

DDT Cobro modelo N28

Documento de Diseño Técnico

Fecha de realización: **08/08/2018**

**Fecha última revisión:**

**Versión nº: 1**

**Capítulo afectado:**

**Autor modificación:**

**ÍNDICE**

1 Descripción del documento 7

1.1 Referencias 7

1.2 Definiciones y acrónimos 7

1.3 Control de modificaciones sobre el documento 7

2 Catálogo de servicios de la aplicación 8

3 Arquitectura de los datos 9

4 Arquitectura de la aplicación 10

4.1 Diagrama de clases 11

4.2 Ciclo de vida de una petición 12

4.3 Diseño de los tratamientos on-line en host 12

4.4 Diseño de los tratamientos diferidos 12

4.5 Diseño de la lógica de presentación 12

5 Descripción detallada de componentes 13

5.1 Servidores de datos 13

5.2 Componentes Folder 13

5.3 Componentes de lógica de negocio 13

5.4 Programas batch 13

5.5 Componentes Java 13

5.5.1 es.cajamadrid.pit.filter 13

5.5.1.1 RequestTrackingFilter 13

5.5.1.2 RequestTrackingData 13

5.5.1.2.1 finalizarPeticion 14

5.5.1.2.2 inicializarPeticion 14

5.5.2 es.cajamadrid.pit.servlet 14

5.5.2.1 ProxyServlet 14

5.5.2.1.1 HTML\_MENSAJES 14

5.5.2.1.2 createContentManagementContext(HttpServletRequest) 15

5.5.2.1.3 createHttpClient(ContentManagementContext) 15

5.5.2.1.4 doGet(HttpServletRequest, HttpServletResponse) 15

5.5.2.1.5 doPost(HttpServletRequest, HttpServletResponse) 15

5.5.2.1.6 executeImAliveResponse(HttpServletRequest, HttpServletResponse) 15

5.5.2.1.7 executeProxyRequest(HttpMethod, HttpServletRequest, HttpServletResponse, ContentManagementContext) 15

