODELO DE CLASES</b>	<b>4</b>
<b>SERVICIOS REST</b>	<b>5</b>
ALTA DE UN NUEVO SERVICIO	6
URLS	6
PARÁMETROS	6
CACHEO	6
EVITANDO EL CACHEO	6
LIMPIEZA DE LA CACHÉ	7
EXPORTACIÓN DE SITIOS	7
LOGEO DE VISITAS DE CONTENIDO	7
<b>PROXY DE RECURSOS DEL CREM</b>	<b>7</b>
CARM.ES	8
EXPORTACIÓN	8
<b>PROPIEDADES</b>	<b>9</b>
<b>DEPENDENCIAS</b>	<b>10</b>

## Figuras

FIGURA 1. DIAGRAMA DE CLASES.....	5
-----------------------------------	---

## REST Services (rest-services)

Este componente dota a los servidores *Liferay* de una pasarela de acceso común a servicios externos.

En concreto se ofrece acceso a los siguientes servicios

- **RESTFul CARM:** Acceso a los servicios *REST* ofrecidos por la plataforma *RESTFul CARM* que se encuentra alojada en otro servidor de aplicaciones
- **CREM:** Acceso a los recursos estáticos del portal [www.carm.es/econet](http://www.carm.es/econet) para servirlos desde el nuevo sitio del centro de estadística de la Región de Murcia
- **CARM.ES:** *Scrapping* de la lista de unidades dependientes para su visualización en el portal de transparencia
- **Exportación a PDF, CSV y XLS:** Permite exportar a uno de estos formatos el contenido de una llamada a *RESTFul CARM*

El empleo de este módulo permite que los distintos *portlets* y contenidos web puedan realizar llamadas *AJAX* a servicios externos sin tener que hacer llamadas *cross-site* no permitidas por los navegadores Web por motivos de seguridad ya que este componente actuará como proxy alojado en el mismo dominio que el servidor de páginas web.

## Instalación

Este modulo no es un módulo estándar de *Liferay* a pesar de que se debe instalar en los servidores de *Liferay*.

Al no tratarse de un módulo *Liferay*, **no se debe instalar mediante copia del fichero war en el directorio *deploy***, ya que esto haría que *Liferay* le añadiera información adicional innecesaria a los descriptores del módulo. El despliegue por tanto ha de hacerse como cualquier despliegue estándar, copiando el fichero *war* en la carpeta de aplicaciones del servidor; en el caso de *tomcat* se debe copiar en

```
${LIFERAY_HOME}/${TOMCAT_HOME}/webapps
```

*Liferay* permite la instalación de servlets adicionales como este componente mediante el empleo del mecanismo *LiferayServletDelegate*