5.5.2.1.8 getDateLastCached(ContentManagementContext) 15

5.5.2.1.9 getServletInfo() 15

5.5.2.1.10 handleConnectionException(ConnectException, HttpServletResponse) 15

5.5.2.1.11 init(ServletConfig) 15

5.5.2.1.12 isIAmAlivePage(HttpServletRequest) 15

5.5.2.1.13 updateContext(ContentManagementContext, int, HttpMethod) 15

5.5.2.1 AdminServlet 15

5.5.3 es.cajamadrid.pit.servlet.core 16

5.5.3.1 HeaderConstants 16

5.5.3.1.1 CONTENT\_LENGTH 16

5.5.3.1.2 CONTENT\_TYPE 16

5.5.3.1.3 HEADER\_F5 16

5.5.3.1.4 HEADER\_LANGUAGE 16

5.5.3.1.5 HOST 16

5.5.3.1.6 LOCATION 16

5.5.4 es.cajamadrid.pit.servlet.utils 16

5.5.4.1 RegExpUtils 16

5.5.4.1.1 createMatcher(String, String) 16

5.5.4.1.2 getFirstNotNullGroupNumber(Matcher, int[]) 16

5.5.4.1.3 getFirstNotNullMatch(Matcher, int[]) 17

5.5.5 es.cajamadrid.pit.servlet.manager 17

5.5.5.1 ConfigurationManager 17

5.5.5.1.1 getContentInterceptors 17

5.5.5.1.2 getIgnoreContentUpdateChecking 17

5.5.5.1.3 getIsoTowebServiceLanguageMapping 17

5.5.5.1.4 getSiteResolvers 17

5.5.5.1.5 getWebServiceTargetEndPoint 17

5.5.5.2 ContentManager 17

5.5.5.2.1 calculateHash 18

5.5.5.2.2 checkContentEncoding 18

5.5.5.2.3 checkFileAvailable 18

5.5.5.2.4 copyContentToResponse 18

5.5.5.2.5 copyHeadersToResponse 18

5.5.5.2.6 escapeExcludedTagsInOriginalContent 18

5.5.5.2.7 extractTextsToTranslateFromOriginalContent 18

5.5.5.2.8 generateTranslatedContent 19

5.5.5.2.9 handleContent 19

5.5.5.2.10 isCachedAndValid 19

5.5.5.2.11 makeAbsolutHrefSrc 19

5.5.5.2.12 makeAbsolutUrl 19

5.5.5.2.13 proxyProcess 19

5.5.5.2.14 saveCache 19

5.5.5.2.15 translateHrefSrc 19

5.5.5.2.16 translateStatics 19

5.5.5.2.17 translateTexts 19

5.5.5.2.18 translateUrls 19

5.5.5.3 HeaderManager 19

5.5.5.3.1 copyHeadersToProxyRequest 19

5.5.5.4 LanguageManager 20

5.5.5.5 PostManager 20

5.5.5.5.1 handleMultipartPost 20

5.5.5.5.2 handleStandardPost 20

5.5.5.6 RedirectManager 20

5.5.5.6.1 handle300al303Response 20

5.5.5.6.2 handle304Response 20

5.5.5.6.3 handleRedirect 20

5.5.5.7 TargetURLManager 20

5.5.5.7.1 loadOriginalTargetDataIntoContext 20

5.5.5.7.2 loadResolvedTargetDataIntoContext 21

5.5.5.7.3 loadTargetHostDataIntoContext 21

5.5.5.8 TranslationTextContext 21

5.5.5.8.1 compareTo 21

5.5.5.8.2 getBeginPos 21

5.5.5.8.3 getEndPos 21

5.5.5.8.4 getOriginalText 21

5.5.5.8.5 getTranslatedText 21

5.5.6 es.cajamadrid.pit.servlet.manager.cache 21

5.5.6.1 CacheAdmin 21

5.5.6.1.1 getCacheAdmin 22

5.5.6.1.2 flushCacheAll 22

5.5.6.1.3 flushCacheDomain 22

5.5.6.1.4 getCacheKey 22

5.5.6.1.5 getCacheObject 22

5.5.6.1.6 putCacheObject 22

5.5.6.2 CachedObject 22

5.5.6.2.1 Date 22

5.5.6.2.2 Hash 22

5.5.6.2.3 Value 23

5.5.7 es.cajamadrid.pit.servlet.manager.configuration 23

5.5.7.1 ConfigurationDaoPropertiesImpl 23

5.5.7.1.1 ConfigurationDaoPropertiesImpl 23

5.5.7.1.2 ConfigurationDaoPropertiesImpl(Properties) 23

5.5.7.1.3 actualizarSiteResolversTipoRegExp 23

5.5.7.1.4 getContentInterceptors 23

5.5.7.1.5 getIgnoreContentUpdateChecking 23

5.5.7.1.6 getIsoToWebServiceLanguageMapping 23

5.5.7.1.7 getSiteResolvers 23

5.5.7.1.8 getTranslatedContentPath 23

5.5.7.1.9 getWebServiceTargetEndPoint 24

5.5.8 es.cajamadrid.pit.servlet.manager.content 24

5.5.8.1 ContentFinder 24

5.5.8.1.1 calculateAbsoluteUrl 24

5.5.8.1.2 ContentFinder 24

5.5.8.1.3 getContentType 24

5.5.8.1.4 getName 24

5.5.8.1.5 getRegExp 24

5.5.8.1.6 getRegExpGroups 24

5.5.8.1.7 processText 24

5.5.8.1.8 updateUrlToAbsolutePath 24

5.5.8.1.9 updateUrlToAbsolutePathFromRegExp 25

5.5.8.2 ContentManagementContext 25

5.5.8.3 ExclusionFinderFactory 25

5.5.9 es.cajamadrid.pit.servlet.manager.content.interceptors 25

5.5.9.1 BaseContentInterceptor 25

5.5.9.1.1 executeAfterInterception 25

5.5.9.1.2 executeBeforeInterception 25

5.5.9.1.3 executeInterception 25

5.5.9.1.4 isApplicable 25

5.5.9.2 IContentInterceptor 25

5.5.9.2.1 executeInterception 25

5.5.9.2.2 isApplicable 26

5.5.10 es.cajamadrid.pit.servlet.manager.resolvers 26

5.5.10.1 AbstractSiteResolver 26

5.5.10.2 RegExpSiteResolver 26

5.6 Descripción técnica de recursos 26

6 Descripción de estructuras de datos 27

6.1 Interfaces de los servidores de datos 27

6.2 Interfaces de los componentes de lógica 27

6.3 Ficheros utilizados en batch 27

6.4 Entidades del modelo lógico 27

# Descripción del documento

El diseño transforma los modelos y especificaciones obtenidas en el análisis funcional del sistema en las unidades organizativas de datos y algoritmos que se van a requerir para construir la aplicación.

Por tanto este documento describirá CÓMO va a ser físicamente el sistema conteniendo:

* El diseño de la arquitectura modular del software
* El diseño de la arquitectura física de los datos

El documento está dirigido a los programadores encargados de codificar el software y a los diseñadores de las pruebas.