## Modelo de clases

A continuación se detalla el modelo de clases del módulo

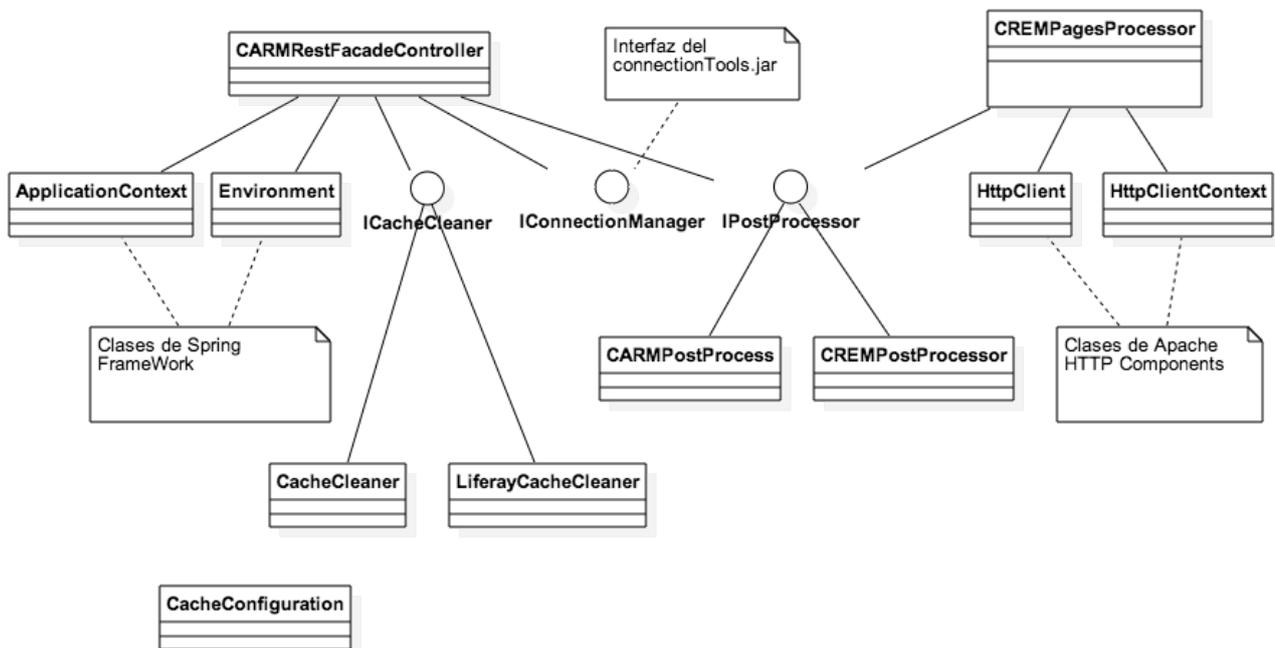


Figura 1. Diagrama de clases

- **CARMRestFacadeController:** Punto de entrada al proxy de Servicios Web REST, al scrapping de **carm.es** y a la exportación a PDF, XLS y CSV.
- **CREMPagesProcessor:** Punto de entrada al proxy de recursos estáticos del CREM
- **ICacheCleaner:** Interfaz para los borradores de caché
- **LiferayCacheCleaner:** Clase responsable de gestionar el borrado de la caché de servicios web para Liferay Cache
- **CacheCleaner:** Clase responsable de gestionar el borrado de la caché de servicios web para EHCACHE (Abandonado en la versión 1.1 de Rest Services)
- **CARMPostProcess:** Clase que procesa el HTML recibido de **carm.es** y elimina elementos innecesarios
- **CREMPostProcessor:** Clase que procesa el HTML recibido de *econet* y elimina elementos innecesarios
- 

## Servicios REST

Como se ha comentado el módulo ofrece una pasarela a *RESTFuICARM* devolviendo al invocador el resultado, sin modificar, de *RESTFuICARM*.

Esta pasarela recogerá los parámetros suministrados por el invocador y los trasladará a *RESTFuICARM*, también sin modificación alguna

Esta pasarela no sólo provee de un mecanismo para evitar llamadas *cross-site* sino que dota de una caché, no persistente, al sistema.

Desde un punto de vista de alto nivel, la caché se comporta como un mapa asociativo donde la clave es la *URL* solicitada y el valor almacenado en caché es el resultado de la llamada ya sea *XML*, *JSON*, etc.

La pasarela dispone de un mecanismo de resolución de nombres dinámica que hace que el alta de un nuevo servicio web sea una cuestión de añadir un nuevo conjunto de propiedades y configuración de caché

## Alta de un nuevo servicio

Para añadir un nuevo servicio al sistema hay que añadir una nueva propiedad al fichero de propiedades

```
url.get${NOMBRE_SERVICIO}=${URL_SERVICIO}
```

Donde:

- **\${NOMBRE\_SERVICIO}**: Es el identificador del servicio. Generará una *URL* única para acceder al servicio
- **\${URL\_SERVICIO}**: La *URL* base a la que la pasarela debe acceder para obtener datos