## Referencias

| Título | Autor | Fecha | Editor | Fuentes |
| --- | --- | --- | --- | --- |
| ERU\_Requisitos\_multidioma\_Web\_cajamadrid | Sergio Mesas | 05/08/09 |  |  |
| ERSW PIT | Javier Antoniucci | 07/08/09 |  |  |

## Definiciones y acrónimos

| Término | Definición |
| --- | --- |
| PIT | Proxy Inverso de Traducción |
| REST | La Transferencia de Estado Representacional (Representational State Transfer) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. |

## Control de modificaciones sobre el documento

| Fecha | Versión | Capítulo Afectado | Observaciones | Autor Modificación |
| --- | --- | --- | --- | --- |
| 07/08/09 | 1.0 | TODO | Creación del documento | Javier Antoniucci |
| 12/10/09 | 1.1 | TODO | Modificación total del documento | Carlos Díez-Gil |

# Catálogo de servicios de la aplicación

# Arquitectura de los datos

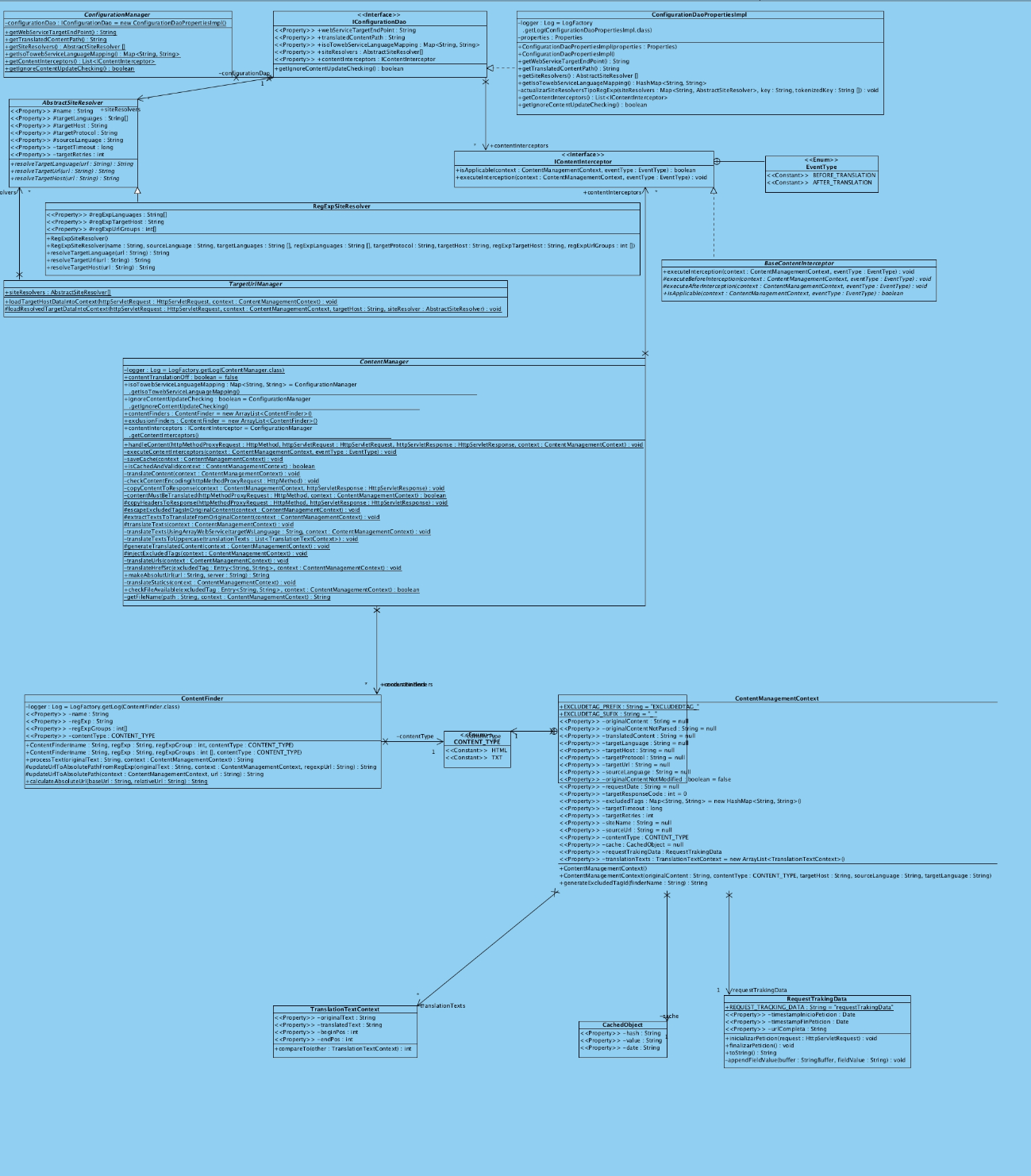
N/A

# Arquitectura de la aplicación

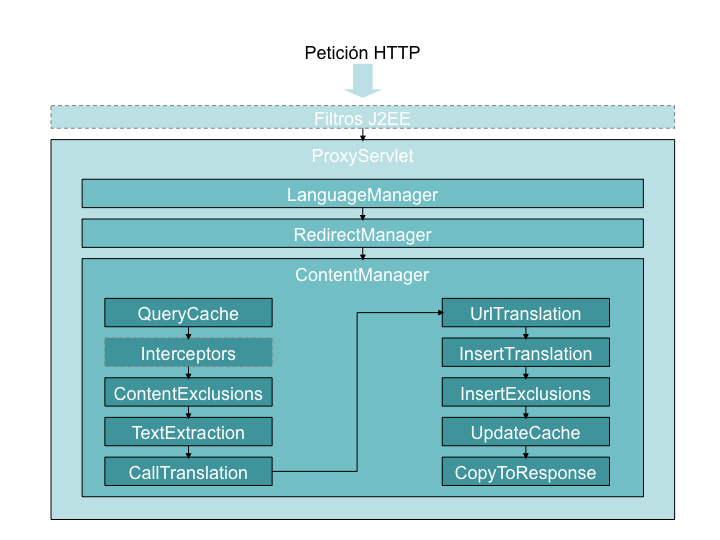


Ver 5.5 “Componentes Java”

## Diagrama de clases



## Ciclo de vida de una petición



## Diseño de los tratamientos on-line en host

N/A

## Diseño de los tratamientos diferidos

N/A

## Diseño de la lógica de presentación

N/A

# Descripción detallada de componentes

## Servidores de datos

Todos los datos necesarios para la ejecución de la aplicación se gestionarán mediante DBProperties.

## Componentes Folder

N/A

## Componentes de lógica de negocio

Ver 5.5 “Componentes Java”

## Programas batch

N/A

## Componentes Java

### es.cajamadrid.pit.filter

Este paquete contiene todo lo relaccionado con los filtros para profiling y análisis de rendimiento. Básicamente consiste en un filtro y un POJO de datos que se almacenará por cada petición.

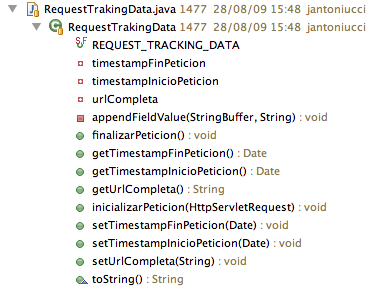
#### RequestTrackingFilter

Genera las trazas para documentar cada solicitud que se realiza al PIT. Utiliza la clase RequestTrackingData como contenedora de datos. Los datos obtenidos serán utilizados para registrar información de la petición y utilizarla en análisis de rendimiento, incidencias, etc…



#### RequestTrackingData

Clase contenedora de los datos de cada solicitud que se realiza al PIT. Con esta clase generamos objetos que contienen todos los datos de profiling que utilizaremos posteriormente en logging.



##### finalizarPeticion

Marca la fecha de fin de una petición para los log.

##### inicializarPeticion

Marcar la fecha de inicio de una petición para los log.

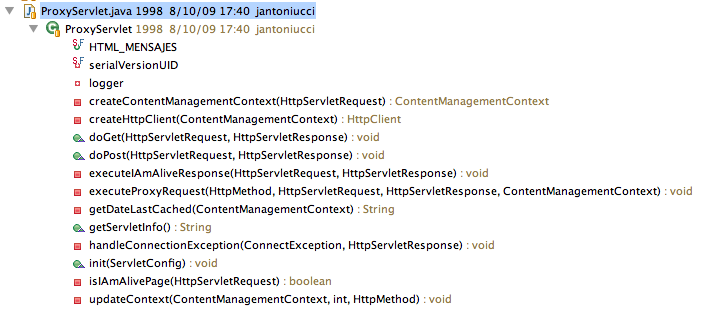
### es.cajamadrid.pit.servlet

Este es el paquete principal del proxy. Incluye los dos servlets que pueden resolver peticiones: ProxyServlet, que realiza el trabajo del proxy y AdminServlet devuelve la página de administración y realiza todas las funciones de la misma.

#### ProxyServlet

Es el punto de entrada de todas las peticiones HTTP y su objetivo es conseguir una traducción del contenido solicitado en el idioma indicado.

El mismo se vale de Managers para realizar esta tarea.



##### HTML\_MENSAJES

Constante que contiene el código HTML necesario para sacar una página que muestre mensajes al usuario.