## URLS

El módulo ofrece una dirección base para sus todos sus servicios cuya ruta es */rest-services/services*

Sobre esa ruta base la pasarela genera los servicios a partir de la ruta */restAPI*. Por lo que la ruta a un servicio en concreto.

```
/rest-services/services/restAPI/${NOMBRE_SERVICIO}
```

## Parámetros

Es posible especificar una lista de parámetros a la llamada añadiéndolos al final de la URL separados por comas

```
/rest-services/services/restAPI/${NOMBRE_SERVICIO}/{PARAMETER_LIST}
```

## Cacheo

El sistema dispone de un mecanismo de cacheo automático de la respuesta de los servicios externos para evitar sobrecargarlos con llamadas innecesaria.

Este mecanismo hace uso del sistema de caché distribuida de *Liferay* llamado *MultiVMPool* que ofrece de serie la posibilidad de crear *cachés* distribuidas en el *cluster* y mantenerlas sincronizadas mediante el paso de mensajes *RMI*

## Evitando el cacheo

Es posible indicar a la pasarela que no se desea que se cachee el resultado mediante el parámetro *HTTP*:

```
nocache=true
```

## Limpieza de la caché

El módulo ofrece la posibilidad de borrar la caché completa de un servicio mediante la invocación a una *URL*. La forma de borrar una caché es usar la *URL* base del servicio pero invocándola mediante el método *HTTP DELETE*

```
/rest-services/services/restAPI/${NOMBRE_SERVICIO}
```

**Nótese que al no estar usando el mecanismo de caché de *Liferay* la limpieza afectará a todos los nodos del cluster, encargándose *Liferay* de notificar a todos los nodos del cambio de estado de las cachés**

## Exportación de sitios

El módulo ofrece la posibilidad de exportar un sitio web de *Liferay* sin necesidad de pasar por la interfaz de exportación. Esta exportación será completa

```
/rest-services/services/exportSite/${scope}/${siteId}
```

donde *scope* indica si se desea exportar el área pública o privada de un sitio y *siteId* es el identificador del sitio a exportar

## Logeo de visitas de contenido

El módulo ofrece la posibilidad registrar en base de datos el acceso a un contenido web por parte de un usuario.

```
/rest-services/services/lopd/logread/${userId}/${articleId}
```

donde *userId* es el identificador de usuario y *articleId* es el identificador del contenido

**Este web service sólo es accesible por el método PUT**

## Proxy de recursos del CREM

El proxy de recursos estáticos para el CREM ofrece sus servicios a través de la ruta:

```
/rest-services/services/crem/
```

El portal del CREM dispone de un mecanismo de reescritura de *URLs* que transforma las llamadas a recursos estáticos para que pasen a través de este *proxy* por lo que el funcionamiento del mismo es automático

## CARM.ES

El mecanismo de *scrapping* de *carm.es* para las unidades organizativas ofrece sus servicios a través de la ruta:

```
/rest-services/services/organizaciones/getData/{idUnidadOrganizativa}
```

## Exportación

El mecanismo de exportación a PDF, XLS y CSV ofrece sus servicios a través de la ruta:

```
/rest-services/services/export
```