##### createContentManagementContext(HttpServletRequest)

Este método recibe como parámetro la request de la petición y genera un objeto de tipo ContentManagementContext que almacena el contexto de la petición.

##### createHttpClient(ContentManagementContext)

Genera un objeto de tipo HttpClient (de commons httpclient) con los datos pasados en el contexto.

##### doGet(HttpServletRequest, HttpServletResponse)

Procesa todas las peticiones GET que entren al proxy. Recoge la peticion en la variable HttpServletRequest y completa la HttpServletResponse.

##### doPost(HttpServletRequest, HttpServletResponse)

Procesa las peticiones POST que lleguen al proxy. Funciona igual que el doGet.

##### executeImAliveResponse(HttpServletRequest, HttpServletResponse)

Ejecuta una petición del tipo “i’m alive” y devuelve un estado OK.

##### executeProxyRequest(HttpMethod, HttpServletRequest, HttpServletResponse, ContentManagementContext)

Este método comprueba si la petición realizada está en cacheada. En caso afirmativo, devuelve directamente la página de caché. En caso negativo, ejecuta el HttpMethod contra el servidor proxeadoy envía la respuesta al cliente mediante el HttpServletResponse.

##### getDateLastCached(ContentManagementContext)

Comprueba si una página está cacheada. Devuelve un string con la fecha en que se cacheó. Si no se encuentra devuelve null.

##### getServletInfo()

Devuelve un String con información del servlet. En este caso es constante “Proxy Inverso de Traducción (PIT)”.

##### handleConnectionException(ConnectException, HttpServletResponse)

Controla las excepciones de conexión. En caso de producirse un error de conexión, escribe el log y devuelve una página al usuario, informando del mismo.

##### init(ServletConfig)

Realiza la inicialización del PIT, cargando todos los ficheros de properties necesarios.

##### isIAmAlivePage(HttpServletRequest)

Devuelve un boolean, que indica si la petición realizada es de la página “I’m alive” o no.

##### updateContext(ContentManagementContext, int, HttpMethod)

Actualiza el contexto con los datos proporcionados.

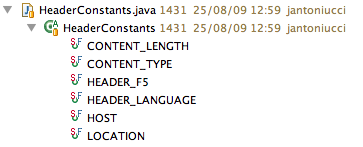
#### AdminServlet

Pendiente de desarrollo.

### es.cajamadrid.pit.servlet.core

#### HeaderConstants

Contiene las constantes con los nombres de las cabeceras HTTP.



##### CONTENT\_LENGTH

Content-Length

##### CONTENT\_TYPE

Content-Type

##### HEADER\_F5

X-Forwarded-for

##### HEADER\_LANGUAGE

X-ProxyMultidioma

##### HOST

Host

##### LOCATION

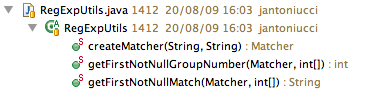
Location

### es.cajamadrid.pit.servlet.utils

Este paquete contiene utilidades que ayuden a los servlets.

#### RegExpUtils

Contiene utilidades para manipular expresiones regulares.



##### createMatcher(String, String)

Genera un Matcher a partir de la expresión regular y el contenido que se pasa como parámetro.

##### getFirstNotNullGroupNumber(Matcher, int[])

Obtiene el número del primer grupo no nulo que cumple con la expresión regular.

##### getFirstNotNullMatch(Matcher, int[])

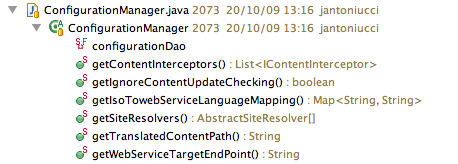
Obtiene el valor del primer grupo no nulo que cumple con la expresión regular.

### es.cajamadrid.pit.servlet.manager

Este paquete contiene todos los managers utilizados por los servlets.

#### ConfigurationManager

Este manager lee las configuraciones y las aplica al PIT.



##### getContentInterceptors

Obtiene una lista de filtros (excepciones de contenido) para el portal en cuestión. Estas excepciones se procesarán posteriormente para solucionar posibles errores en el proxy.

##### getIgnoreContentUpdateChecking

Obtiene el valor de la variable que indica si debemos ignorar si el contenido de la web ha cambiado o no.

##### getIsoTowebServiceLanguageMapping

Devuelve un Map <String, String> con la configuración de idiomas a los que debe responder el portal.

##### getSiteResolvers

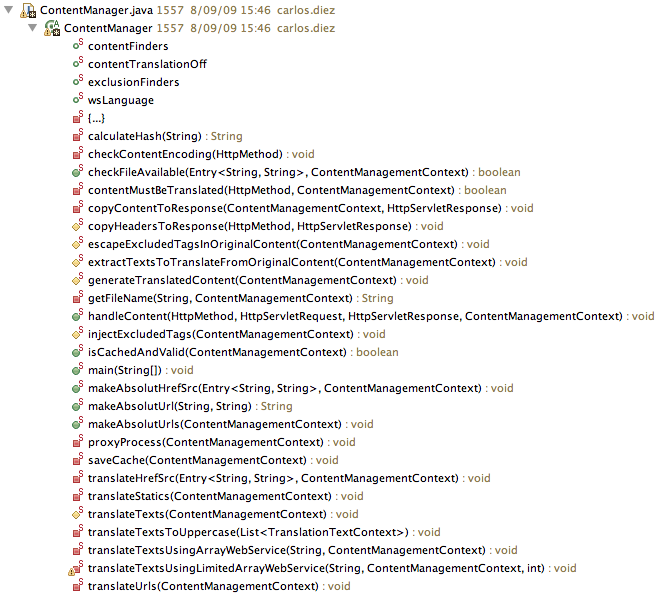
Devuelve los SiteResolvers que debe resolver el portal.

##### getWebServiceTargetEndPoint

Devuelve la URL donde está configurado el WebService de traducción del portal.

#### ContentManager

Componente especializado en la gestión del contenido: extracción de textos, envío a traducción y reemplazos, así como su inclusión en el caché dependiendo de la política seguida.



##### calculateHash

Calcula el Hash de un String, para después almacenarlo en cache.

##### checkContentEncoding

Comprueba el encoding de un contenido.

##### checkFileAvailable

Comprueba la existencia de un fichero en el fileSystem

##### copyContentToResponse

Copia el contenido de una request a la Response.

##### copyHeadersToResponse

Copia el contenido de las cabeceras a la Response.

##### escapeExcludedTagsInOriginalContent

Elimina los tags excluidos dentro del HTML original.

##### extractTextsToTranslateFromOriginalContent

Extrae los textos a traducir del HTML original almacenado en el contexto.

##### generateTranslatedContent

Genera el contenido traducido y lo devuelve en la Response.

##### handleContent

Gestiona el contenido y llama a todos los métodos necesarios para comprobar si un texto se debe traducir o no, y devuelve el contenido.

##### isCachedAndValid

Comprueba si un contenido está en cacheado y no ha caducado.

##### makeAbsolutHrefSrc

Hace absolutas las URLs relativas que están dentro de los SRC.

##### makeAbsolutUrl

Hace absolutas las URLs relativas, para evitar que se procesen através del proxy.

##### proxyProcess

Realiza todas las funciones del proxy.

##### saveCache

Guarda un contenido en la cache para un dominio en particular.

##### translateHrefSrc

Traduce los HREF, para seguir navegando en el mismo idioma en el que nos hayamos logado en el proxy.

##### translateStatics

Traduce los estáticos. Chequea si existen los ficheros traducidos y en el caso de que existan, cambian las URL para que apunten a los recursos traducidos.

##### translateTexts

Traduce textos, utilizando los webservices del servicio de traducción.

##### translateUrls

Traduce las URLs para incorporar el idioma en el que estamos navegando.

#### HeaderManager

Componente especializado en la gestión de encabezados HTTP. Es responsable por la copia, exclusión e inclusión de los encabezados que se envían al portal y al usuario.

Captura de pantalla 2009-09-10 a las 12

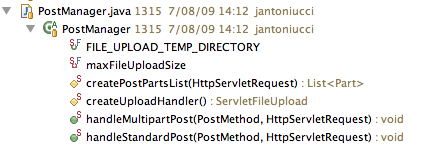
##### copyHeadersToProxyRequest

Pasa las cabecereas a la request que procesa el proxy

#### LanguageManager

#### PostManager

Componente especializado en la gestión de peticiones POST estándar y multipart.



##### handleMultipartPost

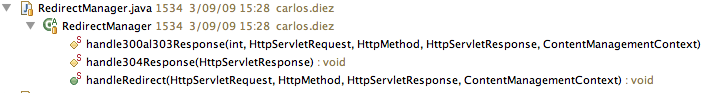
Se ocupa de resolver una petición multipart por POST (un upload de un fichero)

##### handleStandardPost

Se ocupa de resolver una petición normal por POST.

#### RedirectManager

Componente especializado en la gestión de redirecciones (http 300 a 304).



##### handle300al303Response

Gestiona las respuestas del 300 al 303.

##### handle304Response

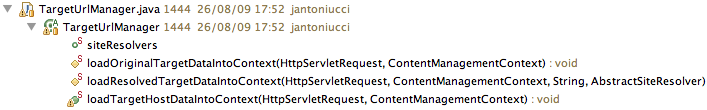
Gestiona las respuestas del tipo304 (No ha cambiado el contenido)

##### handleRedirect

Gestiona todas las redirecciones.