Este servicio ha de ser invocado mediante el verbo *POST* y debe recibir un JSON de configuración con el siguiente formato

```
var configuration = new Object();
  var columna1 = new Object();
  columna1.nombre = "Código";
  columna1.indicesJSON = [0,1,2];
  columna1.separadores = [" ", "/"];
  columna1.tipo="cadena";

  var columna2 = new Object();
  columna2.nombre = "Tipo Contrato";
  columna2.indicesJSON = [];
  columna2.indicesJSON.push(1);
  columna2.tipo="cadena";

  var columna3 = new Object();
  columna3.nombre = "Objeto";
  columna3.indicesJSON = [];
  columna3.indicesJSON.push(3);
  columna3.tipo="cadena";

  var columna4 = new Object();
  columna4.nombre = "Importe adjudicación";
  columna4.indicesJSON = [];
  columna4.indicesJSON.push(5);
  columna4.tipo="cadena";

  var columna5 = new Object();
  columna5.nombre = "Adjudicatario";
  columna5.indicesJSON = [];
  columna5.indicesJSON.push(6);
  columna5.tipo="cadena";
```

```
var columna6 = new Object();
columna6.nombre = "Fecha formalización";
columna6.indicesJSON = [];
columna6.indicesJSON.push(8);
columna6.tipo="cadena";

var columnasArray = [];
columnasArray.push(columna1);
columnasArray.push(columna2);
columnasArray.push(columna3);
columnasArray.push(columna4);
columnasArray.push(columna5);
columnasArray.push(columna6);
configuration.url = "/rest-services/services/contratos/getData";
configuration.nombreFichero = "contratos";
configuration.columns = columnasArray;
configuration.columnsVisibles = [0,1,2,3,4,5];
```

Donde:

- **url:** URL de donde el servidor ha de sacar los datos para montar el listado completo
- **columnas:** array de objetos con la definición de columnas para el listado completo
- **nombreFichero (opcional):** el nombre sin extensión del fichero a generar
- **columnasVisibles(opcional):** array de enteros que indica, para el listado de página actual, que columnas del *datatable* se mantienen en el listado, si se omite, se exporta todo
- **Objeto columna:**
  - **nombre:** nombre de la columna
  - **indicesJSON:** array de enteros que indica las posiciones en el array *JSON* de donde leer los datos
  - **tipo:** el tipo de datos final para cada uno de los elementos de la columna. Ahora mismo soportados:
    - **cadena**
    - **imagen**
    - **fecha**
  - **separadores(opcional):** si en el array *indicesJSON* se ha especificado más de un valor, se puede especificar un array de cadenas que se intercalaran entre cada uno de los valores obtenidos.

## Propiedades

El hook además dispone de estas propiedades para parametrizar el funcionamiento del mismo según el entorno para el que se compila:

**rest-facade.auth.user:** Usuario con el que debe autenticarse en *Liferay* para servicios autenticados

**rest-facade.auth.password:** Password con el que debe autenticarse en *Liferay* para servicios autenticados

**rest-facade.connectionParams.connectionTimeout:** *Timeout* de conexión

**rest-facade.connectionParams.readTimeout:** *Timeout* de lectura

**rest-facade.url.cremBaseURL:** URL base del CREM para realizar las conexiones

**rest-facade.url-exportURLPrefix:** URL base del servicio de exportación para realizar las conexiones

No se listan aquí las propiedades de los servicios web dinámicos, cuyo funcionamiento se ha detallado en el apartado de servicios Web.

## Dependencias

Finalmente se listan las dependencias propias del módulo con independencia de las heredadas desde el *pom.xml* del módulo padre:

- **spring-web y spring-webmvc-portlet v 4.1.0.RELEASE:** Construcción de controladores REST basados en anotaciones
- **google-guava v18:** Utilidades
- **Apache HTTP Client v 4.3.1:** Empleada para realizar las conexiones *HTTP*
- **Apache Log4j 1.2.17:** Sistema de logeo. No suministrado por *Liferay* al no ser este módulo plugin de *Liferay*
- **Apache Log4j Extras 1.2.17:** Extensiones del sistema de logeo no suministradas por *Liferay*
- **connectionTools 1.0:** Librerías propias de gestión de conexiones
- **exportTools 1.0:** Librerías propias que implementan el servicio de exportación
- **spring-jdbc, spring-orm, y spring-tx v 4.1.0 RELEASE** para el acceso a base de datos
- **spring data jpa v 1.2.0 RELEASE** para el acceso a base de datos
- **hibernate jpa v 2.0** para el acceso a base de datos
- **hibernate-entitymanager v 4.0.1 Final** para el acceso a base de datos