#### TargetURLManager

Componente especializado en calcular la URL del portal al que se le pedirán los contenidos originales antes de ser traducidos.



##### loadOriginalTargetDataIntoContext

Carga los datos originales (el HTML, XML o fichero de texto) en el contexto.

##### loadResolvedTargetDataIntoContext

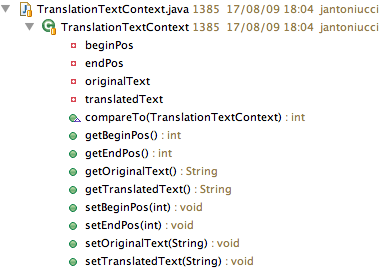
Carga los datos en el contexto.

##### loadTargetHostDataIntoContext

Carga los datos del host origen en el contexto.

#### TranslationTextContext

Esta clase almacena todos los datos que utilizamos para la traducción de textos. Como los métodos son getters y setters estándar, vámos a describir únicamente lo que deben contener las variables, que se accederan por medio de sus accesors estándar.



##### compareTo

Devuelve la resta de las dos posiciones iniciales. Si es 0 entendemos que son la misma cadena a traducir.

##### getBeginPos

Contiene la posicion inicial desde la que traduciremos.

##### getEndPos

Contiene la posicion final hasta le que se debe traducir.

##### getOriginalText

Contiene el texto original (sin traducir).

##### getTranslatedText

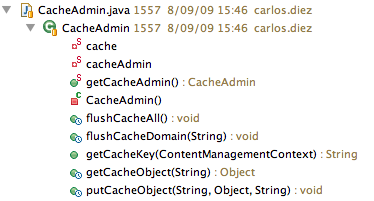
Contiene el texto traducido.

### es.cajamadrid.pit.servlet.manager.cache

Este paquete contiene todas las clases relaccionadas con el cache.

#### CacheAdmin

Esta clase maneja cualquier petición relacionada con la caché.



##### getCacheAdmin

Devuelve una instancia de CacheAdmin.

##### flushCacheAll

Borra todos los elementos que contenga la caché.

##### flushCacheDomain

Limpia la caché para el dominio que se pase como parámetro.

##### getCacheKey

Obtiene la clave de cache para un objeto en particular. La clave está compuesta por varios datos que identifican unívocamente una visita a una URL.

##### getCacheObject

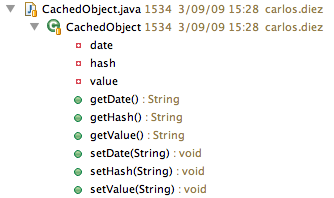
Obtiene de caché el objeto cuya clave coincida con la que se pase como parámetro.

##### putCacheObject

Mete un objeto en caché con la key que se le dé, para el dominio que corresponda.

#### CachedObject

Es la clase que representa a los objetos que se almacenarán en caché. Esta clase define POJOs con las siguientes propiedades:



##### Date

Contiene la fecha en la que se almancenó el objeto en caché.

##### Hash

Contiene el Hash del valor, para una comparación mas rápida.

##### Value

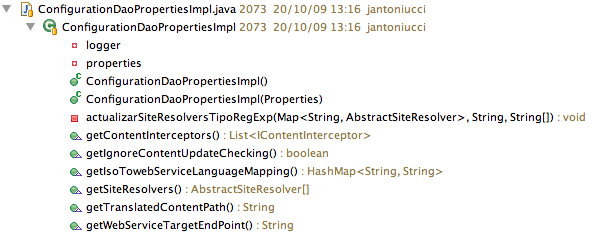
El valor de la página traducida (es un texto que contiene HTML)

### es.cajamadrid.pit.servlet.manager.configuration

Este paquete contiene las clases relaccionadas con la configuración del PIT. Básicamente hay un interfaz que deben implementar todas las configuraciones. Como los métodos son comunes, vamos a describir únicamente la implementación del servicio.

#### ConfigurationDaoPropertiesImpl

Clase que se utilizará para leer todas las configuraciones del PIT.



##### ConfigurationDaoPropertiesImpl

Constructor por defecto.

##### ConfigurationDaoPropertiesImpl(Properties)

Constructor a partir de un fichero de properties determinado.

##### actualizarSiteResolversTipoRegExp

Carga los SiteResolvers desde las expresiones regulares definidas en el fichero de configuración.

##### getContentInterceptors

Obtiene todos los ContentInterceptors (excepciones a las reglas de traducción para por ejemplo, evitar que se traduzca determinado HTML)

##### getIgnoreContentUpdateChecking

Devuelve el valor de la variable de configuración que evita que se chequee si ha cambiado el contenido del servidor destino, para servirlo desde cache directamente.

##### getIsoToWebServiceLanguageMapping

Devuelve toda la lista de idiomas a los que debe traducir el proxy de configuracion para un domienio en concreto.

##### getSiteResolvers

Devuelve todos los siteResolvers a los que debe atender el proxy de traducción.

##### getTranslatedContentPath

Devuelve el contentPath traducido

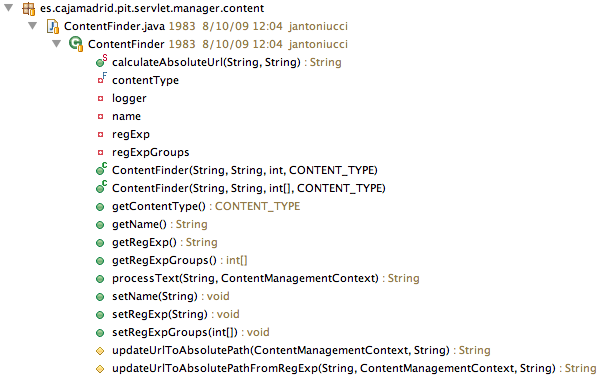
##### getWebServiceTargetEndPoint

Devuelve la URL a la que debe atacar el webservice de traducción.

### es.cajamadrid.pit.servlet.manager.content

Este paquete agrupa todas las clases que tienen que ver con la gestión de contenidos o traducción.

#### ContentFinder



##### calculateAbsoluteUrl

Calcula una URL absoluta, que se utilizará para sacar por fuera del proxy los contenidos estáticos.

##### ContentFinder

Es el constructor del método.

##### getContentType

Devuelve el content-type.

##### getName

Devuelve el nombre.

##### getRegExp

Devuelve la expresión regular que debe cumplir.

##### getRegExpGroups

Devuelve los grupos de expresiones regulares que cumple.

##### processText

Procesa el texto enviado. Devuelve el originalText del contexto.

##### updateUrlToAbsolutePath

Actualiza la URL a un path absoluto.

##### updateUrlToAbsolutePathFromRegExp

Actualiza la URL a un path absoluto desde una expresión regular.

#### ContentManagementContext

Es un POJO que contiene todos las propiedades que pasamos entre todos los managers que tratan con el contenido.

#### ExclusionFinderFactory

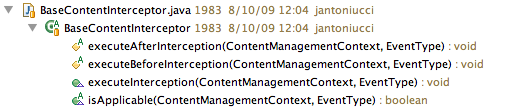
Es una factoria de ExclusionFinder.

### es.cajamadrid.pit.servlet.manager.content.interceptors

Este paquete contiene las clases de intercepción de excepiones. Las clases deben heredar de BaseContentInterceptor e implementar el interfaz IContentInterceptor.

#### BaseContentInterceptor

Clase de la que heredan todos los ContentInterceptor que definiran excepciones.



##### executeAfterInterception

Este método se ejecutará antes de la intercepción. Como recibimos el ContentManagementContext, podemos interactuar contra la el contenido leido antes de procesarlo.

##### executeBeforeInterception

Este método se ejecutará después de la intercepción. Como recibimos el ContentManagementContext, podemos interactuar contra la el contenido traducido antes de mostrar la respuesta al cliente.

##### executeInterception

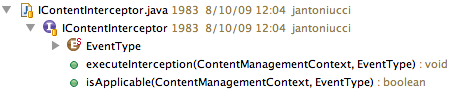
Ejecuta la intercepción.

##### isApplicable

Devuelve un booleano que indica si se debe aplicar la intercepción o no.

#### IContentInterceptor

Interfaz que deben implementar todos los ContentInterceptor.



##### executeInterception

Ejecuta la intercepción.

##### isApplicable

Devuelve un booleano que indica si se debe aplicar la intercepción o no.

### es.cajamadrid.pit.servlet.manager.resolvers

Este paquete incluye todas las clases relaccionadas con los resolvers, tanto estáticos, como por definidos por medio de expresiones regulares.

#### AbstractSiteResolver

#### RegExpSiteResolver

## Descripción técnica de recursos

Pendiente de desarrollarse.

# Descripción de estructuras de datos

## Interfaces de los servidores de datos

Todos los datos necesarios para la ejecución de la aplicación se gestionarán mediante DBProperties, por lo que se utilizará la interfaz de su API.

## Interfaces de los componentes de lógica

Pendiente de incluir la documentación de:

* WebServices de Automatic Trans
* Referencias a la API de DBProperties

## Ficheros utilizados en batch

N/A

## Entidades del modelo lógico

N/A